Honours Research Reports

WST795/STK795

Department of Statistics, University of Pretoria

2017

Booklet compiled by IN Fabris-Rotelli

Multilevel analysis of educational data

Shin-Yan Chen 14039606

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr R. Ehlers

Department of Statistics, University of Pretoria

30th October 2017

Abstract

This research report will cover the theory and an application of the two-level multilevel statistical model where the dependent variable is continuous. The hierarchical structure of educational data where learners are nested within schools results in correlated data that must be accommodated in the analysis. The data from TIMSS (Trends in International Mathematics and Science Study) 2015 will be analysed with SAS/PROC MIXED software, using a multilevel model to determine factors that are significantly related to the mathematics achievement of Grade 9 learners in South Africa.

Declaration

I, *Shin-Yan Chen,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Shin-Yan Chen

_ _

Dr R. Ehlers

_ _

Date

Contents

List of Figures

List of Tables

1 Introduction

Linear mixed models (LMM) are an extension of linear models, but allow for dependency in the observed data [1], [11]. This paper will focus on the theory and understanding behind the two-level multilevel analysis, which is a special case of the LMM. The model, covariance structures, estimation procedures and model diagnostics will be discussed.

As a simple example of a two-level multilevel model, consider the case where y_{ij} is the achievement of learner j in school i, $j = 1, 2, \ldots\ n_i$ and $i = 1, 2, \ldots\ m$. Also suppose that there is one explanatory variable, x, on learner level (level-1), and one, z, on school level (level-2).

From [1] and [11], an example of a two-level multilevel model written on the two levels is:

LEVEL-1

$$y_{ij} = \beta_{0i} + \beta_{1i}x_{ij} + e_{ij}$$

where $e_{ij} \sim N(0, \sigma^2)$, all independent;

LEVEL-2

$$\beta_{0i} = \gamma_{00} + \gamma_{01}z_i + u_{0i}$$

$$\beta_{1i} = \gamma_{10} + \gamma_{11}z_i + u_{1i}$$

where $\begin{pmatrix} u_{0i} \\ u_{1i} \end{pmatrix} \sim N_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{01} & \tau_{11} \end{pmatrix}\right)$ independent for all $i = 1, ..., m$, and independent of e_{ij}.

This model allows for randomness in both the intercept and slope of level-1 parameters, as well as a correlation between the intercept and the slope. The combined model is

$$y_{ij} = \gamma_{00} + \gamma_{01}z_i + \gamma_{10}x_{ij} + \gamma_{11}z_ix_{ij} + u_{0i} + u_{1i}x_{ij} + e_{ij}$$

where $\gamma_{00} + \gamma_{01}z_i + \gamma_{10}x_{ij} + \gamma_{11}z_ix_{ij}$ is the fixed part of the model and $u_{0i} + u_{1i}x_{ij} + e_{ij}$ is the random part. The fixed parameters are γ_{00}, γ_{01} ,γ_{10} and γ_{11} and the random parameters are σ^2, τ_{00}, τ_{01} and τ_{11}.

Assuming that $\tau_{01} = 0$, then $var(y_{ij}) = \tau_{00} + x_{ij}^2\tau_{11} + \sigma^2$ and $cov(y_{ij}, y_{i'j}) = \tau_{00} + x_{ij}x_{i'j}\tau_{11}$. From the latter expression, one can see that this model allows for the dependence between learners within the same school.

In the second part of the paper, the theory of multilevel models will be applied to the TIMSS 2015 data set where learners are nested within schools. The fixed effects and covariances of the random factors will be estimated and the results will be discussed.

2 Literature Review

The essential difference between linear mixed models (LMMs) and fixed effect models is that LMMs can be used to model data when there is dependency between the observations [1], [6]. These models (LMMs) are a more appropriate choice for situations where the data set is of a hierarchical or clustered nature, due to the model accounting for dependency in the data. Covariance structures can be modeled by using LMMs, which will provide more appropriate fixed-effect estimates and standard errors [1]. As a result, LMMs are commonly used for research in the social and medical sciences, where dependency between subjects is observed.

Within the model, there are different parameters associated with the fixed and random factors [11]. Fixed-effect parameters are associated with the fixed factors, and cover all the possible levels of the covariates within the study. Random-effect parameters are associated with the random factors - the levels are considered to be sampled randomly from a population of possible levels. They represent the random deviations from the relationships with the fixed-effects. LMMs involve both fixed-effect parameters and random effects [11].

The data set for LMMs can be either clustered, longitudinal or repeated-measure data [11]. West [11] further defines these types of data, where clustered data is where the subjects have been grouped into clusters, and dependency arises from being in the same cluster. In longitudinal data, the same subjects are studied repeatedly over a period of time, but under different conditions. Repeated-measure data occurs when different measurements are made on the same unit. In the case of longitudinal and repeated-measure data, the dependency is a result of the subjects remaining unchanged.

LMMs are also known as hierarchical linear models (HLMs), a term made popular by Raudenbush in the 1980s [9]. Different software packages can be used to fit these models. Singer [9] describes the HLM software which is specifically designed for analysis of LMMs in a hierarchical manner, which was developed by Raudenbush. With the HLM software, each level of the model is specified individually. Goldstein developed a separate software called MLwiN, with which the user could express the model in a single equation instead [9]. In this paper, the HLM software, SAS/PROC MIXED and R software will be used, and the results for the different software will be compared.

Multilevel models (MLM) are a special case of LMMs. In this study, the focus will be on two-level multilevel models for clustered data. In the case of a two-level model, the model is split into two levels [11]. The first level, level-1, consists of the subjects with a greater level of observation. The next level, level-2, consists of the clusters that those subjects are nested within. Furthermore, MLMs allow for randomness in both the intercept and the slope, which is something that was not allowed for in normal linear models. Where there is randomness in the slope, the model is called a random coefficients model [1].

The Trends in International Mathematics and Science Study (TIMSS) is an international valuation performed on learners in grades 4 and 8 (Grade 9 in South Africa), with the purpose of evaluating their knowledge of mathematics and science [7]. This study will focus only on the results for mathematics of Grade 9 learners. In South Africa, TIMSS is hosted by the Human Sciences Research Council (HSRC). The sampling design is a three-stage stratified cluster design with the following stages:

1. Stratifying and selecting schools from all possible schools in the country;
2. Random selection of a mathematics and science class from each school;
3. Sampling learners within a class if the class size exceeds 40, otherwise including all learners.

The TIMSS 2015 data set consists of 292 schools and 12 514 Grade 9 learners. The achievement score for South Africa was 372, with a standard error of 4.5, which is well below the international centre point of 500 [7]. Since learners are nested within each school cluster, the results of the learners within schools will be correlated, and we can apply the multilevel analysis to account for the dependency [6].

3 Two-level Multilevel Models for Clustered Data

3.1 The model, notation and underlying assumptions

In this section, different two-level multilevel models will be considered to explain the model, notation and underlying assumptions. The models that will be considered are:

1. Null model - intercept only
2. Random intercept with one level-1 covariate
3. Random intercept with one level-1 covariate and one level-2 covariate
4. Random slope with one level-1 covariate
5. Random slope with one level-1 covariate and one level-2 covariate (no interaction)
6. Random slope with one level-1 covariate and one level-2 covariate (interaction)

In all the models, y_{ij} is the response for unit j (level-1) in group i (level-2). The covariates on level-1 and level-2 are respectively indicated by x_{ij} and z_i. Independence between the error terms on level-1 (e_{ij}) and level-2 (u_{0i} and u_{1i}) are assumed.

Each model will be written as an expression on both level-1 and level-2, as well as the combined model. The random parts in the combined model will be indicated in parentheses.

1. **Model 1** - *Null model*

Level-1	$y_{ij} = \beta_{0i} + e_{ij}$	$e_{ij} \sim N(0, \sigma^2)$
Level-2	$\beta_{0i} = \gamma_{00} + u_{0i}$	$u_{0i} \sim N(0,\, \tau_{00})$
Combined model	$y_{ij} = \gamma_{00} + (u_{0i} + e_{ij})$	

2. **Model 2** - *Random intercept with one level-1 covariate*

Level-1	$y_{ij} = \beta_{0i} + \gamma_{10} x_{ij} + e_{ij}$	$e_{ij} \sim N(0, \sigma^2)$
Level-2	$\beta_{0i} = \gamma_{00} + u_{0i}$	$u_{0i} \sim N(0,\, \tau_{00})$
Combined model	$y_{ij} = \gamma_{00} + \gamma_{10} x_{ij} + (u_{0i} + e_{ij})$	

3. **Model 3** - *Random intercept with one level-1 and one level-2 covariate*

Level-1	$y_{ij} = \beta_{0i} + \gamma_{10} x_{ij} + e_{ij}$	$e_{ij} \sim N(0, \sigma^2)$
Level-2	$\beta_{0i} = \gamma_{00} + \gamma_{01} z_i + u_{0i}$	$u_{0i} \sim N(0,\, \tau_{00})$
Combined model	$y_{ij} = \gamma_{00} + \gamma_{01} z_i + \gamma_{10} x_{ij} + (u_{0i} + e_{ij})$	

4. **Model 4** - *Random slope with one level-1 covariate*

Level-1	$y_{ij} = \beta_{0i} + \beta_{1i} x_{ij} + e_{ij}$	$e_{ij} \sim N(0, \sigma^2)$
Level-2	$\beta_{0i} = \gamma_{00} + u_{0i}$	
	$\beta_{1i} = \gamma_{10} + u_{1i}$	$\begin{pmatrix} u_{0i} \\ u_{1i} \end{pmatrix} \sim N_2 \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{01} & \tau_{11} \end{pmatrix} \right)$
Combined model	$y_{ij} = \gamma_{00} + \gamma_{10} x_{ij} + (u_{0i} + u_{1i} x_{ij} + e_{ij})$	

5. **Model 5** - *Random slope with one level-1 and one level-2 covariate (no interaction)*

Level-1	$y_{ij} = \beta_{0i} + \beta_{1i} x_{ij} + e_{ij}$	$e_{ij} \sim N(0, \sigma^2)$
Level-2	$\beta_{0i} = \gamma_{00} + \gamma_{01} z_i + u_{0i}$	
	$\beta_{1i} = \gamma_{10} + u_{1i}$	$\begin{pmatrix} u_{0i} \\ u_{1i} \end{pmatrix} \sim N_2 \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{01} & \tau_{11} \end{pmatrix} \right)$
Combined model	$y_{ij} = \gamma_{00} + \gamma_{01} z_i + \gamma_{10} x_{ij} + (u_{0i} + u_{1i} x_{ij} + e_{ij})$	

6. **Model 6** - *Random slope with one level-1 and one level-2 covariate (interaction)*

Level-1	$y_{ij} = \beta_{0i} + \beta_{1i} x_{ij} + e_{ij}$	$e_{ij} \sim N(0, \sigma^2)$
Level-2	$\beta_{0i} = \gamma_{00} + \gamma_{01} z_i + u_{0i}$	
	$\beta_{1i} = \gamma_{10} + \gamma_{11} z_i + u_{1i}$	$\begin{pmatrix} u_{0i} \\ u_{1i} \end{pmatrix} \sim N_2 \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{01} & \tau_{11} \end{pmatrix} \right)$
Combined model	$y_{ij} = \gamma_{00} + \gamma_{01} z_i + \gamma_{10} x_{ij} + \gamma_{11} x_{ij} z_i + (u_{0i} + u_{1i} x_{ij} + e_{ij})$	

An important measure indicating the need of a multilevel model due to the clustering of the data into a hierarchical structure is the intraclass correlation coefficient (ICC) ([1], [11]), also known as the variance partitioning coefficient (VPC). The ICC gives the proportion of variation in the model due to clustering,

and is calculated as the ratio of group-level variation of error divided by the total variation of error, as follows:

$$ICC = \rho = \frac{\sigma_u^2}{\sigma_u^2 + \sigma_e^2}$$

where σ_u^2 is the variance of the residuals on level-2 and σ_e^2 is the variance of the residuals on level-1 ([10], [11]).

The ICC value gives an indication of the level of homogeneity for the dependent variables in level-1 ([10], [11]), and its value will approach 0 when units within each group are independent i.e. when σ_u^2 is low. If the ICC for the null model is high, it implies that there is significant dependency between units within each group, which justifies the use of a multilevel model. The ICC for the different models above will be discussed in Section 6 where simulated data will be used to explain the effects that covariates on level-1 and level-2 have on this measure.

The next few sections will illustrate the notations in the models in more detail. Due to the similarity of the random part of models 1 to 3 and models 4 to 6, only models 3 and 5 will be discussed. The rest of the models follow from these in a similar fashion.

3.1.1 Model 3 - Random intercept with one level-1 and one level-2 covariate

This model has randomness in the intercept only, and has one fixed level-1 covariate and one level-2 covariate.

The combined model is written as

$$y_{ij} = \gamma_{00} + \gamma_{01} z_i + \gamma_{10} x_{ij} + (u_{0i} + e_{ij})$$

where i indicates the school and j indicates learner. Furthermore, $e_{ij} \sim N(0, \sigma^2)$, independent for all i, j, and $u_{0i} \sim N(0, \tau_{00})$, independent for all i, with $cov(e_{ij}, u_{0i}) = 0$.

The fixed part of the model is $\gamma_{00} + \gamma_{01} z_i + \gamma_{10} x_{ij}$, with fixed parameters γ_{00}, γ_{01} and γ_{10}. The random part of the model is $u_{0i} + e_{ij}$, with random parameters $var(e_{ij}) = \sigma^2$ and $var(u_{0i}) = \tau_{00}$.

To illustrate how the model can be written in matrix notation, consider the case where we have 3 schools, each with 2 learners i.e. $i = 1, 2, 3$ and $j = 1, 2$. In terms of each school $i = 1, 2, 3$:

$$
\begin{aligned}
\boldsymbol{y}_i &= \begin{pmatrix} y_{i1} \\ y_{i2} \end{pmatrix} \\
&= \begin{pmatrix} 1 & z_i & x_{i1} \\ 1 & z_i & x_{i2} \end{pmatrix} \begin{pmatrix} \gamma_{00} \\ \gamma_{01} \\ \gamma_{10} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u_{0i} + \begin{pmatrix} e_{i1} \\ e_{i2} \end{pmatrix} \\
&= \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i u_{0i} + \boldsymbol{e}_i
\end{aligned}
$$

where $u_{0i} \sim N(0,\, \tau_{00})$ and $\boldsymbol{e}_i \sim N_2\left(\boldsymbol{0},\, \sigma^2\boldsymbol{I}_2\right)$.

Since $E(\boldsymbol{y}_i) = \boldsymbol{X}_i\boldsymbol{\beta}$ and

$$
\begin{aligned}
cov(\boldsymbol{y}_i,\, \boldsymbol{y}_i^{'}) &= cov((\boldsymbol{Z}_i u_{0i} + \boldsymbol{e}_i),\, (\boldsymbol{Z}_i u_{0i} + \boldsymbol{e}_i)^{'}) \\
&= \boldsymbol{Z}_i cov(u_{0i},\, u_{0i}^{'})\boldsymbol{Z}_i^{'} + cov(\boldsymbol{e}_i,\, \boldsymbol{e}_i^{'}) \\
&= \tau_{00}\boldsymbol{Z}_i\boldsymbol{Z}_i^{'} + \sigma^2\boldsymbol{I}_2 \\
&= \begin{pmatrix} \tau_{00} + \sigma^2 & \tau_{00} \\ \tau_{00} & \tau_{00} + \sigma^2 \end{pmatrix} \\
&= \boldsymbol{V}_i
\end{aligned}
$$

it follows that

$$
\boldsymbol{y}_i \sim N(\boldsymbol{X}_i\boldsymbol{\beta},\, \boldsymbol{V}_i). \tag{1}
$$

The ICC of the model is $ICC = \frac{\tau_{00}}{\tau_{00}+\sigma^2}$, which is also an indication of the correlation between learners within the same school, since

$$
\begin{aligned}
corr(y_{i1}, y_{i2}) &= \frac{cov(y_{i1}, y_{i2})}{\sqrt{var(y_{i1})var(y_{i2})}} \\
&= \frac{\tau_{00}}{\tau_{00} + \sigma^2} \\
&= ICC.
\end{aligned}
$$

The model for all three schools can be rewritten in matrix form as:

$$
\begin{aligned}
\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \end{pmatrix} &= \begin{pmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \\ y_{31} \\ y_{32} \end{pmatrix} \\
&= \begin{pmatrix} 1 & z_1 & x_{11} \\ 1 & z_1 & x_{12} \\ 1 & z_2 & x_{21} \\ 1 & z_2 & x_{22} \\ 1 & z_3 & x_{31} \\ 1 & z_3 & x_{32} \end{pmatrix} \begin{pmatrix} \gamma_{00} \\ \gamma_{01} \\ \gamma_{10} \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_{01} \\ u_{02} \\ u_{03} \end{pmatrix} + \begin{pmatrix} e_{11} \\ e_{12} \\ e_{21} \\ e_{22} \\ e_{31} \\ e_{32} \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \boldsymbol{X}_3 \end{pmatrix} \boldsymbol{\beta} + \begin{pmatrix} \boldsymbol{Z}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Z}_3 \end{pmatrix} \begin{pmatrix} u_{01} \\ u_{02} \\ u_{03} \end{pmatrix} + \begin{pmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \boldsymbol{e}_3 \end{pmatrix} \\
&= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{e}
\end{aligned}
$$

where $\boldsymbol{u} \sim N_3(\boldsymbol{0},\, \boldsymbol{G})$ and $\boldsymbol{e} \sim N_6\,(\boldsymbol{0},\, \boldsymbol{R})$, where $\boldsymbol{G} = \tau_{00}\boldsymbol{I}_3$ and $\boldsymbol{R} = \sigma^2\boldsymbol{I}_6$.

The elements of the vector $\boldsymbol{\beta}$ give the fixed-effect parameters that need to be estimated.

The random effect part of the model is $\boldsymbol{Z}\boldsymbol{u} + \boldsymbol{e}$, and the variance components that must be estimated are τ_{00} and σ^2.

Since $E(\boldsymbol{y}) = \boldsymbol{X}\boldsymbol{\beta}$ and

$$
\begin{aligned}
cov(\boldsymbol{y},\, \boldsymbol{y}^{'}) &= cov(\boldsymbol{Z}\boldsymbol{u} + \boldsymbol{e},\, (\boldsymbol{Z}\boldsymbol{u} + \boldsymbol{e})^{'}) \\
&= \boldsymbol{Z}cov(\boldsymbol{u},\, \boldsymbol{u}^{'})\boldsymbol{Z}^{'} + cov(\boldsymbol{e},\, \boldsymbol{e}^{'}) \\
&= \boldsymbol{Z}\boldsymbol{G}\boldsymbol{Z}^{'} + \boldsymbol{R}
\end{aligned}
$$

it follows that

$$
\boldsymbol{y} \sim N(\boldsymbol{X}\boldsymbol{\beta},\, \boldsymbol{Z}\boldsymbol{G}\boldsymbol{Z}^{'} + \boldsymbol{R})
$$

where

$$cov(\boldsymbol{y},\, \boldsymbol{y}^{'}) = \begin{pmatrix} \tau_{00}+\sigma^2 & \tau_{00} & 0 & 0 & 0 & 0 \\ \tau_{00} & \tau_{00}+\sigma^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \tau_{00}+\sigma^2 & \tau_{00} & 0 & 0 \\ 0 & 0 & \tau_{00} & \tau_{00}+\sigma^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \tau_{00}+\sigma^2 & \tau_{00} \\ 0 & 0 & 0 & 0 & \tau_{00} & \tau_{00}+\sigma^2 \end{pmatrix}.$$

Therefore, the full model has the distribution $\boldsymbol{y} \sim N(\boldsymbol{X\beta},\, \boldsymbol{V})$, where $\boldsymbol{V} = \boldsymbol{ZGZ}^{'} + \boldsymbol{R} = \begin{pmatrix} \boldsymbol{V}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{V}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{V}_3 \end{pmatrix}$,

where $\boldsymbol{V}_i = \tau_{00}\boldsymbol{Z}_i\boldsymbol{Z}_i^{'} + \sigma^2\boldsymbol{I}_2$ for $i = 1,\, 2,\, 3$.

3.1.2 Model 5 - Random slope with one Level-1 and one Level-2 covariate (no interaction)

We now consider the case where both the intercept and the slope are random, also called a random slopes model. In this specific model, the level-2 covariate is only associated with the intercept, and the combined model, with i indicating the school and j indicating learner, is

$$y_{ij} = \gamma_{00} + \gamma_{01}z_i + \gamma_{10}x_{ij} + (u_{0i} + u_{1i}x_{ij} + e_{ij})$$

with $e_{ij} \sim N(0,\, \sigma^2)$, all independent, $\boldsymbol{u}_i = \begin{pmatrix} u_{0i} \\ u_{1i} \end{pmatrix} \sim N_2\,(\boldsymbol{0},\, \boldsymbol{D})$ with $\boldsymbol{D} = \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{01} & \tau_{11} \end{pmatrix}$ independent for all $i = 1,\, 2,\, 3$, and independent of e_{ij} i.e. $cov(e_{ij},\, u_{0i}) = 0$.

The fixed part of the model is $\gamma_{00}+\gamma_{01}z_i+\gamma_{10}x_{ij}$ with fixed parameters γ_{00}, γ_{01} and γ_{10}. Furthermore, $u_{0i} + u_{1i}x_{ij} + e_{ij}$ is the random part of the model, with random parameters σ^2, τ_{00}, τ_{01} and τ_{11}.

Consider again the case where we have 3 schools, each with 2 learners i.e. $i = 1,\, 2,\, 3$ and $j = 1,\, 2$. This model can be written in a matrix form in terms of each school $i = 1,\, 2,\, 3$:

$$\begin{aligned} \boldsymbol{y}_i &= \begin{pmatrix} y_{i1} \\ y_{i2} \end{pmatrix} \\ &= \begin{pmatrix} 1 & z_i & x_{i1} \\ 1 & z_i & x_{i2} \end{pmatrix} \begin{pmatrix} \gamma_{00} \\ \gamma_{01} \\ \gamma_{10} \end{pmatrix} + \begin{pmatrix} 1 & x_{i1} \\ 1 & x_{i2} \end{pmatrix} \begin{pmatrix} u_{0i} \\ u_{1i} \end{pmatrix} + \begin{pmatrix} e_{i1} \\ e_{i2} \end{pmatrix} \end{aligned}$$

$$= \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{u}_i + \boldsymbol{e}_i$$

where $\boldsymbol{u}_i \sim N_2\left(\boldsymbol{0},\, \boldsymbol{D}\right)$ and $\boldsymbol{e}_i \sim N_2\left(\boldsymbol{0},\, \sigma^2\boldsymbol{I}_2\right)$.

Since $E(\boldsymbol{y}_i) = \boldsymbol{X}_i\boldsymbol{\beta}$ and

$$\begin{aligned}
cov(\boldsymbol{y}_i,\, \boldsymbol{y}_i^{'}) &= cov((\boldsymbol{Z}_i\boldsymbol{u}_i + \boldsymbol{e}_i),\, (\boldsymbol{Z}_i\boldsymbol{u}_i + \boldsymbol{e}_i)^{'}) \\
&= \boldsymbol{Z}_i var(\boldsymbol{u}_i,\, \boldsymbol{u}_i^{'})\boldsymbol{Z}_i^{'} + var(\boldsymbol{e}_i,\, \boldsymbol{e}_i^{'}) \\
&= \boldsymbol{Z}_i\boldsymbol{D}\boldsymbol{Z}_i^{'} + \sigma^2\boldsymbol{I}_2 \\
&= \left(\begin{array}{c|c} \tau_{00} + 2\tau_{01}x_{i1} + \tau_{11}x_{i1}^2 + \sigma^2 & \tau_{00} + \tau_{01}(x_{i1} + x_{i2}) + \tau_{11}x_{i1}x_{i2} \\ \hline \tau_{00} + \tau_{01}(x_{i1} + x_{i2}) + \tau_{11}x_{i1}x_{i2} & \tau_{00} + 2\tau_{01}x_{i2} + \tau_{11}x_{i2}^2 + \sigma^2 \end{array}\right) \\
&= \boldsymbol{V}_i
\end{aligned}$$

it follows that

$$\boldsymbol{y}_i \sim N(\boldsymbol{X}_i\boldsymbol{\beta},\, \boldsymbol{V}_i). \tag{2}$$

The matrix form of the full model, with $i = 3$ schools and $j = 2$ learners, is written as:

$$\begin{aligned}
\boldsymbol{y} &= \begin{pmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \\ y_{31} \\ y_{32} \end{pmatrix} \\
&= \begin{pmatrix} 1 & z_1 & x_{11} \\ 1 & z_1 & x_{12} \\ 1 & z_2 & x_{21} \\ 1 & z_2 & x_{22} \\ 1 & z_3 & x_{31} \\ 1 & z_3 & x_{32} \end{pmatrix} \begin{pmatrix} \gamma_{00} \\ \gamma_{01} \\ \gamma_{10} \end{pmatrix} + \begin{pmatrix} 1 & x_{11} & 0 & 0 & 0 & 0 \\ 1 & x_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & x_{21} & 0 & 0 \\ 0 & 0 & 1 & x_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_{31} \\ 0 & 0 & 0 & 0 & 1 & x_{32} \end{pmatrix} \begin{pmatrix} u_{01} \\ u_{11} \\ u_{02} \\ u_{12} \\ u_{03} \\ u_{13} \end{pmatrix} + \begin{pmatrix} e_{11} \\ e_{12} \\ e_{21} \\ e_{22} \\ e_{31} \\ e_{32} \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \boldsymbol{X}_3 \end{pmatrix} \boldsymbol{\beta} + \begin{pmatrix} \boldsymbol{Z}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Z}_3 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \end{pmatrix} + \begin{pmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \boldsymbol{e}_3 \end{pmatrix} \\
&= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{e}
\end{aligned}$$

where $\boldsymbol{u} \sim N_6(\boldsymbol{0},\, \boldsymbol{G})$ and $\boldsymbol{e} \sim N_6\,(\boldsymbol{0},\, \boldsymbol{R})$, where $\boldsymbol{G} = \boldsymbol{I}_3 \otimes \boldsymbol{D}$ and $\boldsymbol{R} = \sigma^2 \boldsymbol{I}_6$.

The elements of the vector $\boldsymbol{\beta}$ give the fixed-effect parameters that need to be estimated.

The random effect part of the model is $\boldsymbol{Zu}+\boldsymbol{e}$, and the covariance components that must be estimated are τ_{00}, τ_{01}, τ_{11} and σ^2.

Since $E(\boldsymbol{y}) = \boldsymbol{X\beta}$ and

$$\begin{aligned} cov(\boldsymbol{y},\, \boldsymbol{y}^{'}) &= cov(\boldsymbol{Zu} + \boldsymbol{e},\, (\boldsymbol{Zu} + \boldsymbol{e})^{'}) \\ &= \boldsymbol{Z}cov(\boldsymbol{u},\, \boldsymbol{u}^{'})\boldsymbol{Z}^{'} + cov(\boldsymbol{e},\, \boldsymbol{e}^{'}) \end{aligned}$$

it follows that

$$\boldsymbol{y} \sim N(\boldsymbol{X\beta},\, \boldsymbol{ZGZ}^{'} + \boldsymbol{R})$$

where $\boldsymbol{G} = cov(\boldsymbol{u},\, \boldsymbol{u}^{'})$ and $\boldsymbol{R} = cov(\boldsymbol{e},\, \boldsymbol{e}^{'})$.

Therefore, the full model has the distribution $\boldsymbol{y} \sim N(\boldsymbol{X\beta},\, \boldsymbol{V})$ where $\boldsymbol{V} = \boldsymbol{ZGZ}^{'} + \boldsymbol{R} = \begin{pmatrix} \boldsymbol{V}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{V}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{V}_3 \end{pmatrix}$,

where $\boldsymbol{V}_i = \boldsymbol{Z}_i \boldsymbol{D} \boldsymbol{Z}_i^{'} + \sigma^2 \boldsymbol{I}_2$.

3.1.3 The general case

For the general case of $i = 1, 2, ..., m$ schools with $j = 1, 2, ..., n_i$ learners in each school, the general matrix form for each school i, with p fixed effects parameters and q random effects, could be expressed as:

$$\boldsymbol{y}_i = \boldsymbol{X}_i \boldsymbol{\beta} + \boldsymbol{Z}_i \boldsymbol{u}_i + \boldsymbol{e}_i \tag{3}$$

with design matrices $\boldsymbol{X}_i : n_i \times p$ and $\boldsymbol{Z}_i : n_i \times q$. Furthermore, the random effects and residual vectors are $\boldsymbol{u}_i : q \times 1 = \begin{pmatrix} u_{0i} \\ u_{1i} \\ \vdots \\ u_{q-1,\, i} \end{pmatrix} \sim N(\boldsymbol{0},\, \boldsymbol{D})$ and $\boldsymbol{e}_i : n_i \times 1 = \begin{pmatrix} e_{i1} \\ e_{i2} \\ \vdots \\ e_{i,n_i} \end{pmatrix} \sim N(\boldsymbol{0}, \boldsymbol{R}_i)$ with $\boldsymbol{R}_i = \sigma^2 \boldsymbol{I}_{n_i}$.
It is assumed that the residuals are all independent of each other, as well as independent of the random effects $\boldsymbol{u}_1, ..., \boldsymbol{u}_m$.

In (3), $\boldsymbol{y}_i$ is a $n_i \times 1$ vector of the achievement for each learner $j = 1, ..., n_i$ in school $i = 1, ..., m$.

$$\boldsymbol{y}_i = \begin{pmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{i,n_i} \end{pmatrix}$$

Since $E(\boldsymbol{y}_i) = \boldsymbol{X}_i\boldsymbol{\beta}$ and from the independence of the $\boldsymbol{u}_i$'s and $\boldsymbol{e}_i$'s,

$$\begin{aligned} cov(\boldsymbol{y}_i, \boldsymbol{y}_i^{'}) &= cov((\boldsymbol{Z}_i\boldsymbol{u}_i + \boldsymbol{e}_i), (\boldsymbol{Z}_i\boldsymbol{u}_i + \boldsymbol{e}_i)^{'}) \\ &= \boldsymbol{Z}_i var(\boldsymbol{u}_i, \boldsymbol{u}_i^{'})\boldsymbol{Z}_i^{'} + var(\boldsymbol{e}_i, \boldsymbol{e}_i^{'}) \\ &= \boldsymbol{Z}_i\boldsymbol{D}\boldsymbol{Z}_i^{'} + \boldsymbol{R}_i \\ &= \boldsymbol{V}_i \end{aligned}$$

it follows that

$$\boldsymbol{y}_i \sim N(\boldsymbol{X}_i\boldsymbol{\beta}, \boldsymbol{V}_i). \tag{4}$$

Alternatively, the model representing all $j = 1, 2, ..., n_i$ learners and $i = 1, 2, ..., m$ schools is:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{e} \tag{5}$$

$$\begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_m \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \vdots \\ \boldsymbol{X}_m \end{pmatrix} \boldsymbol{\beta} + \begin{pmatrix} \boldsymbol{Z}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{Z}_m \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_m \end{pmatrix} + \begin{pmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_m \end{pmatrix}.$$

The non-diagonal values in the $\boldsymbol{Z}$ matrix are $\boldsymbol{0}$ due to the independence that exists between schools (level-2).

In the above model, $\boldsymbol{u} \sim N(\boldsymbol{0}, \boldsymbol{G})$, where $\boldsymbol{G} = cov(\boldsymbol{u}, \boldsymbol{u}^{'}) = \boldsymbol{I}_m \otimes \boldsymbol{D}$, and $\boldsymbol{e} \sim N(\boldsymbol{0}, \boldsymbol{R})$, where $\boldsymbol{R} = \sigma^2\boldsymbol{I}_n$ with $n = \sum_i n_i$. It is assumed that the residuals are all independent of each other, as well as independent of the random effects $\boldsymbol{u}_1, ..., \boldsymbol{u}_m$.

Note that within each school, we had that $\boldsymbol{u}_i \sim N(\boldsymbol{0}, \boldsymbol{D})$ and $\boldsymbol{e}_i \sim N(\boldsymbol{0}, \boldsymbol{R}_i)$. In the complete case, $\boldsymbol{u} \sim N(\boldsymbol{0}, \boldsymbol{G})$ and $\boldsymbol{e} \sim N(\boldsymbol{0}, \boldsymbol{R})$, where the covariance matrices for the full model $\boldsymbol{G}$ and $\boldsymbol{R}$ consist of stacking the covariance matrices for each school, $\boldsymbol{D}$ and $\boldsymbol{R}_i$ respectively, along the diagonal.

Since $E(\boldsymbol{y}) = \boldsymbol{X\beta}$ and from the independence of $\boldsymbol{u}$ and $\boldsymbol{e}$,

$$\begin{aligned}
cov(\boldsymbol{y}, \boldsymbol{y}^{'}) &= cov((\boldsymbol{Zu} + \boldsymbol{e}), (\boldsymbol{Zu} + \boldsymbol{e})^{'}) \\
&= \boldsymbol{Z}_i var(\boldsymbol{u}, \boldsymbol{u}^{'}) \boldsymbol{Z}_i^{'} + var(\boldsymbol{e}, \boldsymbol{e}^{'}) \\
&= \boldsymbol{ZGZ}^{'} + \boldsymbol{R} \\
&= \boldsymbol{Z} \begin{pmatrix} \boldsymbol{D} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{D} \end{pmatrix} \boldsymbol{Z}^{'} + \begin{pmatrix} \boldsymbol{R}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{R}_m \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{Z}_1 \boldsymbol{D} \boldsymbol{Z}_1^{'} + \boldsymbol{R}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{Z}_m \boldsymbol{D} \boldsymbol{Z}_m^{'} + \boldsymbol{R}_m \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{V}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{V}_m \end{pmatrix} \\
&= \boldsymbol{V}
\end{aligned}$$

it follows that

$$\boldsymbol{y} \sim N(\boldsymbol{X\beta}, \boldsymbol{V}). \tag{6}$$

4 Parameter estimation

Consider the two-level multilevel model given in (3),

$$\boldsymbol{y}_i = \boldsymbol{X}_i \boldsymbol{\beta} + \boldsymbol{Z}_i \boldsymbol{u}_i + \boldsymbol{e}_i.$$

The parameters that need to be estimated are the fixed-effect parameters given in $\boldsymbol{\beta}$, and the covariance parameters for both the random effects and the residuals, that is the elements of $\boldsymbol{D} = cov(\boldsymbol{u}_i, \boldsymbol{u}_i^{'})$ and $var(e_{ij}) = \sigma^2$. Two common methods that are used to estimate these parameters are maximum likelihood (ML) estimation and restricted/residual maximum likelihood (REML) estimation [11]. The main difference between the two methods is that ML estimation results in bias (usually downwards) of the estimates for the covariance parameters ([1], [11]). The reason for this is that ML estimation does not take into consideration the loss of degrees of freedom that occurs when estimating the fixed-effect parameters $\boldsymbol{\beta}$. As a result, REML is generally preferred, as it takes this the loss of degrees of freedom into account and hence results in unbiasedness of the covariance estimates. ML estimation will be discussed

in Section 4.1 and REML estimation will be discussed further in Section 4.2.

Throughout this section, the notation $\boldsymbol{\theta}$ will be used to represent the covariance parameters of the model. In other words, $\boldsymbol{\theta}$ consists of a vector of the residual variance parameter σ^2, followed by the covariance parameters for the random effects in $\boldsymbol{D}$, i.e. $\boldsymbol{\theta} = \begin{pmatrix} \sigma^2 \\ \tau_{00} \\ \tau_{01} \\ \vdots \\ \tau_{q-1,\,q-1} \end{pmatrix}$ (see Section 3.1.3 for the description of the model).

4.1 Maximum likelihood (ML) estimation

The maximum likelihood (ML) method of estimation requires the use of a likelihood function, which is constructed using the parameters that need to be estimated in the model [11]. The function is also based on the assumptions regarding the distributions or density functions of those parameters. The parameters are then estimated by maximising this likelihood function and solving for each parameter. The resulting values are then called the maximum likelihood estimates (MLEs) of the model.

For a multilevel model, the parameters that need to be estimated are the fixed-effect parameters in $\boldsymbol{\beta}$ and the covariance parameters given in $\boldsymbol{\theta}$. In Section 3.1.3, it is shown that

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{u}_i + \boldsymbol{e}_i \sim N(\boldsymbol{X}_i\boldsymbol{\beta},\, \boldsymbol{V}_i).$$

The marginal linear model is specified by excluding the random effect from the model and is expressed as $\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{\varepsilon}_i$ where $\boldsymbol{\varepsilon}_i \sim N(\boldsymbol{0},\, \boldsymbol{V}_i)$. As a result, it follows that $\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{\varepsilon}_i \sim N(\boldsymbol{X}_i\boldsymbol{\beta},\, \boldsymbol{V}_i)$. From the distribution of the vector $\boldsymbol{y}_i$ in the marginal model it follows:

$$f(\boldsymbol{y}_i|\boldsymbol{\beta},\, \boldsymbol{\theta}) = (2\pi)^{-\frac{n_i}{2}} \times det(\boldsymbol{V}_i)^{-\frac{1}{2}} \times exp(-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})'\boldsymbol{V}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}))$$

and the likelihood function is:

$$L_i(\boldsymbol{\beta},\, \boldsymbol{\theta}) = (2\pi)^{-\frac{n_i}{2}} \times det(\boldsymbol{V}_i)^{-\frac{1}{2}} \times exp(-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})'\boldsymbol{V}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}))$$

where $\boldsymbol{y}_i$ is now the observed values of the vector $\boldsymbol{y}_i$.

Since the schools in the model are assumed to be independent, the likelihood function will be the product of of the m independent likelihood functions for each school [1], as follows:

$$
\begin{aligned}
L(\boldsymbol{\beta}, \boldsymbol{\theta}) &= \prod_i L_i(\boldsymbol{\beta}, \boldsymbol{\theta}) \\
&= \prod_i (2\pi)^{-\frac{n_i}{2}} \times det(\boldsymbol{V}_i)^{-\frac{1}{2}} \times exp(-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^{'}\boldsymbol{V}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})) \\
&= (2\pi)^{-\frac{n}{2}} \times \sum_i det(\boldsymbol{V}_i)^{-\frac{1}{2}} \times exp(-\frac{1}{2}\sum_i (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^{'}\boldsymbol{V}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})) \qquad (7)
\end{aligned}
$$

where $n = \sum_{i=1}^{m} n_i$.

The log-likelihood function is taken as the natural logarithm of the likelihood function:

$$
l(\boldsymbol{\beta}, \boldsymbol{\theta}) = -\frac{n}{2}ln(2\pi) - \frac{1}{2}\sum_i ln[det(\boldsymbol{V}_i)] - \frac{1}{2}\sum_i (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^{'}\boldsymbol{V}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}). \qquad (8)
$$

Estimates of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ can be found by optimizing the log-likelihood function (8) above, by solving for each parameter simultaneously using computational algorithms [11]. An alternative method of optimization by profiling out $\boldsymbol{\beta}$ will be discussed in Sections 4.1.1 and 4.1.2 below.

4.1.1 Special case: covariance parameters known

To solve for the parameters, the special case is considered where all covariance parameters in $\boldsymbol{\theta}$ are known. Since $\boldsymbol{\theta}$ is assumed to be known, only the fixed-effect parameters $\boldsymbol{\beta}$ need to be estimated. Furthermore, the log-likelihood function in (8) can be expressed as a function of $\boldsymbol{\beta}$ only.

$$
l(\boldsymbol{\beta}) = -\frac{n}{2}ln(2\pi) - \frac{1}{2}\sum_i ln[det(\boldsymbol{V}_i)] - \frac{1}{2}\sum_i (\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^{'}\boldsymbol{V}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}). \qquad (9)
$$

The function above is optimised with respect to the fixed-effect parameters $\boldsymbol{\beta}$ by using the generalised least squares method. To obtain $\boldsymbol{\beta}$ for which $l(\boldsymbol{\beta})$ is a maximum, the derivative of the log-likelihood function is taken as follows:

$$
\begin{aligned}
\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= \sum_i \boldsymbol{X}_i^{'}\boldsymbol{V}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}) \\
&= \sum_i \boldsymbol{X}_i^{'}\boldsymbol{V}_i^{-1}\boldsymbol{y}_i - (\sum_i \boldsymbol{X}_i^{'}\boldsymbol{V}_i^{-1}\boldsymbol{X}_i)\boldsymbol{\beta}.
\end{aligned}
$$

Setting this equal to 0 and solving for $\boldsymbol{\beta}$ gives the fixed-effects parameter estimate

$$\hat{\boldsymbol{\beta}} = (\sum_i \boldsymbol{X}_i^{'} \boldsymbol{V}_i^{-1} \boldsymbol{X}_i)^{-1} \sum_i \boldsymbol{X}_i^{'} \boldsymbol{V}_i^{-1} \boldsymbol{y}_i$$
$$= (\boldsymbol{X}^{'} \boldsymbol{V}^{-1} \boldsymbol{X})^{-1} (\boldsymbol{X}^{'} \boldsymbol{V}^{-1} \boldsymbol{y}) \tag{10}$$

The ML estimate$\hat{\boldsymbol{\beta}}$ given in (10) has the properties of being the best linear unbiased estimate (BLUE) for the fixed-effects parameter $\boldsymbol{\beta}$ ([11]).

4.1.2 General case: covariance parameters unknown

In the ML estimation of the fixed-effect parameters $\boldsymbol{\beta}$ and the covariance parameters $\boldsymbol{\theta}$ that follows, the covariance parameters in $\boldsymbol{\theta}$ are now assumed to be unknown. To solve for the covariance parameters in $\boldsymbol{\theta}$, a profile log-likelihood function $l_{ML}(\boldsymbol{\theta})$ is constructed by substituting (10) into (8) ([2], [11]). The log-likelihood simplifies to a function of $\boldsymbol{\theta}$ only:

$$l_{ML}(\boldsymbol{\theta}) = -\frac{n}{2} ln(2\pi) - \frac{1}{2} \sum_i ln(det(\boldsymbol{V}_i)) - \frac{1}{2} \sum_i (\boldsymbol{y}_i - \boldsymbol{X}_i \hat{\boldsymbol{\beta}})^{'} \boldsymbol{V}_i^{-1} (\boldsymbol{y}_i - \boldsymbol{X}_i \hat{\boldsymbol{\beta}})$$
$$= -\frac{n}{2} ln(2\pi) - \frac{1}{2} \sum_i ln(det(\boldsymbol{V}_i)) - \frac{1}{2} \sum_i \boldsymbol{r}_i^{'} \boldsymbol{V}_i^{-1} \boldsymbol{r}_i \tag{11}$$

where

$$\boldsymbol{r}_i = \boldsymbol{y}_i - \boldsymbol{X}_i \hat{\boldsymbol{\beta}}$$
$$= \boldsymbol{y}_i - \boldsymbol{X}_i ((\sum_i \boldsymbol{X}_i^{'} \boldsymbol{V}_i^{-1} \boldsymbol{X}_i)^{-1} \sum_i \boldsymbol{X}_i^{'} \boldsymbol{V}_i^{-1} \boldsymbol{y}_i).$$

The function $l_{ML}(\boldsymbol{\theta})$ does not have an explicit solution for an optimized estimation of $\boldsymbol{\theta}$; therefore, the covariance parameters $\boldsymbol{\theta}$ are estimated using the Newton-Raphson algorithm of computational iteration to obtain convergence to the estimates. The MLE of $\hat{\boldsymbol{\theta}}$ is used to give estimates for the covariances of the random effects and the residuals, $\hat{\boldsymbol{D}}$ and $\hat{\boldsymbol{R}}_i$ respectively, which are then substituted to give an estimate for $\boldsymbol{V}_i$; namely $\hat{\boldsymbol{V}}_i = \boldsymbol{Z}_i \hat{\boldsymbol{D}} \boldsymbol{Z}_i^{'} + \hat{\boldsymbol{R}}_i$. Thereafter, this estimate is substituted into (10) to obtain an estimate for the fixed-effects parameters:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^{'} \hat{\boldsymbol{V}}^{-1} \boldsymbol{X})^{-1} \boldsymbol{X}^{'} \hat{\boldsymbol{V}}^{-1} \boldsymbol{y}. \tag{12}$$

Since this formula contains the estimated covariance values $\hat{\boldsymbol{V}}_i$, it is referred to as the empirical best linear unbiased estimator (EBLUE) for the fixed-effects parameter $\boldsymbol{\beta}$ [11].

4.2 Restricted/residual maximum likelihood (REML) estimation

To remove the bias of the ML parameter estimates, the likelihood function used in the restricted/residual maximum likelihood (REML) method of estimation is based on the residual $\boldsymbol{y}_i - \boldsymbol{X}_i\hat{\boldsymbol{\beta}}$ ([1], [11]).

Since $\boldsymbol{y} \sim N(\boldsymbol{X\beta}, \boldsymbol{V})$ (see Section 3.1.3) it follows from (12) that for given $\boldsymbol{V}$, the distribution for$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{y}$ is derived as follows:

$$\begin{aligned} var(\hat{\boldsymbol{\beta}}) &= var((\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{y}) \\ &= (\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{V}^{-1}var(\boldsymbol{y})\boldsymbol{V}^{-1}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1} \\ &= (\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{V}\boldsymbol{V}^{-1}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1} \\ &= (\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1} \\ &= (\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}. \end{aligned} \tag{13}$$

Similarly, its expected value is derived as:

$$\begin{aligned} E(\hat{\boldsymbol{\beta}}) &= E((\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{y}) \\ &= (\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{V}^{-1}E(\boldsymbol{y}) \\ &= (\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X}\boldsymbol{\beta} \\ &= \boldsymbol{\beta}. \end{aligned}$$

Therefore $\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, (\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1})$ [1].

Independence between $\hat{\boldsymbol{\beta}}$ and the residual $\boldsymbol{y}_i - \boldsymbol{X}_i\hat{\boldsymbol{\beta}}$ can be shown as follows:

$$\begin{aligned} cov(\hat{\boldsymbol{\beta}}, (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})') &= cov\left[(\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{y}, (\boldsymbol{y} - \boldsymbol{X}(\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{y})'\right] \\ &= cov\left[(\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{y}, \boldsymbol{y}'(\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1})'\right] \\ &= (\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}var(\boldsymbol{y}_i)\,(\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1})' \\ &= (\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{V}\,(\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1})' \\ &= (\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\,(\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{V}}^{-1})' \end{aligned}$$

$$
\begin{aligned}
&= (\boldsymbol{X}^{'}\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^{'} - (\boldsymbol{X}^{'}\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^{'}\hat{\boldsymbol{V}}^{-1}\boldsymbol{X}(\boldsymbol{X}^{'}\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^{'} \\
&= (\boldsymbol{X}^{'}\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^{'} - (\boldsymbol{X}^{'}\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^{'} \\
&= 0.
\end{aligned}
$$

Therefore, the joint likelihood of the fixed-effect parameters $\boldsymbol{\beta}$ and the covariance parameters $\boldsymbol{\theta}$ is the product of the independent likelihood functions based on $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}$. In other words,

$$
L(\boldsymbol{\beta},\, \boldsymbol{\theta};\, \boldsymbol{y}) = L(\boldsymbol{\theta};\, \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) \times L(\boldsymbol{\beta}; \hat{\boldsymbol{\beta}}, \boldsymbol{\theta}).
$$

Therefore it follows that:

$$
L(\boldsymbol{\theta};\, \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) = L(\boldsymbol{\beta},\, \boldsymbol{\theta};\, \boldsymbol{y}) \,/\, L(\boldsymbol{\beta}; \hat{\boldsymbol{\beta}}, \boldsymbol{\theta}). \tag{14}
$$

From (7), it can be seen that [1]:

$$
L(\boldsymbol{\beta},\, \boldsymbol{\theta};\, \boldsymbol{y}) = (2\pi)^{-\frac{n}{2}} \times det(\boldsymbol{V})^{-\frac{1}{2}} \times exp(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^{'}\boldsymbol{V}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})). \tag{15}
$$

Since $\hat{\boldsymbol{\beta}} \sim N_p(\boldsymbol{\beta},\, (\boldsymbol{X}^{'}\boldsymbol{V}^{-1}\boldsymbol{X})^{-1})$, it follows from [1] that:

$$
L(\boldsymbol{\beta}; \hat{\boldsymbol{\beta}}, \boldsymbol{\theta}) = (2\pi)^{-\frac{p}{2}} \times det(\boldsymbol{X}^{'}\boldsymbol{V}^{-1}\boldsymbol{X})^{-\frac{1}{2}} \times exp(-\frac{1}{2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^{'}\boldsymbol{X}^{'}\boldsymbol{V}^{-1}\boldsymbol{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})). \tag{16}
$$

Substituting (15) and (16) into (14) gives

$$
\begin{aligned}
L(\boldsymbol{\theta};\, \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) &= (2\pi)^{-\frac{n-p}{2}} \times det(\boldsymbol{X}^{'}\boldsymbol{V}^{-1}\boldsymbol{X})^{-\frac{1}{2}} \times det(\boldsymbol{V})^{-\frac{1}{2}} \times exp(-\frac{1}{2}\sum_{i}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^{'}\boldsymbol{V}^{-1}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})) \\
&= (2\pi)^{-\frac{n-p}{2}} \times det(\boldsymbol{X}^{'}\boldsymbol{V}^{-1}\boldsymbol{X})^{-\frac{1}{2}} \times det(\boldsymbol{V})^{-\frac{1}{2}} \times exp(-\frac{1}{2}\boldsymbol{r}^{'}\boldsymbol{V}^{-1}\boldsymbol{r})
\end{aligned}
$$

with a REML log-likelihood function as follows [11]:

$$
l_{REML}(\boldsymbol{\theta}) = -\frac{(n-p)}{2}\, ln(2\pi) - \frac{1}{2}ln(det(\boldsymbol{X}^{'}\boldsymbol{V}^{-1}\boldsymbol{X})) - \frac{1}{2}ln(det(\boldsymbol{V})) - \frac{1}{2}\boldsymbol{r}^{'}\boldsymbol{V}^{-1}\boldsymbol{r}
$$

where

$$
\begin{aligned}
\boldsymbol{r} &= \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}} \\
&= \boldsymbol{y} - \boldsymbol{X}((\boldsymbol{X}^{'}\hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^{'}\hat{\boldsymbol{V}}^{-1}\boldsymbol{y}).
\end{aligned}
$$

Since the function is an expression of $\boldsymbol{\theta}$ only, the covariance parameters can be solved for using an

algorithm to give the estimate $\hat{\boldsymbol{V}}$. This estimate can then be substituted into equations (12) and (13) to give the estimates for the fixed-effect parameters, $\hat{\boldsymbol{\beta}}$, and the corresponding variance, $var(\hat{\boldsymbol{\beta}})$.

5 Model diagnostics

Model diagnostics are performed to assess whether the assumptions regarding the distribution of the residuals are met, as well as to detect either outliers or potentially influential observations. Model diagnostics should be integrated into the model-building process. Two of these diagnostic methods will be discussed, namely: residual diagnostics and influence diagnostics.

5.1 Residual diagnostics

Residual diagnostics make use of different types of residuals that can be calculated. Consider the multilevel model in (3), that is $\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{u}_i + \boldsymbol{e}_i$ where $\boldsymbol{e}_i \sim N(\boldsymbol{0}, \sigma^2\boldsymbol{I}_{n_i})$. The residuals can be divided into two main types, namely marginal and conditional. The raw marginal residuals for unit i on level-2 is given by [11]

$$\hat{\boldsymbol{e}}_{i(m)} = \boldsymbol{r}_{mi} = \boldsymbol{y}_i - \boldsymbol{X}_i\hat{\boldsymbol{\beta}}$$

and the corresponding raw conditional residuals by

$$\hat{\boldsymbol{e}}_{i(c)} = \boldsymbol{r}_{ci} = \boldsymbol{y}_i - \boldsymbol{X}_i\hat{\boldsymbol{\beta}} - \boldsymbol{Z}_i\hat{\boldsymbol{u}}_i.$$

The residuals are conditional since the conditional mean of $\boldsymbol{y}_i$ is $E(\boldsymbol{y}_i \mid \boldsymbol{u}_i) = \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{Z}_i\boldsymbol{u}_i$. The conditional residuals will be considered since we are mainly interested in the difference between the observed values and the level-2 predicted values. The raw conditional residuals have a tendency to be dependent with different variances, even if the residuals of the true model are independent and have equal variances [11].

A method to overcome this problem is to use the conditional studentized residual, whereby a scaling factor (i.e. dividing factor) of either the true or estimated standard deviations of the residuals themselves is applied to the raw conditional residual. This is a more appropriate method because raw residuals may come from a population with unequal variances. Standardized residuals are obtained using the true standard deviations, and studentized residuals are obtained using the estimated standard deviations [11]. Studentized residuals are further classified as internal or external studentization, where internal studentization refers to when the observation corresponding to the residual is included in the calculation of the estimated standard deviation, and external studentization when it is excluded.

An alternative method called the Pearson residual can be used when the variability of the estimated parameters $\hat{\boldsymbol{\beta}}$ are assumed to be insignificant. For this method, the residuals are scaled using the estimated

standard deviation of the dependent variable instead [11].

Residual diagnostics test for the existence of a specific pattern in the residuals of the model, and are carried out by plotting a set of known residuals with the predicted residual values [11]. Once the known and predicted residuals are plotted against each other, they are checked manually to see if a pattern exists between the two. The plots also test for the existence of a constant variance σ^2.

Other tests are also performed on the residuals to test for distributional assumptions, including a box-plot for the level-2 groups, which tests for equal variances for the different clusters - in this case for the schools. A box-plot is also useful for detecting populations with large variations, or that contain unusual observations like outliers or influential observations. The normality assumption is tested using QQ-plots and histogram plots, since the residuals of the population are assumed to follow a normal distribution [11].

5.2 Influence diagnostics

Estimation methods relying on the likelihood are susceptible to influences of unusual observations and outliers. Therefore, influence diagnostics aim to identify which observations may have a heavy influence on the estimates for both the fixed-effect parameters and the variance components, as well as to quantify the effect that removing those values would have on the overall analysis of the full data set.

Influence statistics can be categorised as either iterative or non-iterative methods [8]. An iterative diagnostics method involves refitting the model with a subset of the data, indicated by U, is removed. Hence, the process is slow as the covariance parameters need to be recalculated with each iteration. In contrast, noniterative diagnostics methods make use of explicit formulas, which have the advantage of being more time efficient. However, they require the assumption that all covariance parameters are known, with the exception of the residual variance σ^2 [8]. Note that multilevel models allow for different covariance structures for each cluster; however, this will not be discussed in this essay.

When performing influence diagnostics, typically a "top-down" method is suggested. This means checking the overall influence diagnostics, thereafter using other diagnostic methods to see the effects that a given set of observations has on other aspects of the model. These aspects include changes in the values of the parameter estimates and their precision, as well as the influence on predicted values. Table 1 and 2 below (extracted from [11]) summarise the methods that are considered in this essay, as well as the formulae to calculate the diagnostic statistics and the decision criteria to decide if a subset U is influential or not. Note that the decision criteria are just a guideline for determining whether or not the observations are influential, and vary between sources. Furthermore, note that the same notation as [8] is used whereby a subscript U denotes calculations that have excluded some subset U of the full data set when the calculations were performed.

Name	Aspect	Formula	Decision Criteria
Restricted likelihood distance /displacement	Predicted value	$RLD_U = 2[l_R(\hat{\boldsymbol{\beta}}) - l_R(\hat{\boldsymbol{\beta}}_U)]$	$RLD_U > \chi^2_{0.75}(n)$ where n =number of fixed & covariance parameters
Cook's D	Estimate	$D(\boldsymbol{\beta}) = \frac{(\hat{\boldsymbol{\beta}}-\hat{\boldsymbol{\beta}}_U)'(\hat{\boldsymbol{\beta}}-\hat{\boldsymbol{\beta}}_U)}{rank(\boldsymbol{X})\times \hat{var}(\hat{\boldsymbol{\beta}})}$	$D(\boldsymbol{\beta}) > \frac{4}{n}$
Covariance ratio	Precision	$covratio(\boldsymbol{\beta}) = \frac{\lvert\hat{var}(\hat{\boldsymbol{\beta}}_U)\rvert}{\lvert\hat{var}(\hat{\boldsymbol{\beta}})\rvert}$	$covratio(\boldsymbol{\beta}) < 1$
Sum of squares PRESS residuals	Predicted value	$PRESS_U = \sum_{i\epsilon U}(y_i - x_i^{'}\hat{\boldsymbol{\beta}}_U)$	$PRESS_U$ large

Table 1: Influence on overall and fixed-effect diagnostics for LMMs

Name	Aspect	Formula	Decision Criteria
Cook's D	Estimate	$D(\boldsymbol{\theta}) = \frac{(\hat{\boldsymbol{\theta}}-\hat{\boldsymbol{\theta}}_U)'(\hat{\boldsymbol{\theta}}-\hat{\boldsymbol{\theta}}_U)}{\hat{var}(\hat{\boldsymbol{\theta}})}$	$D(\boldsymbol{\theta}) > \frac{4}{n}$
Multivariate DFFITS statistic	Estimate	$MDFFITS(\boldsymbol{\theta}) = \frac{(\hat{\boldsymbol{\theta}}-\hat{\boldsymbol{\theta}}_U)'(\hat{\boldsymbol{\theta}}-\hat{\boldsymbol{\theta}}_U)}{\hat{var}(\hat{\boldsymbol{\theta}}_U)}$	$MDFFITS(\boldsymbol{\theta}) > \frac{4}{n}$
Covariance ratio	Precision	$covratio(\boldsymbol{\theta}) = \frac{\lvert\hat{var}(\hat{\boldsymbol{\theta}}_U)\rvert}{\lvert\hat{var}(\hat{\boldsymbol{\theta}})\rvert}$	$covratio(\boldsymbol{\theta}) < 1$
Trace of covariance matrix	Precision	$covtrace(\boldsymbol{\theta}) = \lvert trace\left(\frac{\hat{var}(\hat{\boldsymbol{\theta}}_U)}{\hat{var}(\hat{\boldsymbol{\theta}})}\right) - q\rvert$	$covtrace$ large

Table 2: Influence on covariance parameters diagnostics for LMMs

Note that in Table 2 above, the difference between the Cook's D and the Multivariate DFFITS (MDFFITS) statistic is the denominator value. Cook's D statistic used the estimates for the covariance parameters from the full data set, whereas the MDFFITS statistic uses an estimate that has been recalculated after subset U was removed.

The following example, based on the example on the study of rat pups in Chapter 3 in [11], is given to illustrate the process and interpretation of influence diagnostics. The data, provided by JC Pinheiro and DM Bates in [5], consisted of 30 female rats who were given different dosages of an experimental treatment, namely high, low and control dosages. Thereafter the birth weights of the pups that the females gave birth to was measured to analyse the effect that the treatment had on the weights of the pups at birth. Ten female rats were assigned to each dosage group, however, three rats died during the experimental process so only the data from 27 litters was collected. Each litter ranged from 2 pups to 18 pups in size. This data represents a two-level clustered data set, where the level-1 observations of analysis are the rat pups, and the level-2 clusters are the litters that the pups belong to.

The variables used are WEIGHT (dependent variable), SEX (level-1), LITSIZE (level-2) and the dummy variables TREAT1 and TREAT2 for high and low levels of treatment (level-2) respectively. The model is:

Level-1 $$Weight_{ij} = \beta_{0i} + \beta_{1i}SEX_{ij} + e_{ij}$$

Level-2 $$\beta_{0i} = \gamma_{00} + \gamma_{01}LITSIZE_i + \gamma_{02}TREAT1_i + \gamma_{03}TREAT2_i + u_{0i}$$

$$\beta_{1i} = \gamma_{10} + \gamma_{11}TREAT1_i + \gamma_{12}TREAT2_i$$

Several hypothesis tests were performed, resulting in a model with the treatment/sex interaction parameters excluded and different covariances for the control group and the groups that received treatment i.e. $\sigma^2_{control} \neq \sigma^2_{high/low}$. The final model is thus expressed as:

Level-1 $$Weight_{ij} = \beta_{0i} + \beta_{1i} SEX_{ij} + e_{ij}$$

Level-2 $$\beta_{0i} = \gamma_{00} + \gamma_{01} LITSIZE_i + \gamma_{02} TREAT1_i + \gamma_{03} TREAT2_i + u_{0i}$$

$$\beta_{1i} = \gamma_{10}$$

Combined model $$Weight_{ij} = \gamma_{00} + \gamma_{01} LITSIZE_i + \gamma_{02} TREAT1_i + \gamma_{03} TREAT2_i + \gamma_{10} SEX_{ij} + u_{0i} + e_{ij}$$

Model diagnostics were performed on this final model. The values for the various influence diagnostics were calculated using SAS PROC MIXED and graphs were plotted, which are given in Figure 1 and 2 below. Figure 1 gives the graphs obtained for the overall and fixed effect diagnostics, and Figure 2 gives the graphs obtained for covariance parameter diagnostics.

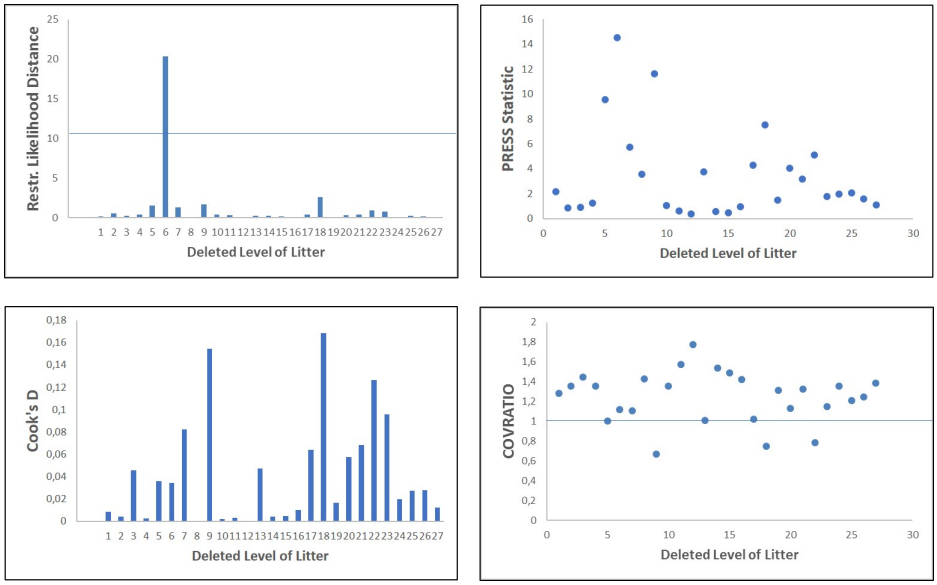


Figure 1: Effect of removing each litter on summary measures of influence for the fixed effects of the model

The first graph in Figure 1 above displays the impact on the restricted likelihood distance statistic when each litter is removed. All points are compared to a reference line equal to the 75^{th} percentile for the χ^2 distribution with degree of freedom equal to the number of fixed-effect and covariance parameters in the model. In this case, the reference is $\chi^2_{0.75}(8) = 10.22$. It can be seen that removing litter 6 has a substantial influence on the restricted likelihood distance statistic, implying that it has the largest overall influence on the model.

The predicted error sum of squares calculated for each litter is given by the PRESS statistic. Litter 6 has a high value for the PRESS statistic, suggesting that it including this litter may not be appropriate when predicting values.

The Cook's D statistic gives a measure of the overall simultaneous effect that removing each litter has on the fixed-effect parameter estimates. Observations are deemed influential if the Cook's D statistic is greater than $\frac{4}{n}$, where n is the number of clusters under evaluation [4]. It can be seen that although litter 6 has a large influence on the restricted likelihood distance statistic, it only has a minor influence on the estimates of the fixed-effect parameters. However, litters 9 and 18 seem to have large values of the Cook's D statistic, indicating that they have a significant influence on the fixed-effect parameter estimates.

The Covratio statistic assesses the change that removing each litter has on the precision of the fixed-effect parameters estimates. The decision criteria for the Covratio statistic is 1; that is, if the statistic value is < 1, it suggests that the variance of that subset is relatively large, thus removing that litter improves the precision for the estimates of the fixed-effect parameters. Litters 9, 18 and 22 give values lower than 1 for the Covratio statistic, implying that the variances for those litters are quite large.

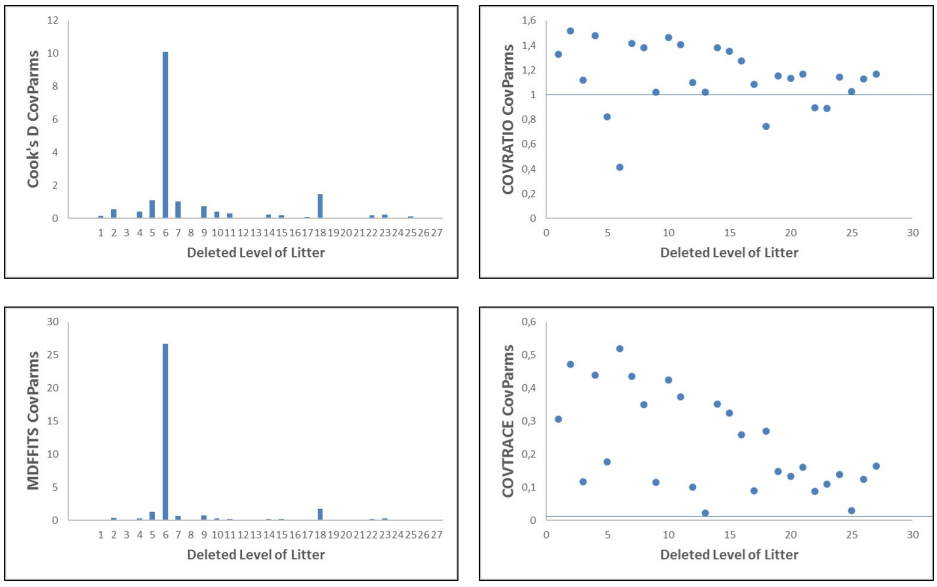


Figure 2: Effect of removing each litter on summary measures of influence for the covariance parameters of the model

Figure 2 displays the effect of removing each litter on the influence diagnostics for the model covariance parameters. The Cook's D statistic and the MDFFITS statistic in Figure 2 give a measure of the influence that removing a subset has on the estimates of the covariance parameters, which includes the residual variance σ^2. In the figure above, litter 6 gives very high values for both the Cook's D and the MDFFITS statistics, implying that removing litter 6 has a large influence on the estimates of the covariance parameters.

The Covratio and Covtrace statistics give a measure of the influence that removing a subset has on the precision of the estimates of the covariance parameters. Subsets that result in a Covratio value less than 1 are deemed influential, therefore removing those subsets will improve the precision of the covariance parameter estimates. In this example, Litters 5, 6, 18, 22 and 23 all gives Covratio values less than 1, and

therefore can be investigated. Moreover, litter 6 gives a significantly lower value than the rest, therefore is has a larger influence on the precision. Subsets that have a value of 0 for the Covtrace statistic are considered not influential. The larger the Covtrace statistic value is, the more influential that subset is. It can be seen above that litter 6 gives the largest Covtrace statistic value, which agrees with the deduction from the Covratio statistic that litter 6 influences the precision of the covariance parameter estimates the most.

When performing influence diagnostics, it is important that all diagnostic methods are taken into account when considering the removal of a subset. If the subset is considered influential, the effects that the removal has on the estimates for the fixed-effects as well as the covariance parameters should be investigated and compared. Different strategies to deal with influential data are available, including removing the data set and re-evaluating the model [4]. Alternatively, the model specifications can be adapted, data consistencies can be checked, or additional data can be obtained so that the influential observations are taken into account [4]. One way of adapting the model specifications is by adding additional variables to the model, which are used to explain the outliers or influential data and result in a better fitting model. If it is viable to the case, data consistencies must be checked, since errors can occur during the experimental process, such as during measurement or coding, which can result in seemingly influential or outlying data.

6 Simulation study

In this section, a two-level multilevel random slopes model with one covariate on each level will be simulated, with known parameter values. The model considered is the same as Model 5 in Section 3.1.2. For unit j in cluster i, the model on the two levels is

Level-1 $\quad y_{ij} = \beta_{0i} + \beta_{1i}x_{ij} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$

Level-2 $\quad \beta_{0i} = \gamma_{00} + \gamma_{01}z_i + u_{0i}$

$\beta_{1i} = \gamma_{10} + u_{1i} \qquad \begin{pmatrix} u_{0i} \\ u_{1i} \end{pmatrix} \sim N_2 \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{01} & \tau_{11} \end{pmatrix} \right)$

and the combined model is

$$\begin{aligned} y_{ij} &= \gamma_{00} + \gamma_{01}z_i + \gamma_{10}x_{ij} + (u_{0i} + u_{1i}x_{ij} + e_{ij}) \\ &= (\gamma_{00} + \gamma_{01}z_i + u_{0i}) + (\gamma_{10} + u_{1i})\, x_{ij} + e_{ij} \end{aligned}$$

where the fixed part of the model is $\gamma_{00} + \gamma_{01}z_i + \gamma_{10}x_{ij}$ and the random part is $u_{0i} + u_{1i}x_{ij} + e_{ij}$. Furthermore, $\gamma_{00} + \gamma_{01}z_i + u_{0i}$ is the random intercept and $\gamma_{10} + u_{1i}$ is the random slope.

After simulation of the data in SAS, the model parameters will be estimated using the SAS PROC MIXED function, and the estimates compared to their true values. Various models will be fitted to the simulated data, and the estimates for the fixed-effect and covariance parameters, as well as the fit statistics will be compared and discussed. A comparison of the REML and the ML estimates of the model from which the data was simulated will be conducted. Furthermore, the parameters will also be estimated using a generalised linear model to illustrate the effect that can result from not taking the dependency of the observations into account. The simulation will be based on an example provided in [3]. See Appendix for the relevant SAS programs.

The simulation will consist of 300 clusters each containing n_i number of observations, where n_i follows a $POI(20)$ distribution.

For cluster i, the combined model in vector notation is

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{u}_i + \boldsymbol{e}_i, \qquad i = 1, 2, ..., 300$$

such that $\boldsymbol{u}_i = \begin{pmatrix} u_{0i} \\ u_{1i} \end{pmatrix} \sim N_2(\boldsymbol{0}, \boldsymbol{D})$ with $\boldsymbol{D} = \begin{pmatrix} 25 & 15 \\ 15 & 35 \end{pmatrix}$, and $\boldsymbol{e}_i \sim N\left(\boldsymbol{0}, \sigma^2\boldsymbol{I}_{n_i}\right)$ with $\sigma^2 = 64^2$.

In the matrices $\boldsymbol{X}_i = \begin{pmatrix} 1 & z_i & x_{i1} \\ 1 & z_i & x_{i2} \\ \vdots & \vdots & \vdots \\ 1 & z_i & x_{i,n_i} \end{pmatrix}$ and $\boldsymbol{Z}_i = \begin{pmatrix} 1 & x_{i1} \\ 1 & x_{i2} \\ \vdots & \vdots \\ 1 & x_{i,n_i} \end{pmatrix}$, the z_i elements will be simulated from a $N(0, 1)$ distribution and the x_{ij} elements from a $UNIF(0, 10)$ distribution.

The fixed-effect parameters are chosen as $\boldsymbol{\beta} = \begin{pmatrix} \gamma_{00} \\ \gamma_{01} \\ \gamma_{10} \end{pmatrix} = \begin{pmatrix} 50 \\ 70 \\ 20 \end{pmatrix}$.

From the selection of the values in the matrix $\boldsymbol{D}$, $corr(u_{0i}, u_{1i}) = 0.507$, which means that as the random variation in the intercept increases, the random variation in the slope also increases.

In the SAS program, data is simulated randomly according to the distributional assumptions above. Thereafter, different models that omit different fixed and random effects are fitted to the simulated data, and the corresponding parameters are estimated by specifying the REML method in the PROC MIXED function. The models that will be fitted are given below. Refer to Section 3.1 for more details.

- Null model: $y_{ij} = \gamma_{00} + u_{0i} + e_{ij}$
- Model 2a: random intercept with level 1 covariate only: $y_{ij} = \gamma_{00} + \gamma_{10}x_{ij} + u_{0i} + e_{ij}$
- Model 2b: random intercept with level 2 covariate only: $y_{ij} = \gamma_{00} + \gamma_{01}z_i + u_{0i} + e_{ij}$
- Model 3: random intercept with level 1 & level 2 covariates: $y_{ij} = \gamma_{00} + \gamma_{01}z_i + \gamma_{10}x_{ij} + u_{0i} + e_{ij}$

- Model 5: random slope with level 1 & level 2 covariates (no interaction):

  $y_{ij} = \gamma_{00} + \gamma_{01} z_i + \gamma_{10} x_{ij} + u_{0i} + u_{1i} x_{ij} + e_{ij}$

- Generalised linear model (GLM): $y_{ij} = \gamma_{00} + \gamma_{01} z_i + \gamma_{10} x_{ij} + e_{ij}$

The fixed-effect and covariance parameter estimates for the different models are given in Tables 3 and 4. Their corresponding standard errors are given in brackets.

Parameters	**Theoretical values**	**Null model**	**Model 2a** (x_{ij})	**Model 2b** (z_i)	**Model 3** (x_{ij}, z_i)
Fixed-effect parameters		Est. (SE)	Est. (SE)	Est. (SE)	Est. (SE)
γ_{00}	50	144.96 (4.63)	44.07 (4.82)	149.56 (2.21)	48.82 (2.58)
γ_{01}	70	-	-	71.75 (2.24)	71.66 (2.15)
γ_{10}	20	-	20.07 (0.29)	-	20.05 (0.29)
Covariance parameters		Est. (SE)	Est. (SE)	Est. (SE)	Est. (SE)
σ^2	4096	7909.19 (145.25)	4437.19 (81.50)	7909.19 (145.25)	4437.19 (81.49)
τ_{00}	25	6037.68 (525.69)	6074.71 (515.12)	1060.42 (119.18)	1111.46 (109.31)
Model fit criteria					
-2 Res log likelihood	-	74434.5	71000.1	73985.9	70531.7
AIC	-	74438.5	71004.1	73989.9	70535.7

Table 3: REML estimates for random intercept models

From the null model, the intraclass correlation coefficient is calculated as $ICC = \frac{\hat{\tau}_{00}}{\hat{\tau}_{00}+\hat{\sigma}^2} = \frac{6037.68}{6037.68+7909.19} = 0.567$. This means that 56.7% of the total variance occurs between clusters. This value is quite large, and supports the decision to use a multilevel model to take into account the dependency between clusters.

In Models 2a and 2b, covariates are introduced on only level-1 and level-2 respectively. The introduction of the covariate x_{ij} on level-1 has the effect of decreasing the within cluster variation by 43.9%, that is

$$\frac{\hat{\sigma}^2(Null\,model) - \hat{\sigma}^2(Model\,2a)}{\hat{\sigma}^2(Model\,2a)} = \frac{7909.19 - 4437.19}{7909.19} = 0.439.$$

Similarly, entering the covariate z_i on level-2 causes a decrease of 82.4% in the between cluster variation,

$$\frac{\hat{\tau}_{00}(Null\,model)}{\hat{\tau}_{00}(Null\,model) - \hat{\tau}_{00}(Model\,2b)} = \frac{6037.68}{6037.68 - 1060.42} = 0.824.$$

Introduction of both covariates x_{ij} and z_i in Model 3 leads to a decrease in both the within and the between cluster variation of 43.9% and 81.6% respectively.

When comparing the fit of the models by calculating the deviance, only models that are nested within each other can be compared. In Table 3, both Models 2a and 2b are nested in Model 3. Comparing Models 2a and 3 gives

$$Deviance = 74434.5 - 71000.1 = 3434.4 > 2(3-2) = 2$$

which means that Model 3 is a significantly better fit than Model 2a. Similarly, when comparing Models 2b and 3, the deviance is calculated as $Deviance = 448.6$, which also indicates that Model 3 fits significantly better than Model 2b.

The estimates of the fixed-effects in Model 3 are quite close to the theoretical values; however, the estimates for the covariance parameters, especially τ_{00}, are far from the true values. This is due to the fact that Model 3 does not take the random slope into account.

Parameters	**Theoretical values**	**Model 5 (REML)**	**Model 5 (ML)**	**Model 5 (GLM)**
Fixed-effect parameters		Est. (SE)	Est. (SE)	Est. (SE)
γ_{00}	50	48.35 (1.67)	48.35 (1.66)	48.95 (1.87)
γ_{01}	70	72.15 (1.49)	72.16 (1.49)	71.22 (0.96)
γ_{10}	20	20.18 (0.44)	20.18 (0.43)	19.94 (0.32)
Covariance parameters		Est. (SE)	Est. (SE)	Est. (SE)
σ^2	4096	4159.25 (78.28)	4159.35 (78.28)	5529.25 (99.09)
τ_{00}	25	25.36 (68.63)	19.92 (67.95)	-
τ_{01}	15	24.21 (13.81)	24.58 (13.72)	-
τ_{11}	35	32.34 (4.60)	32.15 (4.58)	-
Model fit criteria				
-2 Res log likelihood	-	70196.3	70201.7	71362.9
AIC	-	70204.3	70215.7	71364.9

Table 4: Random slopes model (REML and ML estimates) and GLM model

Both REML and ML estimates for Model 5 are given in Table 4. The deviance when comparing Models 3 and 5 (REML) is calculated as $Deviance = 335.4$, showing that Model 5 is a significantly better fit than Model 3.

The main difference between the REML and the ML estimates is the estimate for τ_{00}, the covariance for the intercept variable, which is lower for ML estimation. This downward bias of the ML estimates was

mentioned in the beginning of Section 4 ([1], [11]). It can be seen that the REML covariance estimates are very close to the true values.

Comparing the results of the GLM to that of the mixed model, it can be seen that the estimates for the fixed-effects parameters are very similar. However, the estimate for σ^2 is much larger in the GLM model, which will impact the results for hypothesis tests in the model.

7 Application to TIMSS data

In this section, the two-level multilevel model will be applied to the TIMSS 2015 data set, described in Section 2, to determine which variables have a significant impact on the dependent variable, namely the mathematics score in South Africa. Note that an index of 40 points represents a difference of 1 year in the level of mathematics. The centre point for South Africa's mathematics score is 372, which is 128 points lower than the international centre point of 500. This means that, on average, South Africa is approximately three years behind in terms of mathematical education on a global scale.

The model that will be considered is a random intercept model. Altogether, ten independent variables will be considered as the fixed-effect parameters: four on the learner level (level-1) and six on the school/teacher level (level-2). The independent variables considered for the learner level will be:

- Sex
- Language of learning & teaching (LOLT)
- Digital information devices
- Student confidence in mathematics

The independent variables considered on the school/teacher level will be:

- Number of years teaching
- Teacher's area of study being mathematics
- Teacher's area of study being educational mathematics
- Poverty index of the school
- Immediate area of the school
- Whether teacher arriving late poses a problem.

The fixed-effect parameters, as well as the covariance parameter for the residuals σ^2, and the covariance parameter for the random intercept τ_{00}, will be estimated using REML estimation. The procedure of

step-wise backward elimination will be applied to exclude independent variables that are not significant in the model.

From the null model, the ICC is calculated as $ICC = \frac{\hat{\tau}_{00}}{\hat{\tau}_{00}+\hat{\sigma}^2} = \frac{3090.08}{3090.08+3230.89} = 0.489$. This value is quite large, implying that there is a high dependency between learners in each school, i.e. the between school variation explains 48.9% of the total variation in the model. This supports the importance of the use of a multilevel model that will take the dependency into account.

7.1 Descriptive statistics

The descriptive statistics for the discrete variables are summarised in Table 5. For each category, the number of learners n, the mean mathematics scores and the corresponding standard errors are given. Values were assigned to the categories of ordinal variables for ease of interpretation, with positive values for the categories that were more favourable in terms of the average mathematics score. The category with a code of 0 is the reference category for that variable. For Sex, which is not an ordinal variable, males are the reference variable.

Variable		**Code**	n	**Mean**	**SE**
Sex	Girls	1	5732	362.48	76.93
	Boys	2	5471	364.52	77.45
Language of learning and teaching	Always/Almost always	1	3790	391.34	84.03
	Sometimes	0	6767	352.15	68.73
	Never	-1	554	319.98	67.10
Digital information devices	None	-1	1448	330.41	64.72
	1-3	0	4140	350.81	65.84
	> 3	1	5447	383.27	82.75
Area of study - Maths	Yes	1	8256	363.23	77.19
	No	2	2521	364.23	77.35
Area of study - Edu. Maths	Yes	1	4453	373.11	79.34
	No	2	6327	356.22	74.85
Poverty Index	Low	-2	2788	338.09	65.42
	Moderately low	-1	2333	340.31	61.71
	Moderate	0	2719	354.71	66.84
	Moderately high	1	1758	378.58	74.19
	High	2	1606	439.53	83.46
Immediate area of school	Urban	1	1486	378.02	73.88
	Suburban	2	1330	394.79	91.28
	Medium size city	3	832	387.16	89.28
	Small town/village	4	3834	359.55	71.42
	Remote rural	5	3563	343.16	67.94
Teacher arriving late	Not a problem	1	3615	388.12	84.00
	Minor problem	0	5405	351.23	72.41
	Moderate/serious problem	-1	2184	353.00	66.55

Table 5: Descriptive statistics for discrete variables in TIMSS data set

Descriptive statistics for the continuous variables, student confidence in mathematics and number of years teaching, are given in Table 6. The sample was split according to learners with a score below 372, the overall average score for Grade 9 learners in South African, and those with a score of 372 and above.

		Mean $<$ 372	**Mean $\geq$ 372**
Confidence_1	***n***	6200	4707
	Mean	9.50	10.11
	SE	1.51	1.98
Years_teaching	***n***	6058	4591
	Mean	13.55	14.21
	SE	9.05	10.30

Table 6: Descriptive statistics for continuous variables in TIMSS data set

From Table 6, it can be seen that the average mathematics score has a positive linear relationship with both the student's confidence and the teacher's number of year teaching on the marginal level.

7.2 Fitting the model

A random intercept model was fitted to the data, with four level-1 (learner) variables and six level-2 (school/teacher) variables. The first model that is fitted will consider all ten variables, as follows:

Level-1 $$Average_{ij} = \beta_{0i} + \beta_1 Sex_{ij} + \beta_2 LOLT_{ij} + \beta_3 Digital_{ij} + \beta_4 Confidence_{ij} + e_{ij}$$

Level-2 $$\beta_{0i} = \gamma_{00} + \gamma_{01} Years\_teaching_i + \gamma_{02} Area\_study1_i + \gamma_{03} Area\_study2_i + \gamma_{04} Poverty_i + \gamma_{05} Imm\_area_i + \gamma_{06} Late_i + u_{0i}$$

Combined model $$Average_{ij} = \gamma_{00} + \gamma_{01} Years\_teaching_i + \gamma_{02} Area\_study1_i + \gamma_{03} Area\_study2_i + \gamma_{04} Poverty_i + \gamma_{05} Imm\_area_i + \gamma_{06} Late_i + \beta_1 Sex_{ij} + \beta_2 LOLT_{ij} + \beta_3 Digital_{ij} + \beta_4 Confidence_{ij} + u_{0i} + e_{ij}$$

Insignificant parameters were excluded from the model using a backwards elimination process, resulting in a final model that only considered six parameters. The results obtained for the REML estimation procedure are summarised in Table 7.

When interpreting the results, it should be noted that the fixed-effects in the model are partial effects.

Variable	**Level**	**Code/Range**	**Parameter**	**Estimate**	**SE**	**P-value**
Fixed-effect parameters						
Intercept	-	-	γ_{00}	269.77	3.96	$<.0001$
Sex	1	1 = G, 2 = B	β_1	-4.41	1.06	$<.0001$
LOLT	1	-1, 0, 1	β_2	12.57	1.20	$<.0001$
Digital	1	-1, 0, 1	β_3	5.24	0.83	$<.0001$
Confidence_1	1	[3.196, 15.925]	β_4	9.87	0.31	$<.0001$
Poverty	2	-2, -1, 0, 1, 2	γ_{04}	21.82	1.73	$<.0001$
Late	2	-1, 0, 1	γ_{06}	19.57	3.41	$<.0001$
Covariance parameters						
Between schools	-	-	τ_{00}	1474.94	135.05	$<.0001$
Within schools	-	-	σ^2	2838.98	39.33	$<.0001$

Table 7: Fixed-effect and covariance parameter estimates from fitting Model 5

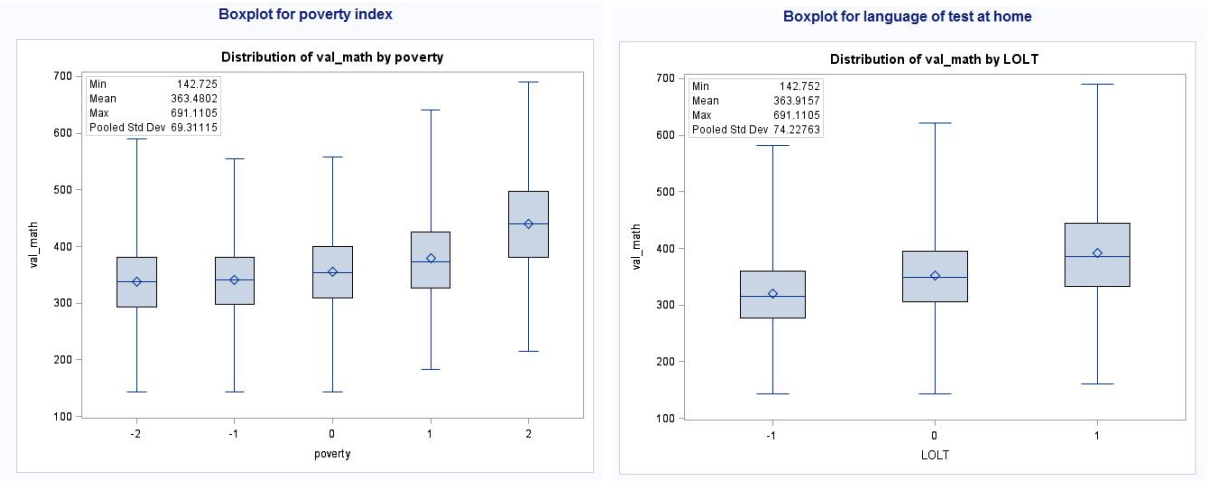


Figure 3: Boxplots for poverty index and language of test at home

The effects of each variable should only be interpreted while keeping the rest of the variables constant.

From the results in Table 7, on the learner level it can be seen that, when keeping all other variables constant, girls score on average 4.41 points lower than boys. Since LOLT is an ordinal variable, learners score 12.57 points higher when they always or almost always speak the language of teaching at home, compared to only speaking it sometimes, and $(12.57 \times 2) = 25.14$ points higher compared to learners who never speak the language of teaching at home. Similarly, there is an increase of 5.24 in the average mathematics score with each increase in category of the number of digital information devices they use. Confidence is a continuous variable. Learners achieve an average that is 9.87 points higher with each unit increase in the SCM scale, keeping all other variables constant. Keeping in mind that confidence ranges from 3.2 to 15.9, there is a difference of approximately 120 points from learners with a high level of confidence and those with a low level of confidence.

On the teacher level, only two out of the six variables were significant. The poverty index variable has an estimate of 21.82, which implies a difference of $(21.82 \times 4) = 87.28$ in scores between the least and most favourable categories, i.e. there is a two year difference in the mathematics level between learners from the highest and the lowest poverty bracket. Furthermore, learners score 19.57 points higher if the teacher arriving late does not cause a problem, compared to if it causes a minor problem, and $(19.57 \times 2) = 39.14$ points higher compared to cases where the teacher arriving late causes a moderate/serious problem.

The interpretations noted above agree with the marginal results given in the descriptive statistics in Section 7.1, especially for the ordinal variables, which have higher average scores the more favourable the category. The boxplots given in Figure 3 for the poverty index and LOLT visually illustrate the positive marginal effect that these two variables have, which are in line with the results from their partial effects.

8 Concluding remarks

In summary, the theory of a two-level multilevel statistical model was investigated; in particular, the theoretical background of the model and the notation were discussed in-depth. Multilevel models take into account the dependency that occurs between groups in clustered or repeated data, which is why this model is often used when analysing data from the fields of education, medicine or the social sciences, where hierarchical data occurs. The two main methods of estimation, namely maximum likelihood and residual/restricted maximum likelihood estimation, were discussed and the log-likelihood formulae were derived for each method of estimation. Thereafter, the types of model diagnostics were explained, and an example from [11] was used to illustrate the application of influence and model diagnostics. A simulation study was conducted to apply and illustrate the theory in estimating the parameters of a random intercept model.

A random intercept model was applied to the TIMSS 2015 data set, which considered a total of ten fixed-effect parameters: four on the learner level (level-1) and six on school/teacher level (level-2). Using a process of backwards elimination, six variables were returned in the final model, namely sex, language of learning and teaching, number of digital information devices and student confidence in mathematics on the learner level. On the teacher level, the significant variables are the number of years teaching, the poverty index of the school and whether or not the teacher arriving late poses a problem. The most significant fixed-effect is the poverty index, inferring that the level of poverty for each school has a large impact on the average score. This variable should be investigated further to understand the effect it has on education in South Africa.

References

[1] Helen Brown and Robin Prescott. *Applied Mixed Models in Medicine*. John Wiley & Sons, 2014.

[2] Stephen R Cole, Haitao Chu, and Sander Greenland. Maximum likelihood, profile likelihood, and penalized likelihood: a primer. *American Journal of Epidemiology*, 179(2):252–260, 2013.

[3] Gretel Crafford and René Ehlers. Estimation of multilevel models with iterative generalised least squares. In *Annual Proceedings of the South African Statistical Association Conference*, volume 2016, pages 17–24. South African Statistical Association (SASA), 2016.

[4] Rense Nieuwenhuis, Manfred te Grotenhuis, and Ben Pelzer. influence.ME: Tools for detecting influential data in mixed effects models. *The R Journal*, 4/2, 2012.

[5] José C Pinheiro and Douglas M Bates. Mixed-effects models in s and s-plus. *Statistics and computing*, 1978.

[6] Stephen W Raudenbush and Anthony S Bryk. *Hierarchical Linear Models: Applications and Data Analysis Methods*, volume 1. Newbury Park, CA: Sage, 1992.

[7] V. Reddy, M. Visser, L. Winnaar, F. Arends, A. Juan, and C.H. Prinsloo. TIMSS 2015: highlights of mathematics and science achievement of grade 9 South African learners. *Human Sciences Research Council*, 2016.

[8] O Schabenberger. Mixed model influence diagnostics in proceedings of the twenty-ninth annual sas users group international conference. In *Proceedings of the Twenty-Ninth Annual SAS Users Group International Conference*, volume 189, page 29, 2004.

[9] Judith D Singer. Using SAS PROC MIXED to fit multilevel models, hierarchical models, and individual growth models. *Journal of Educational and Behavioral Statistics*, 23(4):323–355, 1998.

[10] Jichuan Wang, Haiyi Xie, and James F Fisher. *Multilevel models: applications using SAS*. Walter de Gruyter, 2012.

[11] Brady T West, Andrzej T Galecki, and Kathleen B Welch. *Linear Mixed Models: A Practical Guide Using Statistical Software*. CRC Press, 2014.

Appendix

SAS Program for Section 6 on simulation

```
proc iml;
call randseed(100,1);
N = 300;
beta = {50, 70, 20};
gam_00 = beta[1,];
gam_01 = beta[2,];
beta_1 = beta[3,];
tel = {25 15, 15 35};
sigma = 64;

do i = 1 to N;
n_i = randfun(1,"Poisson", 20);
u_i = randnormal(1, {0,0}, tel);
u_0i = j(n_i,1,u_i[,1]);
u_1i = j(n_i,1,u_i[,2]);

z_i = randfun(1,"Normal", 0, 1);
x_ij = round(randfun(n_i, "Uniform", 0, 10));
matX_i = J(n_i,1,1) || J(n_i,1,z_i) || x_ij;
matZ_i = J(n_i,1,1) || x_ij;
e_i = randfun(n_i, "Normal", 0, sigma);
y_i = matX_i*beta + matZ_i*u_i` + e_i;

matrix_i = J(n_i,1,i)||matX_i||matZ_i|| u_0i || u_1i ||y_i;
if i = 1 then matrix=matrix_i; else matrix=matrix//matrix_i;
end;
varnames={cluster X_i_1 z_i x_ij  z1 x_ij2 u_0i u_1i y};
create simulation from matrix [colname=varnames];
append from matrix;
quit;

proc mixed data=simulation covtest method=reml;
model y = / solution ddfm=sat;                          /*model 1: null model*/
random int / subject=cluster type=un;
run;

proc mixed data=simulation covtest method=reml;
model y = x_ij / solution ddfm=sat;                     /*model 2a: random
                        intercept with level 1 covariate only*/
random int / subject=cluster type=un;
run;

proc mixed data=simulation covtest method=reml;
model y = z_i / solution ddfm=sat;                      /*model 2b: random
                        intercept with level 2 covariate only*/
random int / subject=cluster type=un;
run;

proc mixed data=simulation covtest method=reml;
model y = z_i x_ij / solution ddfm=sat;                 /*model 3: random
                        intercept with level 1 & level 2 covariates*/
random int / subject=cluster type=un;
run;
```

```
proc mixed data=simulation covtest method=reml;
model y = z_i x_ij / solution ddfm=sat;              /*model 5: random slope
                     with level 1 & level 2 covariates (no interaction)*/
random int x_ij / subject=cluster type=un;
run;

proc mixed data=simulation covtest method=ml;
model y = z_i x_ij / solution ddfm=sat;              /*model 5: random slope
                     with level 1 & level 2 covariates (no interaction) ML
                     ESTIMATION*/
random int x_ij / subject=cluster type=un;
run;

proc mixed data=simulation covtest method=reml;
model y = z_i x_ij / solution ddfm=sat;              /*glm of model 5*/
run;
```

SAS Program for Section 7 application to TIMSS 2015

```
proc iml;
use tmp1.timss;
read all into TIMSS;

timss2015 = TIMSS;
n = nrow(timss2015);
status = j(n, 1, .);

do i = 1 to n;
      /*redefine values for LOLT*/
      if timss2015[i,3]=1 then timss2015[i,3] = 1; else;
      if timss2015[i,3]=2 then timss2015[i,3] = 1; else;
      if timss2015[i,3]=3 then timss2015[i,3] = 0; else;
      if timss2015[i,3]=4 then timss2015[i,3] = -1;

      /*redefine values for digital*/
      if timss2015[i,4]=1 then timss2015[i,4] = -1; else;
      if timss2015[i,4]=2 then timss2015[i,4] = 0; else;
      if timss2015[i,4]=3 then timss2015[i,4] = 1; else;
      if timss2015[i,4]=4 then timss2015[i,4] = 1; else;
      if timss2015[i,4]=5 then timss2015[i,4] = 1;

      /*redefine values for poverty*/
      if timss2015[i,13]=1 then timss2015[i,13] = -2; else;
      if timss2015[i,13]=2 then timss2015[i,13] = -1; else;
      if timss2015[i,13]=3 then timss2015[i,13] = 0; else;
      if timss2015[i,13]=4 then timss2015[i,13] = 1; else;
      if timss2015[i,13]=5 then timss2015[i,13] = 2;

      /*redefine values for late*/
      if timss2015[i,15]=1 then timss2015[i,15] = 1; else;
      if timss2015[i,15]=2 then timss2015[i,15] = 0; else;
      if timss2015[i,15]=3 then timss2015[i,15] = -1; else;
      if timss2015[i,15]=4 then timss2015[i,15] = -1;

      if timss2015[i,5] < 372 then status[i] = 0; else;
      if timss2015[i,5] >= 372 then status[i] = 1;
end;

timss2015 = timss2015 || status;
```

```
names = {'school_ID' 'sex' 'LOLT' 'digital' 'val_math' 'confidence_1'
'confidence_2' 'teacher_ID' 'years_teaching'
                  'area_study_1' 'area_study_2' 'feecat' 'poverty' 'imm_area'
'late' 'status'};
create timss2015 from timss2015 [colname = names];
append from timss2015;

/*BOXPLOTS*/

title 'Boxplot for poverty index';
proc sort data=timss2015;
by poverty;
proc boxplot data=timss2015;
plot val math*poverty;
inset min mean max stddev;
run;

title 'Boxplot for language of test at home';
proc sort data=timss2015;
by LOLT;
proc boxplot data=timss2015;
plot val_math*LOLT;
inset min mean max stddev;
run;

/*DESCRIPTIVE STATISTICS*/

/*Sex*/
proc sort data=timss2015;
by sex;
proc means data=timss2015;
var val_math;
by sex;
run;
/*Language of test at home*/
proc sort data=timss2015;
by LOLT;
proc means data=timss2015;
var val_math;
by LOLT;
run;

/*Digital information devices*/
proc sort data=timss2015;
by digital;
proc means data=timss2015;
var val_math;
by digital;
run;

/*Student confidence in mathematics (continuous)*/
proc sort data=timss2015;
by status;
proc means data=timss2015;
var confidence_1;
by status;
run;
```

```
/*Number of years teaching (continuous)*/
proc sort data=timss2015;
by status;
proc means data=timss2015;
var years_teaching;
by status;
run;

/*Area of study - Mathematics*/
proc sort data=timss2015;
by area_study_1;
proc means data=timss2015;
var val_math;
by area_study_1;
run;

/*Area of study - Educational Mathematics*/
proc sort data=timss2015;
by area_study_2;
proc means data=timss2015;
var val_math;
by area_study_2;
run;

/*Poverty Index*/
proc sort data=timss2015;
by poverty;
proc means data=timss2015;
var val_math;
by poverty;
run;

/*Immediate area of school location*/
proc sort data=timss2015;
by imm_area;
proc means data=timss2015;
var val_math;
by imm_area;
run;

/*Teacher arriving late at school*/
proc sort data=timss2015;
by late;
proc means data=timss2015;
var val_math;
by late;
run;

/*FITTING A NULL MODEL*/

proc mixed data=timss2015 covtest method=reml;
model val_math = / solution ddfm=sat;
random int / subject=teacher_ID type=un;        /*cluster = teachers*/
run;
/*FITTING A RANDOM INTERCEPT MODEL*/

/*Model - full model*/
proc mixed data=timss2015 covtest method=reml;
class sex area_study_1 area_study_2 imm_area;
model val_math = sex LOLT digital confidence_1  /*level-1 covariates*/
                 years_teaching area_study_1 area_study_2 poverty imm_area
                 late                           /*level-2 covariates*/
                 / solution ddfm=sat;
random int / subject=teacher_ID type=un;        /*cluster = teachers*/
run;
```

```
/*Model 2 - removing: area_study_2*/
proc mixed data=timss2015 covtest method=reml;
class sex area_study_1 imm_area;
model val_math = sex LOLT digital confidence_1  /*level-1 covariates*/
                 years_teaching area_study_1 poverty imm_area late
                                                /*level-2 covariates*/
                 / solution ddfm=sat;
random int / subject=teacher_ID type=un;        /*cluster = teachers*/
run;

/*Model 3 - removing: area_study_1*/
proc mixed data=timss2015 covtest method=reml;
class sex imm_area;
model val_math = sex LOLT digital confidence_1  /*level-1 covariates*/
                 years_teaching poverty imm_area late
                                                /*level-2 covariates*/

                 / solution ddfm=sat;
random int / subject=teacher_ID type=un;        /*cluster = teachers*/
run;

/*Model 4 - removing: imm_area*/
proc mixed data=timss2015 covtest method=reml;
class sex;
model val_math = sex LOLT digital confidence_1  /*level-1 covariates*/
                 years_teaching poverty late    /*level-2 covariates*/
                 / solution ddfm=sat;
random int / subject=teacher_ID type=un;        /*cluster = teachers*/
run;

/*Model 5 - removing: years_teaching*/
proc mixed data=timss2015 covtest method=reml;
class sex;
model val_math = sex LOLT digital confidence_1  /*level-1 covariates*/
                 poverty late                   /*level-2 covariates*/
                 / solution ddfm=sat;
random int / subject=teacher_ID type=un;        /*cluster = teachers*/
run;

/*Model 5 - ML estimation*/
proc mixed data=timss2015 covtest method=ml;
class sex;
model val_math = sex LOLT digital confidence_1  /*level-1 covariates*/
                 poverty late                   /*level-2 covariates*/
                 / solution ddfm=sat;
random int / subject=teacher_ID type=un;        /*cluster = teachers*/
run;
```

CLARA algorithm for image clustering

Mark de Lancey 10595466

WST795 Research Report

Submitted in partial fulfilment of the degree BCom(Hons) Mathematical Statistics

Supervisor: Dr Inger Fabris-Rotelli

Department of Statistics, University of Pretoria

31st October 2017

Abstract

Clustering is a procedure of partitioning data into meaningful groups or clusters and has important applications in artificial intelligence and pattern recognition. The CLARA algorithm is one of many clustering methods and is used to clustering large sets of data specifically. CLARA does so by repeatedly sampling a data set and then applying the PAM algorithm, a $k-$medoids solver, to these samples and the clusters the remainder of the data according to the medoids given by these sampled results. This research examined the CLARA algorithm and how it can be used to cluster images, which are examples of big data. CLARA proved to be an efficient method of clustering large sets of data by being superior in speed to other well known methods (such as $k-$means or PAM) while having a comparable quality of clustering.

Declaration

I, *Mark Stephen de Lancey,* declare that this essay, submitted in partial fulfillment of the degree *BCom(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Mark Stephen de Lancey

_ _

Dr Inger Fabris-Rotelli

_ _

Date

Acknowledgements

I acknowledge the financial support from the Center for AI Research, Meraka Institute, CSIR.

Contents

List of Figures

1 Introduction

Big data is a much discussed topic at present, and being able to cluster big data is essential. Clustering is a procedure by which a set of objects (or data) is partitioned into groups (known as clusters). The objects within each cluster should have some level of similarity to each other according to predefined criteria in order to make the groups meaningful. Clustering procedures have many applications including image/pattern recognition and artificial intelligence [4, 6, 9]. Image clustering, also known as image segmentation, is when clustering is applied to the data (pixels) that makes up an image. Image clustering plays an important role in computer vision and can also be used for image modification tasks such as compression [8]. Images are large data sets by nature, with even small 30×30 pixel icons constituting a data set with 900 observations. The colour, intensity, or texture of an image is often used as criteria for segmentation of an image. Images can be represented in different colour spaces, such as RGB (red, green, blue), HSV (hue, saturation, value) and several others [12]. RGB will primarily be used here, along with some testing in the HSV space.

There are many clustering methods available. One of the earliest known partitioning methods used for image clustering is the k-means algorithm, first published by MacQueen [7], which attempts to partition n objects into k clusters by associating each object with the nearest mean. The k-medoids algorithm was developed later [6]. It is similar to the k-means algorithm but instead of using means, it assigns actual objects in the data set as representatives (medoids) to centre the clusters around. The k-medoids algorithm is more robust than k-means, especially when outliers are taken into account [6]. One method of applying the k-medoids algorithm is the Partitioning Around Medoids (PAM) procedure, developed by Kaufman and Rousseeuw [5]. The PAM procedure adjusts the k-medoids algorithm to prevent an exhaustive search for optimal clusters. However the PAM procedure has exponential computational complexity $O(n^2)$ for n objects in the set, rendering it inefficient for large data sets [6, 4]. To improve the PAM procedure, Kaufman and Rousseeuw developed the Clustering LARge Applications program (CLARA), which combines random sampling and the PAM algorithm to cluster large sets of data. Unlike PAM, the CLARA algorithm theoretically only has linear complexity $O(n)$ [9, 6].

This document will focus on the CLARA algorithm and how it can be used to cluster images and other large data sets. The CLARA method was tested against the k-means and PAM algorithms to explore its advantages and disadvantages in criteria such as computational complexity, robustness and suitability for image clustering. This report first establishes how images and colours are defined digitally in section 2. Then it defines the dissimilarity and evaluation criteria that can be used to perform clustering and measure its effectiveness in section 3. Details and procedures for the algorithms used are explained in section 4 and finally results from testing the algorithms are given in section 5.

2 Image Clustering

Before clustering can even begin, it needs to be established what data will be clustered and what is the criteria that will be used to determine meaningful groups. This section will discuss the data stored inside a typical digital image and what measure of similarity can be used to partition the data within the image into clusters.

2.1 Matrix representation of a digital image

An image needs some form of digital representation in order to be viewed and edited on a computer. Images can be represented in a matrix format because a 2-dimensional photo or picture can be translated to an $m \times n$ matrix (which is itself a 2-dimensional structure). For example the image shown in Figure 1 is a 30×30 pixel jpeg image of a logo belonging to social media website Twitter. It is 30 pixels wide and 30 pixels high (900 pixels total) and can be represented by a 30×30 square matrix where each element of the matrix contains information about the corresponding pixel in the same position. The matrix in this case will have 900 entries (one per pixel). A 30×30 pixel image would normally be displayed smaller, however it has been enlarged for easier viewing in this document.

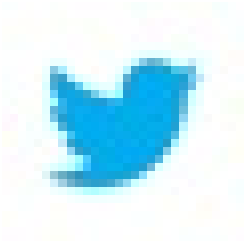


Figure 1: 30×30 pixel Twitter bird icon in colour

2.2 Univariate Images

The content in each cell of the matrix will determine what is shown on screen when an image file is read. A simple univariate form of an image is a greyscale. In this case each cell of the matrix contains a single variable, luminous intensity, and the picture will only display as shades of grey. In digital images, the greyscale variable ranges from 0 (pure black) to 255 (pure white) in integer values. A greyscale representation of the logo in the previous section is shown in Figure 2.

This image would be represented by a 30×30 matrix with each element containing a number in the greyscale range. On the outskirts of the greyscale image, where it appears mostly as white, the equivalent greyscale matrix elements are near the value 255, or pure white. Near the centre, however, there will be a range of much lower values indicating where the grey Twitter bird appears in the greyscale. Of note is that this is a tiny image, of only 30×30 pixels (as previously stated, it is in fact enlarged in this

Figure 2: 30×30 pixel Twitter bird icon as a greyscale

document). Despite this it has 900 entries in its greyscale matrix, a simple univariate representation of the image. Thus a large amount of data is required to represent even the smallest and simplest digital representations of images. This can lead to some extremely large data sets for today's modern high definition images. A greyscale image of 1920×1080 pixels will have $2'073'600$ entries in it's greyscale matrix. This is merely for the univariate greyscale case. For the multivariate cases (i.e. colour images) it becomes even more complex. This means that very efficient algorithms will be needed to simply traverse these matrices, let alone group the data within them into meaningful clusters as required. These issues of clustering large data sets and images are explored in greater detail later in this report.

2.3 Multivariate Images

In order to view and edit images in colour on a computer, more than one variable will be required. Images on a computer are typically represented by a mix of the colours red, green and blue. These are the additive primary colours and the model is known as the RGB colour space. This is a very common colour model and is used to display images on an LCD screen. A picture represented by the RGB model has 3 variables for each pixel, one for the intensity of each additive primary colour: red, green and blue. Each intensity variable has a range of 0-255, just like the greyscale variable. If all three RGB variables are set to 0, a pure black pixel is represented. If all three RGB variables are set to 255, a pure white pixel is represented [10]. RGB images are usually represented by having 3 matrices, one for each intensity variable. These 3 matrices combined form a 3-dimensional array data structure which represents a 2-dimensional image. For example an image with 4×4 pixels could be represented in RGB form by the $4 \times 4 \times 3$ array shown in Figure 3 [1].

[1] Abhineet Saxena, 29 June 2016, "Convolutional Neural Networks (CNNs): An Illustrated Explanation", XRDS, http://xrds.acm.org/blog/2016/06/convolutional-neural-networks-cnns-illustrated-explanation, accessed 26 July 2017

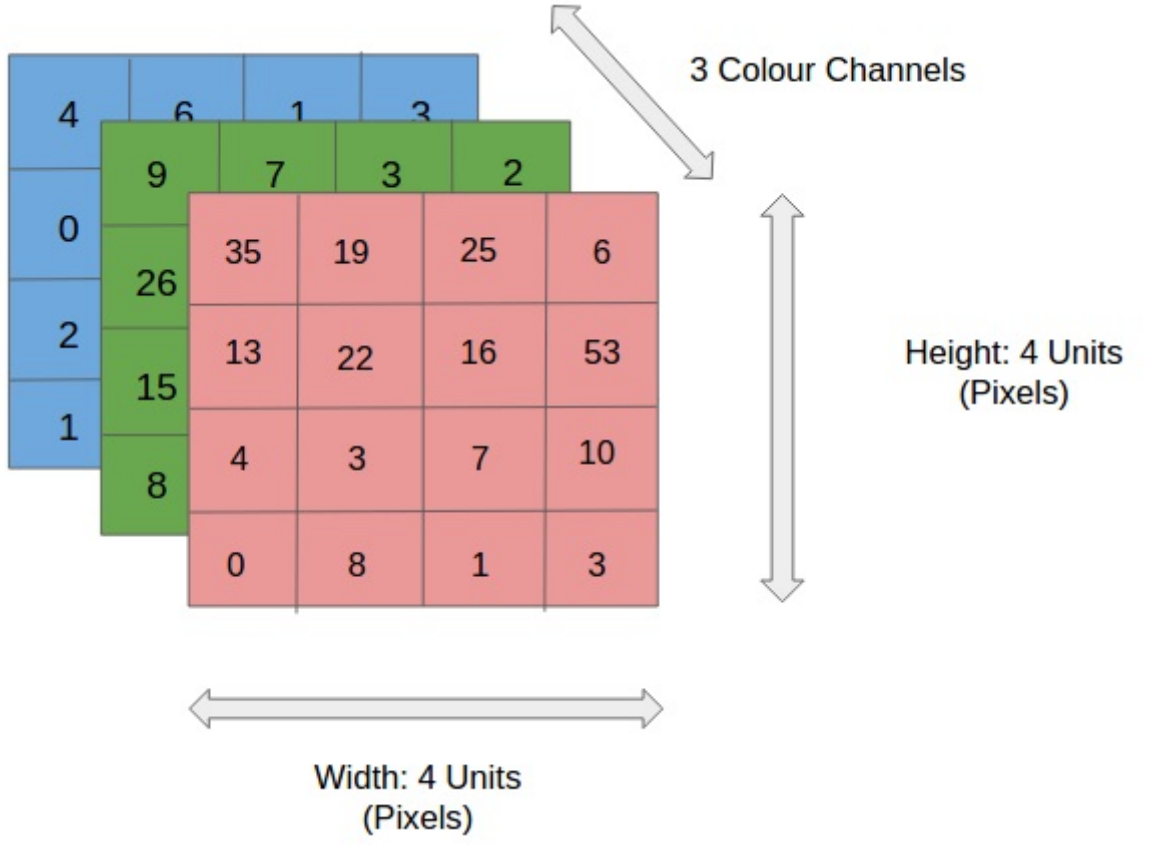


Figure 3: Red, green and blue matrices for an RGB image.

The colour version of the Twitter logo shown in Figure 1 with therefore have a $30 \times 30 \times 3$ RGB array and thus be represented by 2700 data points (900 red, green and blue). RGB values can also be displayed in vector form as a point for a single pixel, for example, a pure red pixel can be denoted by (255,0,0) with each value on the point corresponding to the value in the red, green and blue matrices for that pixel [10].

There are other colour models to represent images. The CMYK model uses the subtractive primary colours cyan, magenta and yellow along with "key", or black, to create the colour spectrum. It is often used in colour printing. The HSL (hue, saturation and lightness) or HSV (hue, saturation and value) models are cylindrical coordinate representations of RGB values. These models are used in image editing software. Lab colour space (or CIELAB) is a more complex system that uses two variables each representing a colour spectrum and a third variable for lightness [2]. Lab colour can represent a larger set of colours than RGB and can mathematically represent all perceivable colours. Lab space is therefore the most accurate representation of colour but because computer monitors use RGB or CMYK it is not commonly used and is often converted to one of these less accurate systems [10].

There is some criticism as to the use of RGB for image clustering, since it does not define the saturation

[2] 1994-2017 The MathWorks, Inc., "Representing color with the Lab color space", *MATHWORKS*, https://www.mathworks.com/discovery/lab-color.html, accessed 26 July 2017

or illumination of a picture in its variables. However it is a very common and simple colour model and will be the primary one used in this report. HSV space will also be tested on in this report .

2.4 Clustering an Image's Data

Even though an image is represented by a 2-dimensional matrix (or multiple 2-dimensional matrices in the multivariate case), the values in each matrix are all observations of a single variable. Therefore, as long as the position of each pixel's values are not lost in the process, these matrices can be unfolded into vector representations. For an $m \times n$ matrix X this can be done by stacking the columns of a matrix on top of each other into a single column vector as

$$X : m \times n = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & & & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix}.$$

$$Vector(X) : (m \times n) \times 1 = \begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{m1} \\ x_{12} \\ \vdots \\ x_{m2} \\ \vdots \\ \vdots \\ x_{1n} \\ \vdots \\ x_{mn} \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ \vdots \\ y_M \end{pmatrix} = Y : mn \times 1$$

.

Thus the observations of one of the images variables is now stored in a single column vector (this can also be done similarly using a row vector). It is now easier to cluster the image as only a single index needs to be used. This is especially useful for 'loops' in software programming. This will also allow the clustering algorithm to be used easily for other non-image data, as the observations of a single variable can be passed as a vector to the program. There will be a column vector for each variable, so a greyscale image will be represented by a single column vector, while an RGB image will have 3 such vectors. It is important to note that in general, computers will display or edit images using the matrix format. Once

the data set has been clustered and a new clustered data set (in column vector format) formed it is important to unstack the vector back into matrix form with the original dimensions and the pixels back in their original position in order to display or use the image on a computer. Therefore it is essential to keep track of the original dimensions as well as the method used to unfold the matrix (so that the reverse steps can be done once clustering is complete). If this is not done the final image can have the wrong dimensions or the transpose of the original matrix can be displayed, which might not resemble the image or it's properly clustered form at all.

The next step will be to choose a criterion to be used as a similarity (or dissimilarity) measure for the clusters of the image. This will be used to choose the cluster centres (which also depend on the algorithm chosen in the next chapter) as well as for assigning each object in the data set to that cluster centre. This is explained in greater detail in Section 3.

3 Similarity measures and evaluation criteria for clustering

Before presenting the clustering algorithms, some similarity measures for determining dissimilarity (or similarity) between points in a data set are discussed as well as evaluation methods to determine how meaningful the final clustering is.

3.1 Euclidean distance

Taking another look at the greyscale version of the Twitter logo shown in Figure 2, the picture can be intuitively separated into two parts: the grey "bird" in the foreground and the white background. Therefore, for this greyscale, one way to segment the image would be to partition it into 2 clusters. One cluster can have it's centre in the 100-200 range and group all the pixels in that range and the other cluster's centre would be near the value 255 and group all pixels near that value.

This gives a simple criterion for clustering a greyscale image, namely, selecting several cluster centres (how these are chosen is explain in section 4), say $C_j's$, meaningfully according to the greyscale variable (0-255) of the pixels and then assigning the pixels y_i where $i = 1, 2...M$ to the $C_j's$ in such a manner that the C_j that is selected has the lowest Euclidean distance from the y_i :

$$d_E(y_i, C_j) = \sqrt{(y_i - C_j)^2}$$

The problem now is to determine how to find cluster centres mathematically. The cluster centres themselves have to be meaningful otherwise the group assignment itself will not be useful. For example, choosing a pure white (255) cluster centre and an almost pure white (say 254) cluster centre as the two cluster centres for the Twitter bird image would leave the entire image represented by two very close

shades of bright white, which is not useful.

For the RGB image, there are 3 variables to consider as criterion for clustering similarity. Several possibilities arise from this multivariate case. One option is to cluster the images according to a single variable alone, e.g. the image can be segmented entirely according to its red matrix (using Euclidean distance such as in the greyscale case), thus completely ignoring the green and blue values in the image. This could be useful if for some reason an analysis of the red intensity of an image is needed. If the image is to be clustered according to all its colours however , none of the variables can be ignored. One possibility is to extend the Euclidean distance used in the greyscale case to 3 dimensions. In order to do this the image first needs to be represented with vectors for its red, green and blue values. Let

$$R = \begin{pmatrix} r1 \\ \vdots \\ \vdots \\ r_M \end{pmatrix}; G = \begin{pmatrix} g1 \\ \vdots \\ \vdots \\ g_M \end{pmatrix}; B = \begin{pmatrix} b1 \\ \vdots \\ \vdots \\ b_M \end{pmatrix}$$

be the vectors representing the red, green and blue values of the image respectively.

Hence the pixel in the first column and first row of the images matrix form, namely $x_{11} = y_1$, can be represented in vector form with RGB values, i.e $\boldsymbol{y_1} = (y_{1r}, y_{1g}, y_{1b})$.

Then cluster centres will be needed, these will also be represented in vector form:

$$\boldsymbol{C_j} = (c_{jr}, c_{jg}, c_{jb})$$

So that c_{jr}, c_{jg} and c_{jb} represent the red, green and blue values of cluster centre $\boldsymbol{C_j}$ respectively. The Euclidean distance for multiple dimensions can then be considered:

$$d_E(\boldsymbol{y}_i, \boldsymbol{C_j}) = \sqrt{(y_{ir} - c_{jr})^2 + (y_{ig} - c_{jg})^2 + (y_{ib} - c_{jb})^2}$$

This reduces the problem of clustering a multivariate representation of an image to using a single similarity measure for clustering, namely the 3-dimensional Euclidean distance between an RGB centre and the RGB values of each observation in the image. This measurement is simple and quick to calculate, only requiring the RGB values of the two pixels to be compared, however a notable problem arises from this similarity criterion for RGB. Only the final Euclidean distance calculated is used as a similarity measure without discriminating what values of red, green or blue were used as original input into the formula. To illustrate this problem, consider pure black as a cluster centre:

$$Let\, \boldsymbol{C_B} = (0, 0, 0)$$

Then consider pure red and pure green as pixel observations from the image. Let

$$R = (255, 0, 0) \text{ and let } \boldsymbol{G_1} = (0, 255, 0)$$

Thus the Euclidean distance between each of these points and the pure black cluster centre is as follows:

$$d_E(\boldsymbol{R_1}, \boldsymbol{C_B}) = \sqrt{(255-0)^2 + (0-0)^2 + (0-0)^2} = 255$$

$$d_E(\boldsymbol{G_1}, \boldsymbol{C_B}) = \sqrt{(0-0)^2 + (255-0)^2 + (0-0)^2} = 255$$

Therefore the distance between pure red and pure black is the same as the distance between pure green and pure black. In this case $\boldsymbol{G_1}$ and $\boldsymbol{R_1}$ could be assigned to the same cluster even though pure red and pure green might not be considered similar. There are alternative methods to calculate similarity between colour pixels. One of them is discussed in the next section.

3.2 Mahalanobis distance

Since the data set of an RGB image is multivariate, a similarity measure that accounts for multivariate statistical distance is preferred. A commonly used option for cluster analysis is the Mahalanobis distance. The Mahalanobis distance [2] measures the statistical distance between a point and a given distribution. The Mahalanobis distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ from a multivariate data set with sample variance-covariance matrix $\boldsymbol{S}$ is:

$$d_M(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^{\boldsymbol{T}} \boldsymbol{S^{-1}} (\boldsymbol{x}-\boldsymbol{y})}$$

To apply this dissimilarity metric to an RGB image, consider the matrix of RGB values for each pixel, where the pixels are stored in column vector form:

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{y_1} \\ \vdots \\ \vdots \\ \boldsymbol{y_M} \end{pmatrix} = \begin{pmatrix} r1 & g1 & b1 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ r_M & g_M & b_M \end{pmatrix}.$$

From this the sample variance-covariance matrix for the R,G and B variables can be seen as:

$$\boldsymbol{S} = \begin{bmatrix} s_{rr} & s_{rg} & s_{rb} \\ s_{gr} & s_{gg} & s_{gb} \\ s_{br} & s_{bg} & s_{bb} \end{bmatrix}$$

The cluster centres vector is $\boldsymbol{C_j} = (c_{jr}, c_{jg}, c_{jb})$ and the i^{th} RGB pixel is $\boldsymbol{y_i} = (r_i, g_i, b_i)$. To apply this dissimilarity metric to a cluster, let S be the sample variance-covariance matrix for the elements to be assigned to cluster j. The Mahalanobis distance from a pixel to a cluster's distribution can be calculated as:

$$d_M(\boldsymbol{y_i}, \boldsymbol{C_j}) = \sqrt{(\boldsymbol{y_i} - \boldsymbol{C_j})^{\boldsymbol{T}} \boldsymbol{S^{-1}} (\boldsymbol{y_i} - \boldsymbol{C_j})}$$

The Mahalanobis distance accounts for the statistical distance of a point inside the cluster from the given distribution within a cluster (using the cluster centre as the mean) [2]. This will help, for example, in differentiating pure red from pure green in an image that has a very high red mean. A data set with 2 variables is plotted in Figure 4[3]. Four points on the axis are shown with coloured stars (2 yellow and 2 purple) on the axis. These emphasised points are all equally distant from the mean of the data in terms of euclidean distance, however it is clear that the points shown in yellow are outside the cluster formed by the data set, whilst the purple points are within the range of the data (and therefore can be considered statistically closer in terms of standard deviations). These points therefore have a different Mahalanobis distance from the mean and they are coloured according to these distances as shown on the right.

3.3 Structural Similarity (SSIM) index

Structural similarity, or SSIM [13] is a measure of the perceived quality of an image. SSIM is not used during clustering but rather as a final measure of the quality of a clustered image. The luminance, l, contrast, c, and structure, s, of an image are compared between two images. Given a reference image y, and another image x (usually a processed form of image y), the SSIM is calculated as:

$$SSIM(x, y) = l(x, y) \times c(x, y) \times s(x, y)$$

where

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}, c(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}, s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}$$

and (c_1, c_2, c_3) are stabilising coefficients [13].

The SSIM index ranges from -1 (poor similarity) to 1 (if and only if identical images are compared).

[3] 1994-2017 The MathWorks, Inc., Mahalanobis distance, MATHWORKS, https://www.mathworks.com/help/stats/mahal.html, accessed 26 July 2017

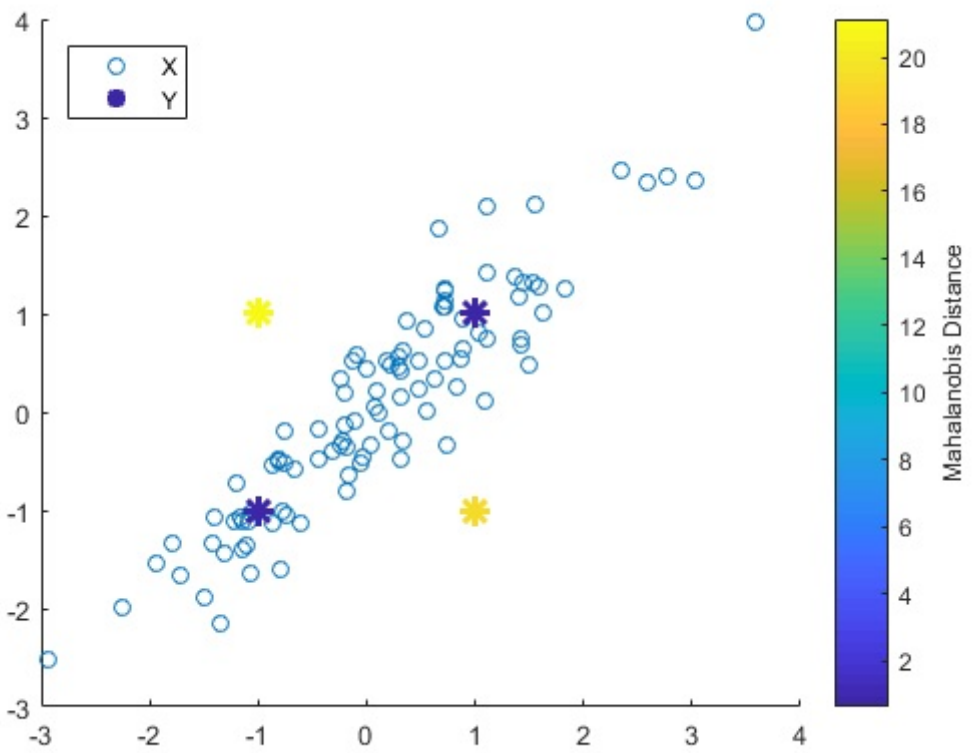


Figure 4: Mahalanobis Distance used in a bivariate data set

This is considered an improvement over more traditional measures such as the mean square error (MSE) for image data [13]. For testing in this research report, SSIM is to be used as the objective function to maximize. SSIM is fast to calculate and each image clustered for testing was tagged with a corresponding SSIM for analysis.

3.4 Silhouette

The silhouette is a measurement on each point in the data set after clustering, of how well the point matches its own cluster vs. the other clusters. The silhouette value ranges from -1 to +1. A high silhouette value for a point indicates that the point is very similar to other points within it's cluster, very dissimilar to points outside the cluster. The silhouette for a point y_i is calculated as:

$$S(y_i) = \frac{b(y_i) - a(y_i)}{max[a(y_i), b(y_i)]}$$

where $a(y_i)$ is the average distance from y_i to other points in its own cluster and $b(y_i)$ is the average distance between y_i and points in it's neighbouring cluster (i.e, the cluster with the next best fit for point y_i).

The silhouette for each point can be plotted or an average silhouette can be obtained to determine

how good the clustering is. A high average silhouette indicates a good clustering solution. A low or negative average silhouette can be a sign that either the number of clusters chosen is poor (too many or too few), or the algorithm had poor initialization or convergence [11].

3.5 Image intensity histograms

An image histogram plots the possible pixel values (such as greyscale intensity) a picture consists of against the number of pixels with that value. A histogram of reference image y, can be compared to a histogram of clustered image x to compare the similarity of the images, and therefore, how well the clustering algorithms grouped the pixels.

4 Clustering Algorithms

4.1 $k-$Means Algorithm

The $k-$means algorithm partitions a set of objects, X, into a fixed number of clusters, k. [7]. The algorithm begins by generating k centroids within the space of the data set. This may be done randomly or via a heuristic which is the case when using the $k-$means++ initialization algorithm, explained in the next subsection [1]. Each object in the data set is then assigned to the centroid that will minimize the dissimilarity between it and the object. These sets of objects each form a cluster. After this, the mean of the cluster is computed, and then the centroid is replaced with the newly calculated mean. The objects in the set are then reassigned to these new centroids, again by selecting the centroid that minimises the distance. The algorithm then alternates between recalculating the means and reassigning the objects until there are no reassignments, in an expectation–maximization fashion.

Let $X = \{x_1, x_2, ..., x_n\}$be the objects in the data set. Perform $k-$means as follows:

STEP 1 Generate k centroids within the domain of the object space. These centroids will each represent a cluster. Let y_j denote the centroid $j = 1, 2, ..., k$

STEP 2 Assign each object x_i to each centroid y_j with the minimum distance $d(x_i, y_j)$ between the two. The sets of objects assigned to each centroid form the k clusters, so that cluster $C_j = \{x_i : i \in S_j\}$where S_j is the set of indices for objects assigned to cluster j.

STEP 3 Calculate the mean of each cluster. These k means will replace each corresponding centroid in its cluster as the new centroid, namely the y_j's.

STEP 4 Re-assign each x_i to the new centroid that minimizes the distance between the two.

STEP 5 Repeat steps 3 and 4 until no new re-assignments are needed, or until a predetermined stopping point.

4.2 $k-$Means++ Initialisation

$k-$means can be performed upon randomly generated initial centroids, however this can produce poor results. To improve the initial starting values, the $k-$Means++ initialisation algorithm is proposed [1].

STEP 1 Select a single centroid at random from the observations.

STEP 2 Calculate the distance between every point in the data set and the nearest centroid that has already been chosen.

STEP 3 Select a new centroid, with a probability proportional to the distances calculated in step 2, so that points further away from any current centroids are more likely to be chosen as the new centroids.

STEP 4 Repeat steps 2 and 3 until k centroids have been chosen.

Once the initialisation algorithm has been complete, $k-$means can be performed upon these initial centroids given. These centroids are equivalent to medoids for PAM as actual objects from the data set are used. Thus $k-$means++ can also be used to initialise PAM, which is explained in the next section [1].

4.3 $k-$Medoids problem and Partitioning Around Medoids (PAM) algorithm

The minimization problem and algorithm given here is based on the $k-$medoid method as given in Kaufman and Rousseeuw [6]. The $k-$medoid problem, like the $k-$means algorithm, partitions a set of objects, X, into a fixed number of clusters, k. These k clusters are each represented by a single representative object, or medoid, that is an element of the original set X. The remaining objects are then assigned to these medoids according to the similarity measure. This algorithm is repeated until the total dissimilarity of the entire clustered set, $\sum_{i=1}^{k}\sum_{j=1}^{N_i} d(x_i, x_j)$, is minimized, where the x_i's are the medoids, x_j's are non-representative objects and N_i is the total number of non-medoid objects currently within cluster i. The $k-$medoid problem can be mathematically formulated as follows:

The following minimization problem needs to be solved:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} d(x_i, x_j) z_{ij}$$

subject to the following restrictions:

$$\sum_{i=1}^{n} z_{ij} = 1,\ j = 1, 2, ..., n$$

$$z_{ij} \leq y_i,\ i, j = 1, 2, ..., n$$

$$\sum_{i=1}^{n} y_i = k,\ k = \text{number of clusters}$$

$$z_{ij}, y_i \in \{0, 1\},\ i, j = 1, 2, ..., n$$

Where y_i is a binary variable such that:

$$y_i = \begin{cases} 1 & \text{if and only if } x_i \text{ is selected as a representative object} \\ 0 & \text{otherwise} \end{cases}$$

and z_{ij} is a binary variable such that:

$$z_{ij} = \begin{cases} 1 & \text{if and only if } x_j \text{ is assigned to } x_i's \text{ cluster} \\ 0 & \text{otherwise} \end{cases}$$

so that:

- exactly k representatives are chosen
- each object must be assigned to a representative
- a non-representative object can only be assigned to a representative

The PAM algorithm is one solution to the $k-$medoids minimization problem. It can be mathematically formulated as follows:
Let $X = \{x_1, x_2, ..., x_n\}$. Perform PAM as follows:

STEP 1 Select k objects from X as representatives either randomly or using $k-$means++. These k x_i's each represent a cluster.

STEP 2 Assign each object x_j that is not a medoid, to the medoid x_i with the least dissimilarity, $d(x_i, x_j)$, between the two. These are the clusters.

STEP 3 Within each cluster i, calculate the distance between the medoid x_i and all other objects within its cluster, $\sum_{j=1}^{N_i} d(x_i, x_j)$. Then, replace the medoid with each and every other object

in the cluster, each time recalculating the total distance within the cluster. If another object besides the original medoid has the lowest total dissimilarity, assign it as the new medoid and assign the old medoid as a non-representative in the cluster.

STEP 4 Repeat steps 2 and 3 until the total dissimilarity across all clusters,$\sum_{i=1}^{k}\sum_{j=1}^{N_i} d(x_i, x_j)$, is minimized.

The key differences between the $k-$means and PAM algorithm is that PAM uses actual objects in the data as centroids and constantly recalculates the distance within each cluster and the entire data set. $k-$means reassigns objects to its own calculated centres and then adjusts these centres accordingly. A specific build step recommended by Kaufman and Rousseeuw can also be used to initialise the medoids [6]. The PAM algorithm lowers the computational time required to find good medoids compared to an exhaustive $k-$medoid algorithm, however the PAM algorithm still has exponential computational complexity, $O(n^2)$ [6, 4]. This makes PAM cumbersome for large data sets [6]. For this reason, research will be conducted into the CLARA algorithm, which uses random sampling in combination with the PAM algorithm to find medoids [9].

4.4 Clustering LARge Applications (CLARA algorithm)

The CLARA algorithm makes use of the PAM algorithm as a subroutine, but randomly samples the data instead of using the entire data set. T samples (usually 5) are drawn from the entire data set, each with a sample size of $40 + 2k$. CLARA is performed as follows:

STEP 1 Draw a sample from the data set of size $40 + 2k$.

STEP 2 Apply the PAM algorithm over this sample.

STEP 3 Use the medoids obtained in step 2 to cluster the entire original data set and calculate the total distance, $\sum_{i=1}^{k}\sum_{j=1}^{N_i} d(x_i, x_j)$, over all the clusters.

STEP 4 If the total distance from the last sample is the lowest so far, store the medoids given by PAM as well as the total distance. Discard previous results.

STEP 5 Repeat steps 1 to 4 a total of T times, then use the best set of medoids to cluster the entire data set and then save this as the final clustering.

This algorithm is of linear computation complexity $O(n)$ and is less complex than the $O(n^2)$ PAM algorithm on its own [6, 9]. CLARA will be the focus of the research and will be replicated and tested against other algorithms for clustering images.

5 Application

The algorithms explained in Section 4 are tested on various images in this section. All algorithms were programmed entirely in MATLAB unless otherwise indicated. The author developed a stand alone application entitled "Creeping Barrage" to facilitate the processing of images and the creation of the data sets required for analysis of $k-$means, PAM and CLARA. This application can run independently of a full MATLAB package, requiring only the runtime libraries. This allowed the program to be run simultaneously on several computers, speeding up the process of the analysis, particularly for the intense computational requirements of PAM. This application was entitled "Creeping Barrage" due to the manner in which it slowly increments the number of clusters to be formed with each iteration of an algorithm and, therefore, the computational intensity "creeps up" as the program progressively barrages the computer with increasing instructions. Comprehensive MATLAB code to produce the Creeping Barrage application and the algorithms it runs may be found in the appendix.

Of note is that MATLAB's built in k-medoids function does not use PAM by default for data sets larger than 3000 observations. Even when forced to use PAM, MATLAB was often unable to complete the clustering, requiring memory (RAM) of over 83 gigabytes for some of the matrices required to calculate distances. For this reason the author developed a custom-made version of PAM for this research, which decides whether to iterate commands or build matrices depending on memory available, and was thus able to cluster larger data sets. This version of PAM gives similar MSE and similarity to the MATLAB PAM on smaller data sets which the built in MATLAB PAM can successfully process. For fair comparisons between algorithms, all remaining algorithms were also coded without using the built-in MATLAB clustering functions. This also allowed for more control during testing.

MATLAB and the child program Creeping Barrage produced images and data sets. These data sets in turn, were analysed using SAS (Statistical Analysis System) with PROC GLM for ANOVA's. The SAS code used for analysis may be found in the appendix.

5.1 Brief overview of the Creeping Barrage application

Figure 5 shows the user interface (UI) for Creeping Barrage. The user can input a folder filled with images, all of which will be analysed. Other parameters to be chosen are in the options panel on the left of the UI. The parameter "seed" fixes the seed to be used for random number generation for the initial clusters (either generated randomly or by $k-$means++). This is so that a fair analysis can be conducted on the various algorithms as the performance tested with depend on the efficiency and randomness included in the algorithm itself, and vary less due to random number generation of initial clusters. $k-$means and PAM are systematic and have no randomness in the algorithm. CLARA samples randomly but is not given a fixed seed as the randomness of CLARA is part of this research analysis. "Start Clusters" and

"End Clusters" gives the boundaries on the number of clusters to be formed. The program will run the number of trials specified for each k. Thus if "Start Clusters" is set to 1, "End Clusters" set to 5, and trials set to 10, the program will run a total of 50 runs of the given algorithm on every image in the chosen folder with 10 trials for each $k = 1, 2, 3, 4, 5$. "CLARA T" refers to the number of samples that CLARA will perform PAM on. In the instructions and output panel, Creeping Barrage can save, load and execute sets of instructions with different parameters or sets of images each time. This allows it to be left alone for extended periods of time to process the data required for analysis. Figure 6 shows an example of Creeping Barrage loaded with an instruction set. Each line of the instruction set will be run separately and consecutively. A different data set will be produced with each instruction line. Figure 7 shows the single picture analysis tab of Creeping Barrage which allows an in depth analysis for clustering a single image by providing the user with summary statistics and statistical diagrams such as image intensity histograms and silhouette plots.

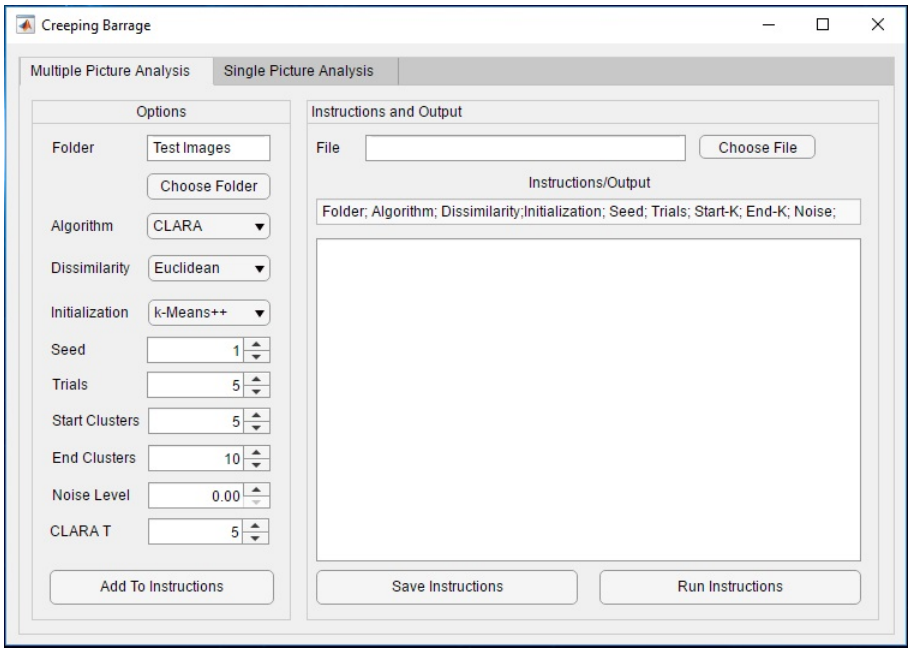


Figure 5: User interface for Creeping Barrage mass clustering interface

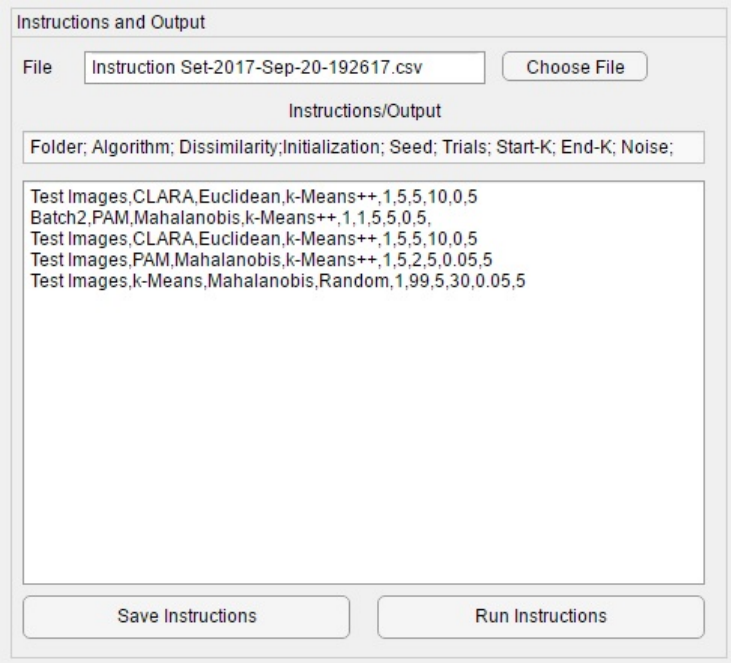


Figure 6: Saved instruction set loaded

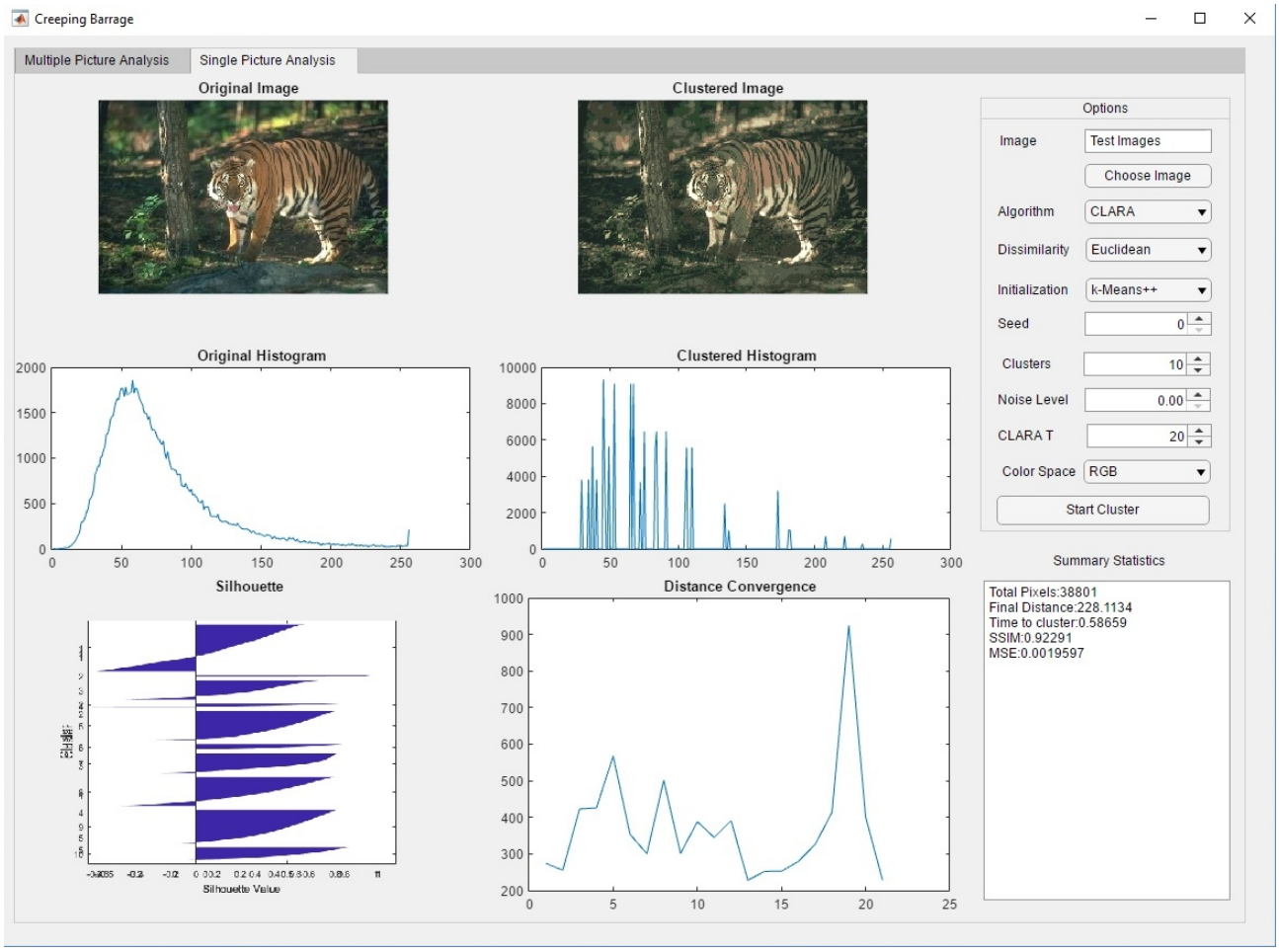


Figure 7: User interface for Creeping Barrage's single picture analysis tab

5.2 $k-$Means Algorithm testing for benchmarks

The algorithm tested here is $k-$means using $k-$means++ initialization. Since $k-$means is commonly used and is relatively efficient, it will be used as a reference algorithm to test which colour space (RGB, HSV) and dissimilarity measure (euclidean, Mahalanobis) will output images clustered with good SSIM values. $k-$means was performed on 100 different 240×160 pixel images using both colour spaces and dissimilarity measures. Clusters were fixed at 10. This was repeated 5 times per image, per colour space and per distance measure for a total data set of 2000 observations. An ANOVA performed on the resulting data set is show in Figure 8. A significant interaction was found between the colour space used and which of the two dissimilarity measures were used. From figure 8, it can be seen that the best distance measure was euclidean, which performed better in both colour spaces, and the best colour space to work with was RGB when using the euclidean measure. The mean SSIM for these best measures was 0.915. For the remaining trials and algorithms, the RGB space and euclidean distance is used for clustering. The output of various clustered images from the different spaces and dissimilarity measures can be seen in Figure 9.

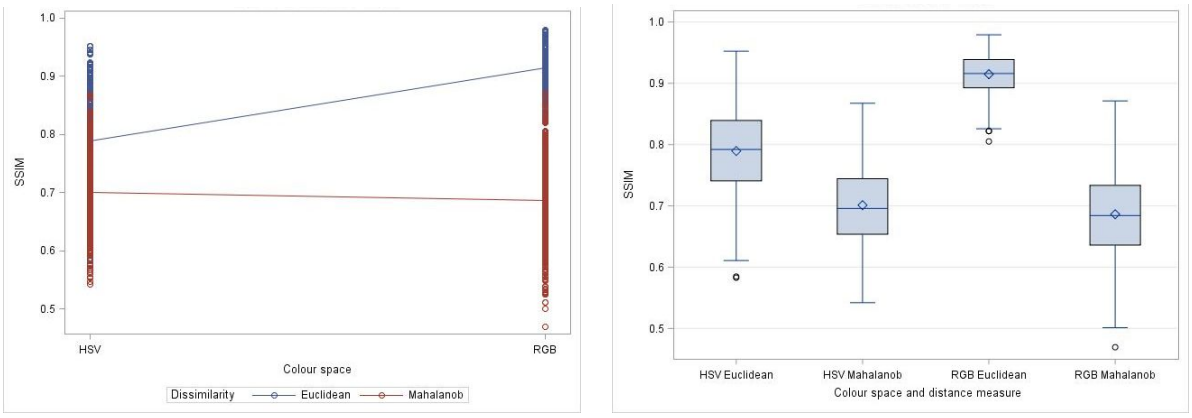
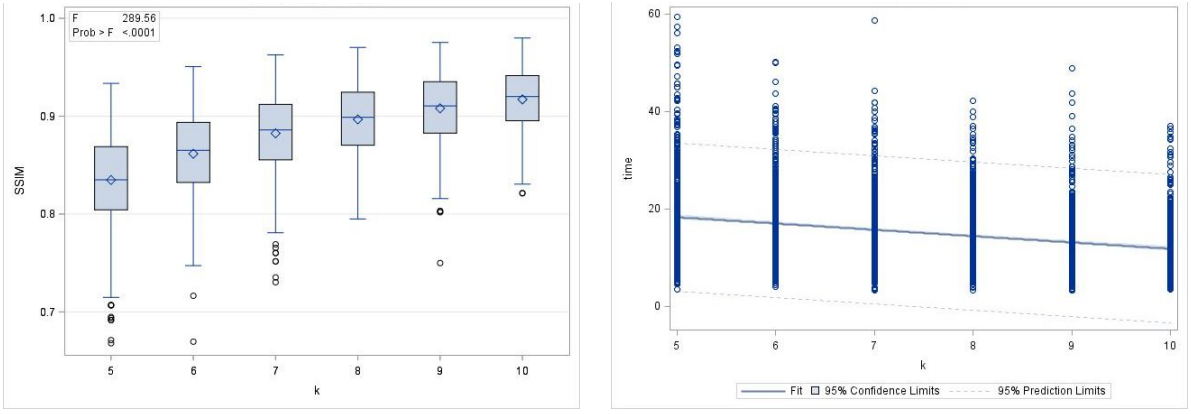


Figure 8: ANOVA of SSIM for $k-$means in different colour spaces

(a) (b) (c) (d) (e)

Figure 9: Output images for $k-$means with 10 clusters using different dissimilarity measures and different colour spaces, (a) the original image, (b) HSV space and Euclidean distance, (c) HSV space and Mahalanobis distance, (d) RGB space and Euclidean distance, (e) RGB and Mahalanobis distance.



(a) (b)

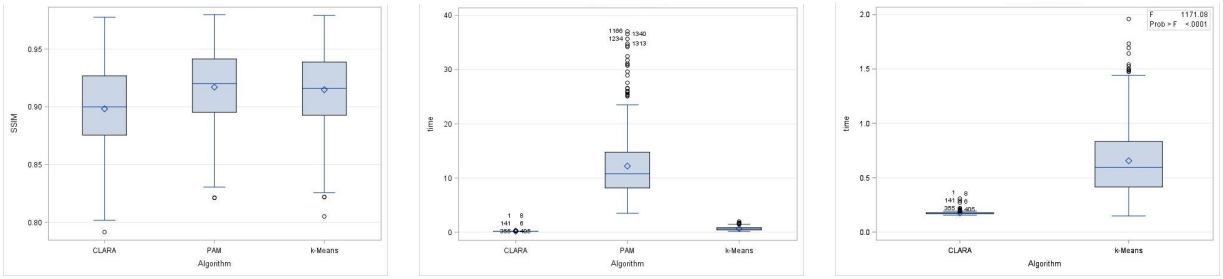
Figure 10: PAM algorithm: Box plots of the SSIM against number of clusters is shown in (a) while time required to cluster against number of clusters is shown in (b)

5.3 PAM analysis

In this section the algorithm PAM is analysed, also using $k-$means++ initialization. PAM is extremely slow and computationally intensive, so PAM was only tested with small images. For testing accuracy against the number of clusters, PAM was run on the same 100 BSDS500 images that $k-$means clustered in the previous section, but with the number of clusters increasing from 5 to 10 in each trial, performed twice on each image. A positive relation was found between the resulting SSIM index and the number of clusters used which can be seen in figure 10. This makes sense as the more clusters an algorithm has to work with, the closer to the original image the clustered version can become. This may not be optimal for the purposes of segmenting the image however, and may produce a low silhouette. However, only a weak negative relationship was found between the number of clusters and time to cluster, also shown in figure 10. This relationship explains very little of the variance in the time it takes to cluster an image, and the time is more likely related to the composition of the image itself. The lack of a strong relation between time and number of clusters can be explained by the fact that if there are more clusters, there will be, on average, less objects per cluster. Thus PAM will spend less computational time rotating medoids inside each cluster, but more time reassigning objects to clusters.

5.4 Comparison between $k-$Means, PAM and CLARA

In this section PAM and CLARA are tested, also using $k-$means++ initialization, and compared with the $k-$means algorithm. PAM and CLARA were run on the same 100 BSDS500 images that $k-$means clustered in the previous section. The process was repeated 5 times per picture. Clusters were again fixed at 10. The results were then combined with results from $k-$means for euclidean dissimilarity and RGB space. An ANOVA was performed on the final data set (with 1500 observations), which can be seen in figure 11. Significance was found for differences in both time and SSIM. The variance in SSIM explained



(a) (b) (c)

Figure 11: ANOVA results from combined data set of 1500 clustering trials. Box plots of the SSIM of the resulting images is shown in (a), the time required (in seconds) to cluster the images in (b) and again in (c) without PAM due to the massive difference in scale of the timings.

by different algorithms was small however, and CLARA, PAM and $k-$means had average SSIM values of 0.90, 0.92 and 0.91 respectively. A large variance in time can be seen clearly from the different algorithms in figure 11. CLARA's time was superior with an average clustering time of 0.17 seconds per image, while PAM had the worst average time of 12.18 seconds per image and the largest variance in time.

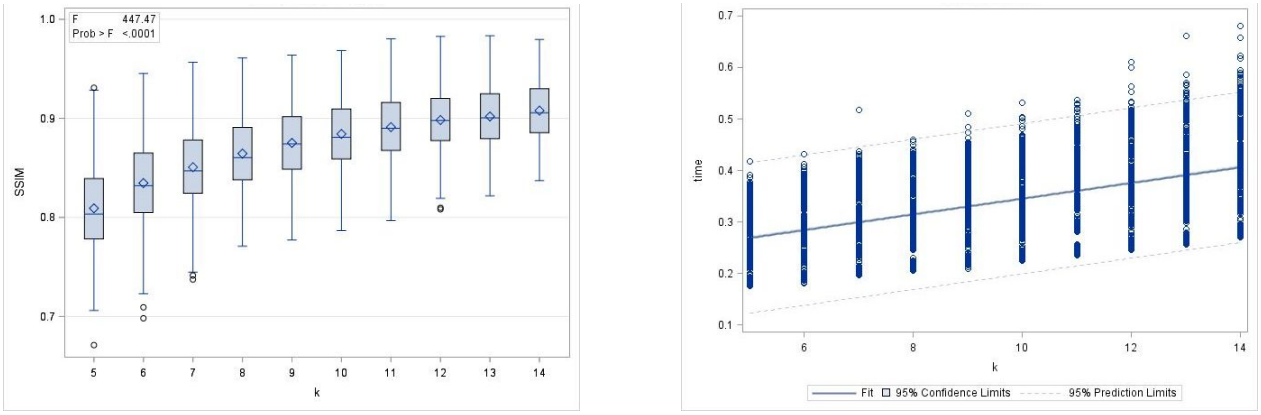
5.5 CLARA analysis

Additional analysis conducted using CLARA are shown here to explore the impact of the number of clusters used with CLARA as well as the number of samples to be taken. Figure 12 shows SSIM and time plotted against the number of clusters used. Similarly to PAM, an increasing number of clusters yields an improved SSIM. However unlike PAM, time to cluster increases with the number of clusters desired. This is due to the recommendation by Kaufman and Rousseeuw [6] to use samples of size $40+2k$ (as explained in section 4). This, in turn, decides on the size of the data set to send to PAM within CLARA's iterative process, and PAM has exponential complexity. Notably this $40+2k$ sample size is a recommendation and can be altered depending on the requirements. Other sample size recommendations have been proposed, such as sampling 1% of the total data set in M.K Pakhira's modified CLARA algorithm[9]. Figure 13 shows what happens to the output when the number of samples to run PAM on, T, is altered. Increasing T from the recommended 5 by Kaufman and Rousseeuw[6] does little or nothing to improve the SSIM output, but as expected the time required to cluster increases linearly with the number of samples chosen. Thus Kaufman and Rousseeuw's recommendation of 5 samples is confirmed to be good.

5.6 CLARA stress tests

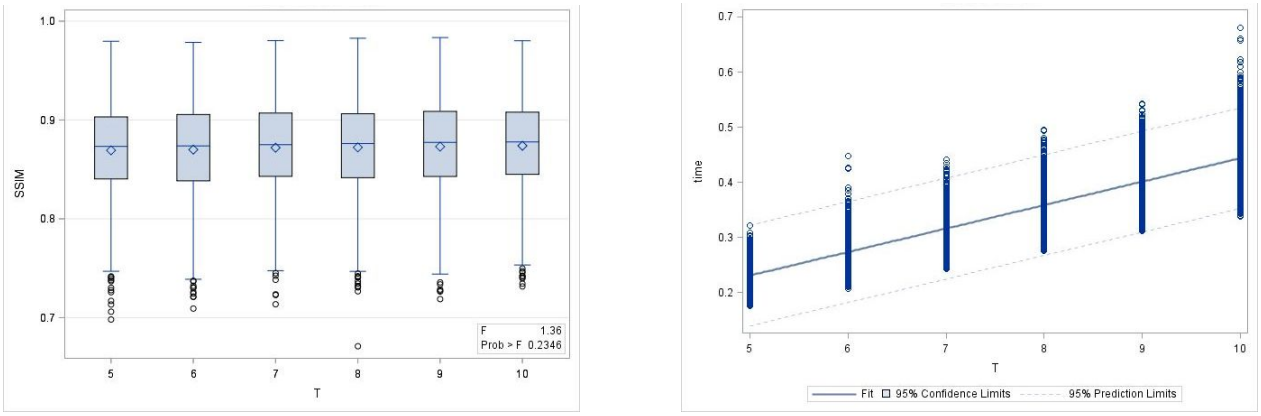
Stress tests were conducted using the CLARA algorithm to see how effectively it could cluster large images. As a final stress test, a large 5458×2915 image[4] was clustered with 30 clusters both by CLARA and $k-$means. When compared to the original, CLARA's output image had an SSIM of 0.8697, while

[4]Image file obtained from Wikimedia Commons, https://commons.wikimedia.org/wiki/File:WA_-_Dry_Falls_-_Huge_Channel_v1.png, accessed 20 September 2017.



(a) (b)

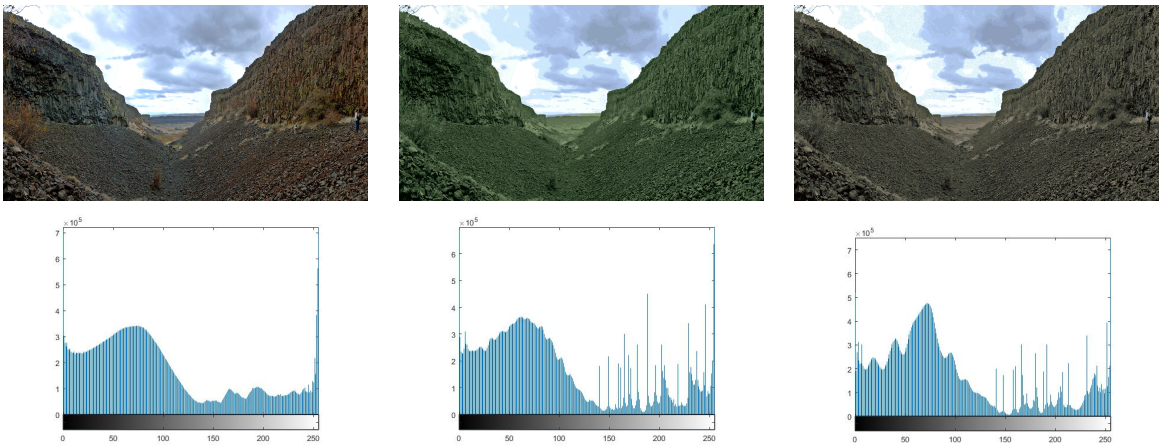
Figure 12: CLARA algorithm: Box plots of the SSIM against number of clusters, k, is shown in (a) while time required to cluster against number of clusters is shown in (b)



(a) (b)

Figure 13: CLARA algorithm: Box plots of the SSIM against number of samples chosen, T, is shown in (a) while time required to cluster against number of samples is shown in (b)

$k-$means produced an image with a SSIM of 0.9109, thus both have comparable structural similarity with the original (while the $k-$means SSIM is slightly higher than CLARA's). However, the CLARA algorithm took 23 minutes (as timed by MATLAB) to cluster the image while $k-$means took 4 hours and 55 minutes to cluster the same image on the same computer. Greatly reduced versions of these images along with histograms of the tonal distribution for comparison are shown in Figure 14.

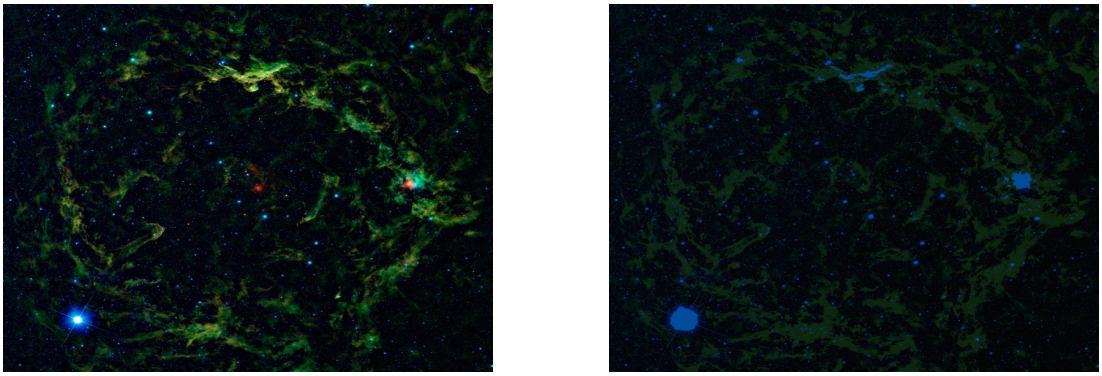


(a) (b) (c)

Figure 14: Stress test performed on a large 5458×2915 image. Image output shown along with corresponding tonal histograms shown below of (a) the original image (heavily reduced in size), (b) k-Means clustered image with 30 clusters, (c) CLARA clustered image with 30 clusters

To further illustrate CLARA's power in clustering large data sets a massive 15800×14700 infrared photo image of a nebula in the constellation Orion taken by NASA's Wide-field Infrared Survey Explorer[5] was clustered. This image is therefore a data set with 232.26 million RGB pixels. Several clusterings were done of this image on a machine with 64GB of RAM. The fastest clustering was completed within 1.9 minutes using 4 clusters, with an SSIM of 0.752. The best clustering in terms of SSIM was 0.876 using 15 clusters, however this took 2 hours and 14 minutes, again demonstrating the exponential complexity of the PAM algorithm. Notably, there was an observation clustered with an SSIM of 0.821 using 5 clusters that took under 3 minutes, indicating that this may be an example where taking more samples or performing CLARA multiple times may be preferable than increasing the number of clusters. This is especially the case when considering the additional 2 hours to cluster for a marginal increase in SSIM. The original image along with and example of a clustered version is shown in figure 15.

[5]Image file obtained from NASA's Jet Propulsion Laboratory, California Institute of Technology PHOTOJOURNAL, https://photojournal.jpl.nasa.gov/catalog/PIA14040, accessed 20 September 2017

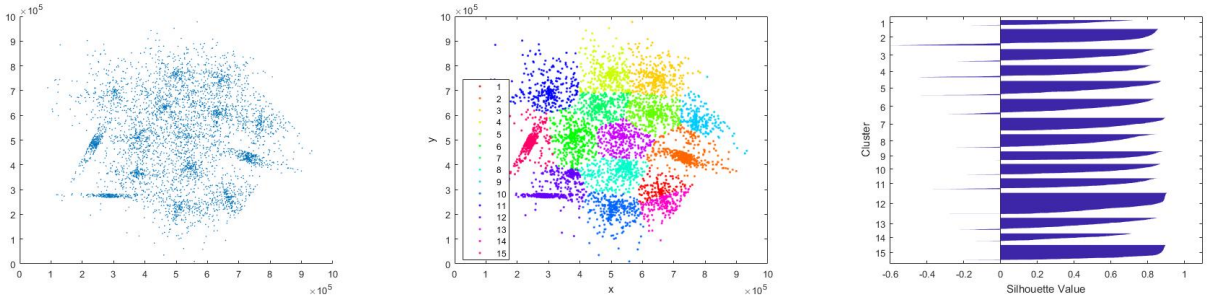


(a) (b)

Figure 15: NASA image. (a) Shows the original while (b) shows the image segmented into 5 clusters with CLARA

5.7 CLARA used on non-image data

Here CLARA is tested on a non-image benchmark data set. The data set is a synthetic 2-d data with N=5000 vectors and k=15 Gaussian clusters known as the noisy S4 data set [3]. CLARA successfully clustered the noisy data set into the 15 required clusters with an average silhouette of 0.5895, indicating the clusters were meaningful. The data set and results are shown in figure 16.



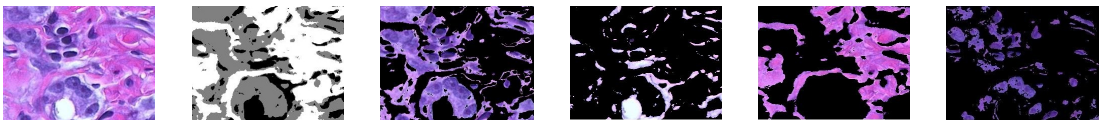
(a) (b) (c)

Figure 16: CLARA clustering a benchmark data sets into meaningful clusters. A noisy data set (a) is segmented into 15 clusters (b) with an average silhouette of 0.5895. A silhouette plot is shown in (c).

5.8 Practical application of CLARA on images

For a practical demonstration of the use of CLARA, an experiment was performed to separate the cell nuclei from an image of a stained tissue sample. In order to do this, the image was first segmented into 3 clusters as shown in figure 17. These clusters are then applied again to the original image to separate the pixels associated with each cluster into a new image (thus one additional image per cluster). It was then experimentally determined that the first cluster contained the nuclei. Finally the colours of the nuclei were separated from the rest of this first cluster by clustering again. The results are shown in figure 17. This experiment was originally performed using $k-$means. This demonstrates that CLARA is also

capable of performing such tasks.



(a) (b) (c) (d) (e) (f)

Figure 17: Separating the cell nuclei from a tissue sample using CLARA. (a) shows the original image while (b) shows how CLARA clustered the image into 3 clusters. (c) to (e) shows the clusters reapplied to the image and (f) shows the final image with the cell nuclei separated from the first cluster.

6 Conclusion

Big data is a much discussed topic at present. The CLARA clustering algorithm presented here provides a useful method for this case. The experiments conducted showed that clustering images with CLARA can achieve similar SSIM to the original image when compared with commonly used methods such as $k-$means and PAM with much less computational time needed. For further testing, CLARA is to be used on a training data set to determine how many clusters or samples are required to produce optimal or asymptotic SSIM indices. All clustering algorithms presented here can also be applied to non-image data.

References

[1] David Arthur and Sergei Vassilvitskii. K-means++: the advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[2] Brian S. Everitt, Sabine Landau, Morven Leese, and Daniel Stahl. *Cluster Analysis*. John Wiley & Sons Ltd, The Atrium, West Sussex, United Kingdom, 2010.

[3] P. Fränti and O. Virmajoki. Iterative shrinking method for clustering problems. *Pattern Recognition*, 39(5):761–765, 2006.

[4] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304, 1998.

[5] Leonard Kaufman and Peter J. Rousseeuw. Clustering by means of medoids. *Statistical Data Analysis Based on the L1-Norm and Related Methods*, pages 405–416, 1987.

[6] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Inc., Hoboken, New Jersey, 1990.

[7] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, 1967.

[8] Valmik. B. Nikam, Vinod J. Kadam, and Bandu. B. Meshram. Image compression using partitioning around medoids clustering algorithm. *IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 1*, 2011.

[9] Malay K. Pakhira. Fast image segmentation using modified CLARA algorithm. In *2008 International Conference on Information Technology*, pages 14–18. Institute of Electrical and Electronics Engineers (IEEE), December 2008.

[10] J.M. Prats-Montalbán, A. de Juan, and A. Ferrer. Multivariate image analysis: A review with applications. *Chemometrics and Intelligent Laboratory Systems*, 107(1):1–23, may 2011.

[11] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, nov 1987.

[12] Linda G. Shapiro and George C. Stockman. *Computer Vision*. Addison-Wesley Publishing Company, Inc, 2001.

[13] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Appendix

Appendix A: Creeping Barrage Main MATLAB Code

Listing 1: Creeping Barrage Main Entry Point

```
%void Main(Params)
function Main(instructions)
%Make new folder for results:
mainfolder = strcat('Test Set ',datestr(now,'yyyy-mmm-dd-HHMMSS'));
mkdir('Results',mainfolder);
%Make new file to output test results:
csvMain =strcat('Results\',mainfolder,'\','Test Data ',datestr(now,'
  yyyy-mmm-dd-HHMM'),'.csv');
csvFailed = strcat('Results\',mainfolder,'\','Failed Trials ',datestr
  (now,'yyyy-mmm-dd-HHMM'),'.csv');

%Begin Trials:
%
  **************************************************************************

obs = 0;
folderID = 0;
for i =1:length(instructions)
    rawstring = char(instructions(i,:));
    params = strsplit(rawstring, ',')
    folder = char(params{1});
    algorithm = char(params{2});
    dissimilarity = char(params{3});
    init = char(params{4});
    seed = str2double(params{5});
    trials = str2double(params{6});
    k_start = str2double(params{7});
    k_end = str2double(params{8});
    noise = str2double(params{9});
```

```
26      T = str2double(params{10});
27      HSVorRGB = char(params{11});
28      HSVorRGB
29
30      PathName = strcat('Test Images\',folder,'\');
31      files = dir(strcat(PathName,'\*.jpg'));
32
33      folderID=folderID+1;
34      subfolder =strcat(sprintf('%s %d','Subset',folderID),algorithm,'
          ',init,' ',dissimilarity,' ',HSVorRGB,' ',num2str(k_start),'to'
          ,num2str(k_end),' Noise_',num2str(noise));
35      mkdir(strcat('Results\',mainfolder),subfolder);
36
37      for file = files'
38          inimg = strcat(PathName,file.name);
39          for k=k_start:k_end
40              for i=1:trials
41  %               try
42                    obs = obs+1;
43                    [mn,distance,outimg,time,SSIM,MSE,~,~] =
                        Clustering(inimg,algorithm,dissimilarity,k,init
                        ,seed,T,noise,HSVorRGB,0);
44                    distance = distance(end);
45                    namestr = string(file.name);
46                    namestr = erase(namestr,'.jpg');
47                    baseFileName = sprintf('%s_%d_%d.jpg',namestr,k,
                        obs); % e.g. "1.png"
48                    fullFileName = fullfile('Results\',mainfolder,
                        subfolder,baseFileName);
49                    imwrite(outimg, fullFileName);
50                    M(obs,:) = {[obs],algorithm,dissimilarity,init,
                        HSVorRGB,[k],[time],[SSIM],[MSE],[mn],[distance
                        ],[noise],[T]};
51  %               catch
52                    try
```

```
53                          FAILED(obs,:) = {[obs],algorithm,
                              dissimilarity,init,HSVorRGB,[k],[noise],[T
                              ]};
54                          failTable = cell2table(FAILED);
55                          writetable(failTable,csvMain)%
56                      catch
57                          continue
58                          disp('failed to write failure');
59                      end
60 %                  end
61              end;%for_i
62          subTable = cell2table(M);
63          subTable.Properties.VariableNames = {'Obs','Algorithm','
              Dissimilarity','init','HSVorRGB','k','time','SSIM','MSE','
              mn','distance','noise','T'};
64          writetable(subTable,csvMain);
65          end;%for_k
66      end;%for_files
67      disp('Clustering Set Complete')
68
69 end;%for instructions
70 %
     **************************************************************************

71 %END TRIALS
```

Listing 2: Code to split image into dataset and perform iterative clustering commands

```
1  function [mn,distance,outimg,time,SSIM,MSE,silh,figs] = Clustering(
     inimg,algorithm,dissimilarity,k,init,seed,T,noise,HSVorRGB,single)
2  %Main Program for clustering without GUI
3  rgb = imread(inimg);
4  rng(seed);
5  if (noise)
6      rgb = imnoise(rgb,'salt & pepper',noise);
7  end%if
8  rng('shuffle');
9
10 Y = im2double(rgb);
```

```
11 switch HSVorRGB
12     case  'HSV'
13         Y = rgb2hsv(Y);
14     case 'LAB'
15         Y = rgb2lab(Y);
16     otherwise
17 %         Y = im2uint8(Y);
18 end%switch
19
20 [m,n,p] = size(Y);
21 isgray = 1;
22 i=1;
23 while(isgray&(i<p-1))
24     for i=1:p-1
25         if Y(:,:,i)==Y(:,:,i+1)
26             isgray = isgray;
27         else
28             isgray = 0;
29         end;%if Y
30     end;%for i
31 end;%while(isgray)
32
33 if(isgray)
34     Y = Y(:,:,1);
35     p=1;
36 end;%if(isgray)
37
38 mn = m*n;
39
40 %Convert to mn*p matrix as data set:
41 for i=1:p
42     X(:,i) = reshape(Y(:,:,i),mn,1);
43 end;%fori=1:p
44
45 %perform desired clustering:
46 timerVal = tic;%Start Timer
47 switch algorithm
```

```
48      case 'k-Means'
49          [L,C,dhistory] = kmeansMark(X,k,dissimilarity,init,seed);
50      case 'PAM'
51           [L,C,dhistory] = PAM(X,k,dissimilarity,init,seed);
52           disp('Kmedoids currently only uses kmeans++ initialization')
53       case 'CLARA'
54            [L,C,dhistory] = CLARA(X,k,dissimilarity,init,T,seed);
55       case 'MATPAM'
56            [L,C,dhistory,~,~,~] = kmedoids(X,k,'Algorithm','pam','
                Distance',dissimilarity);
57            dhistory = cumsum(dhistory);
58            iter = 1;%Dummy
59      otherwise
60          disp('Algorithm not found');
61  end;%switch
62  %TIMER:
63  time = toc(timerVal);
64  %^MUST COME RIGHT AFTER CLUSTERING
65  %
      ************************************************************************

66  %SILH TOOK TO LONG TO CALC: SAVE FOR SINGLE IMAGE ANALYSIS:
67  if (single)
68       disp('Calculating silhouette, please wait');
69       [silh,figs] = silhouette(X,L);
70  else% silh = mean(silh);
71       silh=0;
72       figs=0;
73  end;
74
75   %Convert back to image:
76  Avec = zeros(mn,p);
77  % disp('L= '), disp(L');
78  for i = 1:mn
79       Avec(i,:) = C(L(i),:);
80  end%for i%
81  A = reshape(Avec,m,n,p);
```

```
% A = im2double(A);
MSE = immse(A,Y);

disp('Calculating SSIM, please wait...');
switch HSVorRGB
    case  'HSV'
        A = im2uint8(hsv2rgb((A)));
        Y = im2uint8(hsv2rgb((Y)));
    case 'LAB'
        A = im2uint8(lab2rgb((A)));
        Y = im2uint8(lab2rgb((Y)));
    otherwise
end%switch
% SSIMtest = rgb2gray(im2double(A));
SSIMtest = rgb2gray((A));
% Y = rgb2gray(im2double(Y));
Y = rgb2gray((Y));

SSIM = ssim(SSIMtest,Y);


disp('re-conversion complete')
distance = dhistory';
% outimg = uint8(round(A));
outimg = A;
% figure
% imshow(SSIMtest)
% figure
% imshow(Y);

end
```

Listing 3: Code that calculates dissimilarity for a given cluster

```
function [distance] = Dissimilarity(X,C,dissimilarity)
%Returns a vector of the dissimilarity of the cluster centre
%CHECK COVARIANCE MATRIX, WITHIN OR OUT CLUSTER??
    [n,p] = size(X);
    L = ones(1,n);
```

```
    switch dissimilarity
        case 'Euclidean'
%             D = X-C(L,:);
%             D = sqrt(dot(D,D,2));
            D = pdist2(X,C,'squaredeuclidean');
        case 'Mahalanobis'
            if n<=p%Returns euclidean distance if Mahal not viable
                D = pdist2(X,C,'squaredeuclidean');
            else

              [~,check] = chol(nancov(X));
              if check==0
                  D = pdist2(X,C,'mahalanobis');
              else
                   D = pdist2(X,C,'squaredeuclidean');%USE EUCLIDEAN
                     ??

              end
            end
        otherwise
            D = pdist2(X,C,'euclidean');
    end;% switch type
    distance = D;
end
```

Listing 4: Code to initialise clusters according to k-means++ seeding

```
function [DTot,L,C,Lmed ] = kmeanspp(X,k,dissimilarity,seed)
%Initializes clusters using kmeans++ for dataset. Can be used for k-
  medoids
%too
L = 0;
L1 = 0;
n = size(X,1);%#observations
p = size(X,2);%#variables (MIGHT NOT BE NEEDED)
if (seed ==0)
    rng('shuffle');
else
    rng(seed);%Fix k-means++
```

```
12 end;%if seed
13     while length(unique(L)) ~= k
14         Lmed = zeros(1,n);
15         %Select an observation uniformly random from X as c1:
16         r = 1+round(rand*(n-1));
17         C = X(r,:);
18         Lmed(1,r) = -1;
19 %          DTot = Dissimilarity(X,C,dissimilarity);
20         onevec  = ones(n,1);
21         for i = 2:k
22             %Compute distances from each observation to ci,
                 cumulatively:
23             try
24                     D = X-C(onevec,1);
25                     D = cumsum(sqrt(dot(D,D,2)));
26             catch
27                 disp('Cant use dot(D) method');
28                 D = Dissimilarity(X,C(i-1,:),dissimilarity);
29                 D = cumsum(D);
30             end%trycatch
31             if D(end) == 0
32                 C(i:k,:) = X(ones(1,k-i+1),:);
33                 disp('Returned')
34                 return;
35             end%if D(end)
36             %Select c2 at random from X with probability proportional
                  to
37             %D(x,c)^2:
38             r = rand;
39             ratio = D/D(end);
40             C(i,:) = X(find(r < ratio,1),:);
41             Lmed(1,find(r < ratio,1)) = -i;
42
43                 %Try something new:
44 %                 weights =  D/D(end);
45 %                 [C(i,:),index] = datasample(X,1,'Weights',weights);
46 %                 Lmed(1,index) = -i;
```

```
47          end
48          try
49              DTot(:,1:k) = Dissimilarity(X,C(1:k,:),dissimilarity);
50          catch
51              disp('old school loop');
52              for i =1:k
53                  DTot(:,i) = Dissimilarity(X,C(i,:),dissimilarity);
54              end;%for j
55          end%trycatch
56          [~,L] = min(DTot');
57          Lmed(1,find(Lmed==0)) = L(1,find(Lmed==0));
58      end%while unique
59      disp('kmeans++ complete')
60
61  rng('shuffle')
62  end
```

Appendix B: k-Means, CLARA and PAM MATLAB Code

Listing 5: k-Means Algorithm Code

```
function [L,C,dhistory] = kmeansMark(X,k,dissimilarity,init,seed)
%KMEANS Cluster p-variate data using the k-means algorithm.
%   [L,C] = kmeans(X,k) produces a 1-by-size(X,2) vector L with one class
%   label per column in X and a size(X,1)-by-k matrix C containing the
%   centers corresponding to each class.

L = [];
L1 = 0;
n = size(X,1);%#observations
p = size(X,2);%#variables
distanceM = [];


    %initialize clusters:
    if strcmp(init,'k-Means++')
        [DTot,L,C,~ ] = kmeanspp(X,k,dissimilarity,seed);
    else
        %Random initialisation
         C = 255*rand(n,p);
         %TEST AGAIN:
         DTot = zeros(n,k);
         try
             DTot(:,1:k) = Dissimilarity(X,C(1:k,:),dissimilarity);
         catch
             disp('Old School Loop');
             for i=1:k
                 DTot(:,i) = Dissimilarity(X,C(i,:),dissimilarity);
             end;%fori
         end%trycatch
         %END TEST
         [~,L] = min(DTot');
         disp('Random initialisation complete')
```

```
    end;%if init


     disp('initialisation complete')
     %k-means algorithm.
     dhistory = zeros(1,n);
     iter = 0;
     while any(L ~= L1)
         L1 = L;
         DTot = zeros(n,k);
         for i = 1:k
              l = L==i;
              C(i,:) = sum(X(l,:),1)/sum(l);
              DTot(:,i) = Dissimilarity(X,C(i,:),dissimilarity);
         end
         iter = iter+1;
         [distanceM,L] = min(DTot');
         dhistory(iter) = sum(distanceM);
     end
     dhistory = dhistory(1,1:iter);
     L = L';
     disp('K-means complete')
%end

end
```

Listing 6: PAM Algorithm Code

```
function [L,C,dhistory] = PAM(X,k,dissimilarity,init,seed)
L = [];
Lmedprev = 0;
n = size(X,1);%#observations
p = size(X,2);%#variables
distanceM = [];

%       Always initializes as Kmeans++ (RANDOM TO BE INPUT LATER)
%
  ************************************************************************
```

```
10      [DTot,L,C,Lmed ] = kmeanspp(X,k,dissimilarity,seed);
11           Lmed = Lmed';
12      L = L';
13      Medoids = zeros(k,1);
14      for i = 1:k
15          Medoids(i,1) = find(Lmed==-i,1);
16      end%for i
17      M = X(Medoids(:,1),:);
18  %
    ************************************************************************
19
20
21          [~,L] = min(DTot');
22          L = L';
23          Lmed = L;
24          for i = 1:k
25              Lmed(Medoids(i,1),1) = -i;
26          end;%for
27
28          %
              ************************************************************************
29
30      %k-medoids algorithm
31      [distanceM,~] = min(DTot');
32      Dcheck = sum(distanceM);
33      Dcheckprev = Dcheck + 1;
34      dhistory = zeros(1,n);
35      iter = 0;
36      Lmedprev = zeros(n,1);
37
38      while (not(isequal(Lmedprev,Lmed)) && iter < 100)
39          Lmedprev = Lmed;
40          Lmedprev;
41
42  %               while (Dcheck < Dcheckprev)
```

```
43          %any(Lmed ~= Lmedprev)
44          Dcheckprev = Dcheck;
45          %Within each cluster, test distances and reasign:
46          for i = 1:k
47              %Muster the cluster:
48              Cpos = find(abs(Lmed)==i);
49              Ci = X(Cpos,:);
50              [c,~] = size(Ci);
51              %Between each object in current cluster, test distances
                  and reasign:
52              Di = zeros(c,1);
53  %            ***************************************************
54
55              try
56                  Di(1:c) = sum(Dissimilarity(Ci,Ci(1:c,:),
                      dissimilarity));
57  %                Di
58              catch
59                  %Otherwise do it the old fashioned way:
60                  disp('old school loop');
61                  for j =1:c
62                      Di(j,:) = sum(Dissimilarity(Ci,Ci(j,:),
                          dissimilarity));
63                  end;%for j
64              end;%try
65  %             Di
66              [~,pos] = min(Di);
67  %             pos
68              M(i,:) = Ci(pos,:);
69  %             M
70              Medoids(i,1) = Cpos(pos,1);
71              Medoids;
72  %             DTot(:,i) = Dissimilarity(X,M(i,:),dissimilarity);
73          end
74
75          DTot = zeros(n,k);
76          try
```

```
77                  DTot = Dissimilarity(X,M,dissimilarity);
78              catch
79                  for i=1:k
80                      DTot(:,i) = Dissimilarity(X,M(i,:),dissimilarity);
81                  end%for i
82              end%try
83
84  %             DTot
85              %Reasign elements as needed:
86              iter = iter+1;
87              iter;
88              [distanceM,L] = min(DTot');
89              L = L';
90              Lmed = L;
91              for i = 1:k
92                  Lmed(Medoids(i,1),1) = -i;
93              end;%for
94  %             Lmed'
95  %             distanceM
96              dhistory(iter) = sum(distanceM);
97              dhistory = dhistory(1,1:iter);
98              Dcheck = dhistory(end);
99              Lmed;
100         end;%end while
101 %         DTot;
102 %         L'
103         dhistory = dhistory(1,1:iter);
104         L = L';
105         disp('PAM complete')
106         C = M;
107 end
```

Listing 7: CLARA Algorithm Code

```
1  function [L,C,dhistory] = CLARA(X,k,dissimilarity,init,T,seed)
2  L = [];
3  L1 = 0;
4  [n,p] = size(X);%#n-observations,p-space
5  p = size(X,2);%#variables
```

```
distanceM = [];
ksize = 40 + 2*k;
Sample = [];

dhistory = [];
iter = 0;
C_Best = 0;
DTot_Best = 0;
D_Best = n*p*255;
L_Best = [];
    for i = 1:T
        Sample = [datasample(X,ksize,'Replace',false)];%Get Sample
        %Perform PAM on sample:
        [L,C,~] = PAM(Sample,k,dissimilarity,init,seed);

        %First iteration outside while loop:

         try
             DTot(:,1:k) = Dissimilarity(X,C(1:k,:),dissimilarity);
         catch
             disp('old school loop');
             for i = 1:k
                 DTot(:,i) = Dissimilarity(X,C(i,:),dissimilarity);
             end%for_i
         end%trycatch
         iter = iter+1;
         [distanceM,L] = min(DTot');
         dhistory(iter) = sum(distanceM);
         %Keep the best clusters so far:
         if C_Best == 0;
             disp('hi');
             D_Best = dhistory(iter);
             C_Best = C;
             DTot_Best = DTot;
             L_Best = L;
         end
         if dhistory(iter) < D_Best
```

```
                D_Best = dhistory(iter);
                C_Best = C;
                DTot_Best = DTot;
                L_Best = L;
            end%if
        end%For i=1:T

        %Create final clusters from lowest distance:
        C = C_Best;
        L = L_Best;
        dhistory(iter+1) = D_Best;
%       try
%           DTot(:,1:k) = Dissimilarity(X,C(1:k,:),dissimilarity);
%       catch
%           disp('old school loop');
%           for i = 1:k
%               DTot(:,i) = Dissimilarity(X,C(i,:),dissimilarity);
%           end%for_i
%       end%trycatch
%       iter = iter+1;
%       [distanceM,L] = min(DTot_Best');
%       dhistory(iter) = sum(distanceM);
disp('CLARA Complete');
end%CLARA
```

Appendix C: SAS Code for analysis of data sets

Listing 8: k-Means Algorithm Code

```
ODS GRAPHICS ON;
data sasuser.PAM_clusters (drop = MSE mn distance noise T);
    set sasuser.PAM_CLARA;
    if algorithm = 'PAM';
run;
data sasuser.CLARA_clusters (drop = MSE mn distance noise);
    set sasuser.PAM_CLARA;
    if algorithm = 'CLARA';
run;
```

```
10
11 PROC GLM data=SASUSER.COMBINENOPAM;
12      class Algorithm;
13      model SSIM = Algorithm;
14      means Algorithm;
15 run;
16
17 PROC GLM data=SASUSER.COMBINENOPAM;
18      class Algorithm;
19      model time = Algorithm;
20      means Algorithm;
21 run;
22
23
24 PROC GLM data=SASUSER.CLARA_clusters  PLOTS(MAXPOINTS= 64000);
25      class T;
26      model SSIM=T;
27 run;
28
29 PROC GLM data=SASUSER.CLARA_clusters  PLOTS(MAXPOINTS= 64000);
30      model time=T;
31 run;
32
33
34 PROC GLM data=SASUSER.CLARA_clusters  PLOTS(MAXPOINTS= 64000);
35      class k;
36      model SSIM=k;
37 run;
38
39 PROC GLM data=SASUSER.CLARA_clusters  PLOTS(MAXPOINTS= 64000);
40      model time=k;
41 run;
42
43 PROC GLM data=SASUSER.Pam3000_5to10  PLOTS(MAXPOINTS= 3000);
44      class k;
45      model SSIM=distance k distance*k;
46 run;
```

```
47
48 PROC GLM data=SASUSER.Kmeansbench10k  PLOTS(MAXPOINTS= 64000);
49      class HSVorRGB Dissimilarity;
50      label HSVorRGB = 'Colour space and distance measure';
51      label interaction = 'Colour space and distance measure';
52      title "Interaction plot of color space ";
53      model SSIM = HSVorRGB Dissimilarity HSVorRGB*Dissimilarity;
54      means HSVorRGB*Dissimilarity;
55
56 run;
57
58 ODS GRAPHICS OFF;
59 quit;
```

Bayesian networks applied to forensic science: A South African case study

Iena Petronella Derks 13075782

STK795 Research Report

Submitted in partial fulfillment of the degree BCom(Hons) Statistics

Supervisor(s): Dr. A. de Waal, Dr. P.J. van Staden

Department of Statistics, University of Pretoria

30 October 2017

Abstract

Evidence is often limited, incomplete and vague, thus uncertainty is an extensive component in criminal investigations. This phenomenon makes it difficult to assess and interpret evidence. The aim of this report is to address probabilistic reasoning with forensic evidence, by making use of graphical methods and probabilistic calculus, used in Bayesian networks. During any legal investigation where burned bodies are assessed, the victim's exposure to fire is the main concern. Primary attention is drawn to evaluate the forensic assessment of burned bodies. More specifically, to evaluate whether tongue protrusion can be used as an indicator of vital burning. Therefore, a Bayesian network is constructed to investigate this claim. By making use of inference, the Bayesian network confirms that tongue protrusion can be used as an indicator of vital burning.

Keywords: Bayesian networks; criminal investigations; forensic evidence; probabilistic reasoning; tongue protrusion; uncertainty; vital burning.

Declaration

I, *Iena Petronella Derks,* declare that this essay, submitted in partial fulfillment of the degree *BCom(Hons) Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Iena Petronella Derks

_ _

Dr. A. de Waal

_ _

Dr. P.J. van Staden

_ _

Date

Acknowledgements

I acknowledge the financial support from STATOMET and the Center for AI Research, Meraka Institute, CSIR.

Contents

List of Figures

List of Tables

1 Glossary

Term	Description
Ancestor node	A node that is connected to other nodes, with information flow towards the other nodes, in such a way that there is at least one child with a parent that is a child of this node.
Child node	A node that has one or more nodes connected by a directed link with information flow from another node towards this node.
Descendant node	A node that is connected to other nodes, with information flow from the other nodes, in such a way that there is at least one parent with a child that is a parent of this node.
Forensic science	The application of scientific methods to legal investigations.
Latent variable	A variable that is not directly observed but is rather inferred from other observed variables.
Leaf node	A node with no children.
Manifest variable	A variable that is directly measured or observed.
Parent node	A node that has one or more nodes connected by a directed link with information flow from this node towards other nodes.
Root node	A node with no parents.
Scientific evidence	Evidence that supports a particular hypothesis in forensic science investigations.
Tongue protrusion	A tongue that is extended beyond the dental arch with separated teeth.

2 Notation

$X \subset S$	X is a subset of S.
$X \subseteq S$	X is a subset or equal to S.
$X \cup Y$	The union of X and Y.
$X \cap Y$	The intersection of X and Y.
$\neg X$	Not X.
$P(X)$	Probability of X.
$P(X\|E)$	Probability of X given evidence E.
$X \rightarrow Y$	A directed link.
$\perp\!\!\!\perp$	Independence.

3 Introduction

There is a popular concept in the field of forensic science, proposed by Dr Edmond Locard (1877-1966), known as Locard's Exchange Principle. This states that when there is physical contact between two distinct items, an exchange of some sort will take place. That is, every contact leaves a trace [1]. This poses a challenge to forensic scientists, who need to produce, process, and present accurate data that will assist the court of law in reconstructing past events. There are two different approaches to statistical inference: Frequentism and Bayesianism. The former relates to probabilities calculated by frequencies of events, while the latter represents probability related to knowledge about the event, advocated by Rev. Thomas Bayes (1702-1761) [34]. The approach of this paper is based on the use of an artificial intelligence (AI) system known as Bayesian networks (BNs) to visually represent probabilistic reasoning of variables associated with past events. BNs are frequently used in investigations to aid the forensic scientist in decision making under conditions of uncertainty. During the course of this paper, data will be referred to as evidence since the main focus will be on past events associated with a crime under investigation. It is important to first understand uncertainty, the evaluation of evidence and Bayes' theorem before proceeding to implement BNs in forensic investigations.

Dennis Lindley (2006) states in his book 'Understanding Uncertainty' [21], that uncertainty is the phenomenon in which the outcome can be either true or false, with limited knowledge on whether it is true or false. Often knowledge about an event is limited, therefore incomplete evidence leads to fallible conclusions that need to recover from error. There are three basic forms of uncertainty namely ignorance, physical randomness, and vagueness. Which can be classified as either aleatoric, i.e., when a system behaves in a stochastic manner, or epistemic, which represents the lack of knowledge of the true system [13].

During any legal investigation, evidence may be used to support the nature of a criminal act, or help illustrate the links between elements in a criminal act. Since evidence is often misinterpreted, special care should be given to how it is analysed. Because there is usually insufficient scientific evidence, uncertainty is frequently observed in investigations. Raw evidence does not give more information in itself; its significance needs to be elucidated with propositions and background knowledge [33]. Therefore, the forensic scientist is required to quantify the prior knowledge and consult about the uncertainties associated with inference. BNs applies probabilistic reasoning in the field of forensic science illustrated as a graphical model. As a result, theory and probability calculus is fundamental in the evaluation of scientific evidence.

This research report addresses Bayesian network theory, how to apply this theory to provide an efficient study of the evaluation of evidence in a forensic context, as well as give practical examples. The report will conclude with a practical application of Bayesian networks in a real-life forensic investigation.

4 Bayesian networks

A BN is a probabilistic model based on directed acyclic graphs (DAG) in which a set of random variables, connected by directed links, make up the nodes in the network. Rev. Thomas Bayes (1702-1761) developed a theory for updating probabilities when new evidence becomes available. Both discrete and continuous probability distributions can be addressed by making use of this theory [9]. When working with discrete probability distributions, Bayes' theorem uses the conditional and marginal probabilities of events C and D, with background information I.

$$Pr(C|D,I) = \frac{Pr(C,D|I)}{Pr(D|I)} \tag{1}$$

Conditional probability tables (CPTs) are constructed for each node in the network to quantify the effect one node has on another. BNs represent the qualitative relationships between variables, in other words, it illustrates the causal connections between variables [33].

In order to apply and interpret the propagation of uncertainty in a BN, a few important definitions and terminology must be understood - both in the fields of graph theory and probability calculus.

4.1 Causal graphs

Sewall Wright (1889 - 1988), a geneticist, was the first to use causal graphs, however, he referred to these graphs as 'path diagrams' [36]. A causal graph has directed paths that represent causal relationships between nodes in a diagram. Causal networks consist of interrelated nodes connected by directed links and are known as a directed graph [7].

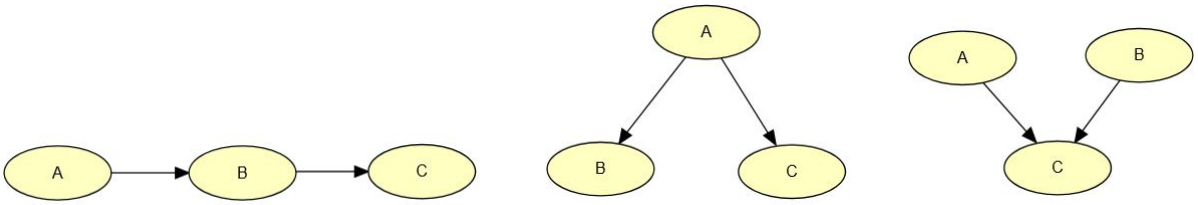
4.1.1 Connections in a causal graph

Variables are represented by nodes and the link between two nodes represent the relationship. Directed graphs that consist of directed cycles represent mutual causation. Causal graphs that contain directed cycles are known as acyclic causal graphs. Therefore, a graph that is directed and acyclic is known as a directed graph [28].

Definition 1. Directed acyclic graph

A directed graph that consist of a topological order with a sequence of nodes connected by directed links.

Kinship terminology is used to characterise the relationships in a DAG: Consider a set of directed links between three nodes, $A \rightarrow M \rightarrow C$: A is a parent of M and M is a child of A. C is a descendant of A and A is an ancestor of C. Tree terminology (e.g., root and leaf) is also used to describe nodes. Therefore, A is a root node, representing original causes, and C is a leaf node, representing final effects.



(a) Serial connection. (b) Diverging connection. (c) Converging connection.

Figure 1: Possible connections in causal graphs.

Every DAG has at least one root and at least one leaf [28]. Causal graphs have three main connections, namely *serial connections*, *diverging connections* and *converging connections*.

Serial connections

Consider the causal graph between three nodes illustrated in Figure 1a. In this particular connection, A causes B, and B causes C. Therefore, serial connections represent conditional independence [19]. When evidence of C is introduced to the network, C will influence A through B. If B's state is known, the path from A to C is blocked, resulting in independence between A and C. If this is the case, A and C is said to be *d-separated* given B. A variable is instantiated when the state of the variable is known [27].

Diverging connections

The causal graph illustrated in Figure 1b is known as a diverging connection, often referred to as a common cause graph. The common cause node is A, shared by B and C. Diverging connections is used when evidence is transmitted to the children of A when the state A is unknown. The same conditional independence structure used in the chain rule (Theorem 8) is applicable to common cause:

$$P(C|A \cap B) = P(C|B) \equiv A \perp\!\!\!\perp C|B$$

Converging connections

Figure 1c illustrates a basic converging connection in causal graphs. Converging connections are represented in a v-shape, where a particular node (effect) has two causes. Therefore, converging connections are known as common effect [19]. The parent nodes, A and B are said to be independent if there is no prior knowledge known about B, except what is deduced based on knowledge of A and C. Possessing knowledge of a cause for a particular event does not necessarily indicate knowledge about other possible causes. However, if the consequences of an event are known, then the knowledge of a particular cause may influence information about another cause. This concept is known as *explaining away* [27].

4.1.2 Information flow in a causal graph

Information in a causal graph flows along paths between variables. Information processing relates to the direction of flow. Depending on the type of connection (as described in Section 4.1) and instantiation of variables, paths can connect or block variables. Connected variables have a dependence and unconnected variables are independent. This concept is known as *d-separation*.

Definition 2. D-separation

Nodes X and Y is said to be d-separated if for all links between X and Y, there is another node Z such that either

- *Z is instantiated, and the connection in the causal graph is either serial or diverging; or*
- *Z has not received evidence and the causal graph is converging.*

Nodes X and Y are d-connected if they are not d-separated.

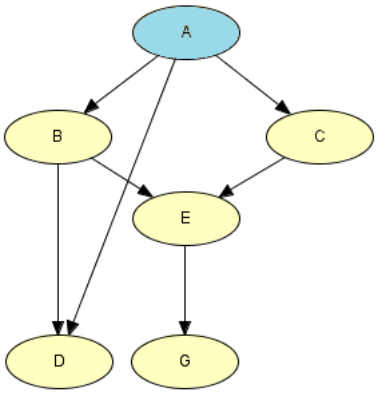


Figure 2: D-separation by node A.

Consider the causal graph shown in Figure 2. Notice that nodes B and C are d-separated by A.

Node A blocks paths $B - E - C$ and $B - A - C$.

Another important property that provides information on information flow in causal graphs is the *Markov blanket*. The Markov blanket for a variable A states that all information about A is contained within this blanket (set of nodes). In short, every node is only dependent on its parents, children and children's parents.

Definition 3. Markov Blanket

A set of parent, children and common causes of a particular node.

When instantiated, the Markov blanket has an interesting property, the node of interest becomes d-separated from the other nodes present in the network (see Figure 3) [27].

The Markov blanket for node A in Figure 3 is {B,C,D}.

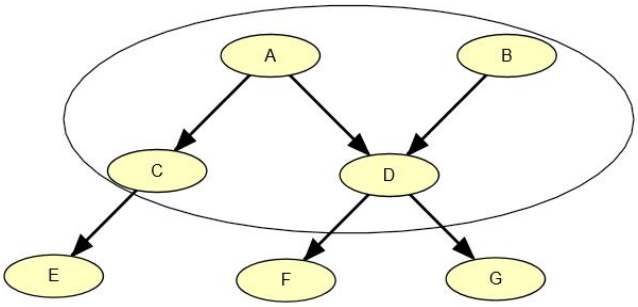


Figure 3: Markov blanket for node A.

4.2 Probability theory

4.2.1 Probability Rules

There are two distinct approaches to statistics [34]: Frequency type probability, advocated by John Venn [35] and belief type probability, promoted by Reverend Thomas Bayes [3]. The difference between frequency type probability and belief type probability lies in the definition of probability used for each approach. Frequency type probability is the case of repeated measurements [16]. The central limit theorem is a tool for frequency type probability, as well as laws for large numbers since observed characteristics stabilize with repeated trials. Belief type probability is extended to cover degrees of certainty about statements. Therefore, probabilities are related to a subjective degree of belief. It is measured by personal assessment or logical probability. Probability calculus is used to represent the strengths of belief [19]. Bayes' theorem is used in belief type probability to show how new evidence influence prior probabilities. The frequency principle aids in deciding whether to use frequency type or belief type probability: When only the frequency is known about an outcome, it is advised to use the frequency type probability approach. However, in other situations, belief type probability is used to assess the outcome of an event. Frequency type probabilities are related to risk, whereas belief type probabilities are related to uncertainties [16]. Note that the formulas in this section are adapted from Modern Business Statistics with Microsoft Excel (2014) [2] and Probability theory: The logic of science (2003) [18].

The degree of uncertainty can be measured by assigning a probability $P(A)$ to each event $A \subseteq S$. Probabilities must adhere to the following three axioms [27] [28]:

Fact 4. *S is the event space of all possible events. Therefore, if event X is in S, then: $P(X) + P(\neg X) = P(S) = 1$*

Fact 5. *The event X must have a positive probability: $P(X) \geq 0$ for all $X \subseteq S$.*

Fact 6. *If event X and Y are mutually exclusive, the sum of the individual events equals the probability of the combined event: If $X \subseteq S$, $Y \subseteq S$ and $X \cap Y = \varnothing$, then $P(X \cup Y) = P(X) + P(Y)$ [27].*

Theorem 7. *Total probability [19]*

Assume $X \subseteq S$, and for any i and j $X_i \cup X_j = \varnothing$. Then,

$$P(S) = \sum_i P(X_i),$$

This can be adapted for a single event Y instead of the whole event space. Therefore, under the above conditions, and if $\forall i X_i \neq \varnothing$,

$$P(Y) = \sum_i P(Y|X_i)$$

The intersection of two events X and Y:

The intersection of two events is the event containing points that belong to both events simultaneously. Therefore, the probability of intersection between event X and Y is given by [2]:

$$\begin{aligned} P(X,Y) &= P(X \cap Y) \\ &= P(X) + P(Y) - P(X \cup Y) \end{aligned}$$

The sum rule

The union of two events is the event containing points belonging to event A or event Y or both. Therefore, the probability of the union of event X and Y [2]:

$$P(X \cup Y) = P(X) + P(Y) - P(X \cap Y)$$

If the events are mutually exclusive, the union of event X and Y become:

$$P(X \cup Y) = P(X) + P(Y)$$

Conditional probability

The conditional probability of an event occurring given evidence E, can be calculated as a ratio of a joint probability to a marginal probability:

$$P(X|E) = \frac{P(X \cap E)}{P(E)} = \frac{P(X,E)}{P(E)}$$

The product rule

The multiplication law is used to compute the intersection of two events. Thus, the multiplication law depends on the conditional probability of event X and Y [18]:

$$P(X \cap Y) = P(X)P(Y|X)$$

Theorem 8. *Chain rule*

Given three events X, Y, Z. The chain rule divides the probabilistic influence of event Z on event A across the different states of a third event Y [19]:

$$P(Z|X) = P(Z|Y)P(Y|X) + P(Z|\neg Y)P(\neg Y|X),$$

Bayes' theorem can be derived through the use of the chain rule:

Event Y, posterior probabilities $P(X_1|Y)$ and $P(X_2|Y)$. From the definition of conditional probability:

$$P(X_1|Y) = \frac{P(X_1,Y)}{P(Y)} = \frac{P(X_1 \cap Y)}{P(Y)} \tag{2}$$

However,

$$P(X_1 \cap Y) = P(X_1)P(Y|X_1) \tag{3}$$

To find $P(Y)$, note that event Y can occur in two ways: $(X_1 \cap Y)$ and $(X_2 \cap Y)$. Therefore,

$$\begin{aligned} P(Y) &= P(X_1 \cap Y) + P(X_2 \cap Y) \\ &= P(X_1)P(Y|X_1) + P(X_2)P(Y|X_2) \end{aligned} \tag{4}$$

Substituting equations 3 and 4 into equation 2, Bayes' theorem for a two event case is obtained:

$$P(X_1|Y) = \frac{P(X_1)P(Y|X_1)}{P(X_1)P(Y|X_1) + P(X_2)P(Y|X_2)} \quad (5)$$

where,

$P(X_1|Y)$ is the posterior probability,$P(Y|X_1)$ relates to the likelihood, and $P(X_1)$ is the prior probability.

Equation 5 can be written in a general form to accommodate cases with more than two events:

$$P(X_i|Y) = \frac{P(X_i)P(Y|X_i)}{P(X_1)P(Y|X_1) + P(X_2)P(Y|X_2) + \ldots + P(X_n)P(Y|X_n)}$$

Example on Bayes' rule

Scenario: Meningitis causes stiff necks with probability 0.5. Having meningitis has a prior probability of 0.00002, and having a stiff neck has a prior probability of 0.05. What is the probability of having meningitis given that you have a stiff neck?

Let s = patient has meningitis

Let s = patient has stiff neck

$P(s|m) = 0.5$

$P(m) = 0.00002$

$P(s) = 0.05$

Therefore,

$$P(m|s) = \frac{P(s|m)P(m)}{P(s)} = \frac{(0.5)(0.00002)}{0.05} = 0.0002$$

4.2.2 Conditional independence

Probabilistic models allow the use of probabilistic inference to compute a probability distribution. The full joint probability distribution (FJPD) tabulates probabilistic inference values. However, For N variables, each with k values, the joint distribution has K^N numbers. Therefore, BNs are used to represent joint distributions using independence and conditional independence.

Independence

Two events are independent when the probability of one event remains unchanged if conditioning is applied to the other:

$$Y \perp\!\!\!\perp X \quad = \quad P(Y|X) = P(Y)$$

This can be generalized for *conditional independence*:

Two events are independent given an additional event:

$$Y \perp\!\!\!\perp X|Z = P(Y|X,Z) = P(Y|Z)$$

Definition 9. *Full joint distribution*: *The full joint distribution of two random variables, M and Y, is the probability distribution containing the probabilities for every M and Y contained in a specific range of values.*

Using a full joint distribution; the product rule, sum rule and Bayes' theorem can be used to create any combination of joint and conditional probabilities. Obtaining a full joint distribution is difficult as it requires a vast amount of data even for few variables. The conditional independence assumptions make it possible to reduce the required probabilities for the joint distribution of a BN by making use of the properties of causal graphs.

4.3 Construction of BNs

BNs represent possible connections between attributes in events where uncertainty is present in the domain. A key feature of a BN is that it allows the user to move from prior to posterior probabilities as evidence becomes available. The nodes in a BN represent a set of random variables, $X = X_1, ..X_i, ..X_n$. Directed links connect pairs of nodes, $X_i \rightarrow X_j$. These links represent the direct dependencies between variables. The only constraint on the links in a BN is that there must not be any directed cycles, therefore a BN is a DAG. This section will focus on how to construct a BN.

Definition 10. *A BN consists of the following:*

- *A set of nodes connected by links.*
- *Each node has a finite set of disjoint states.*
- *The nodes and the links form a DAG.*
- *Every node X with parents $Y_1, \ldots, Y_n$, has a conditional probability table $P(X|Y_1, \ldots, Y_n)$.*

4.3.1 Variables

The first step is to determine the variables of interest. Note that a variable can take on exactly one value at a time, therefore, the values are mutually exclusive. Three types of discrete nodes exist: Boolean

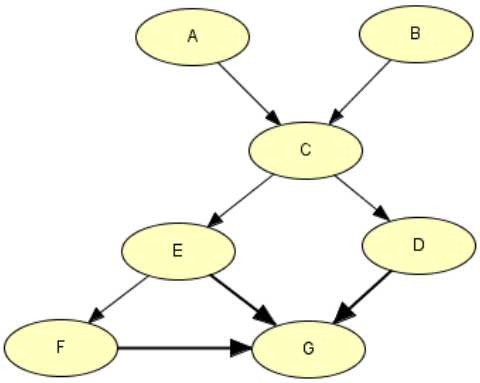


Figure 4: Typical Bayesian network.

nodes, ordered nodes, and integral nodes. Values should be chosen such that each value represents the domain in an efficient manner.

4.3.2 Topology

The network structure should express qualitative relationships between variables. Thus, two nodes should be directly connected if one affects or causes the other, the link between the two nodes indicates the direction of flow. The topology of a BN takes the same form of that of a causal graph, as seen in Section 4.1. Figure 4 illustrates a typical BN, using the family metaphor specified in Section 4.1, A and B are parent nodes of C, depicting a diverging connection. While C is a parent node of E and D, illustrating a converging connection. Nodes A and B are root nodes, and G is a leaf node. The descendants of A are C, D, E, F and G. Node F has the ancestor nodes E, C, A and B. The Markov blanket for D is $\{C, E, F, G\}$.

4.3.3 Conditional probability table

A variable can either be discrete or continuous. A conditional probability distribution for each node quantifies the relationship between two or more nodes. Hence, a conditional probability table (CPT) should be constructed to show the numerical probabilities. The CPT is defined for a set of discrete variables to represent the marginal probability of a variable taking into account the certainty of other variables in the network. Possible combinations of each node should be specified in order to compute the CPT, i.e., the parent, child, descendant and ancestor nodes should be identified - this is done in Section 4.3.2. For each instantiation of parent nodes, the probability of the child node taking the values needs to be specified. The probabilities to specify in Figure 4 are, $P(A)$, $P(B)$, $P(C|A,B)$, $P(E|C)$, $P(D|C)$, $P(F|E)$, and $P(G|D,E,F)$. A node's CPT summarizes the probability of possible values, conditional upon parent node values. Once the values for all the CPTs are added, the BN is 'parameterised'. When a BN is parameterised, probabilistic inference is used to calculate probabilities of numerous possible

hypothesis, given prior evidence, as well as predicting the states of evidence nodes not yet detected [20]. Figure 5b represents typical CPTs associated with different nodes.

4.4 Inference with Bayesian networks

The primary task of a probabilistic graphical model is belief updating. That is, the computation of posterior probability distributions for a set of nodes, given a set of evidence nodes. As evidence with respect to a specific node becomes available, beliefs of the other nodes in the network are updated. Therefore, inference in BN is flexible [19]. BNs can be conditioned upon any set of nodes, supporting any direction of reasoning. There exists three main types of reasoning: diagnostic reasoning, concerned with inference from effect to cause; predictive reasoning, which applies to reasoning from cause to effect as new knowledge becomes available; and inter-causal reasoning, which relates to mutual causes of a common effect. BNs can be used to solve 'what-if' questions - such as the probability of observing evidence if a hypothesis is false [20]. Inference in BNs is illustrated by making use of an example discussed in Bayesian Artificial Intelligence (2010) [19]:

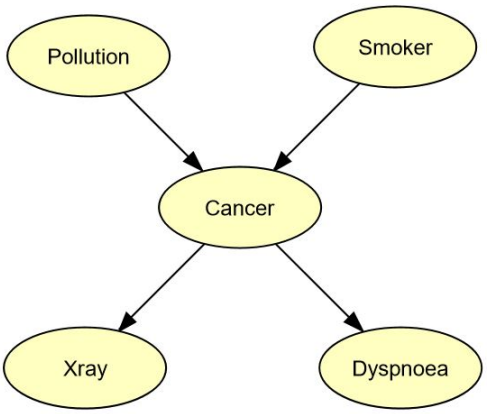
Example 1

A doctor treats a patient suffering of dyspnoea. The patient is worried she has lung cancer. The doctor has prior knowledge that other diseases, such as bronchitis, show similar symptoms. The doctor knows that being a smoker can increase the chances of cancer and bronchitis, as well as that the patient has been exposed to some sort of air pollution. The positive result of an X-ray could indicate lung cancer.

The propositions are: 'Patient exposed to pollution' ('*Pollution*'), 'Patient identified as a smoker' ('*Smoker*'), 'Lung cancer present' ('*Cancer*'), 'Patient suffering of dyspnoea' ('*Dyspnoea*'), and 'X-Ray result' ('*XRay*'). Table 1 contains the preliminary node choices for building the BN. Figure 5 illustrates the BN for this scenario, where Figure 5a is the BN itself, Figure 5b is the conditional probability table for this scenario, and Figure 5c is the monitor windows associated with the BN before any belief updating due to new evidence in network.

Node name	Type	Value
Pollution	Binary	$\{low, high\}$
Smoker	Boolean	$\{T, F\}$
Cancer	Boolean	$\{T, F\}$
Dyspnoea	Boolean	$\{T, F\}$
X-ray	Binary	$\{low, high\}$

Table 1: Preliminary node choices.



(a) BN for known variables.

Pollution

Low	0.9
High	0.1

Smoker

True	0.3
False	0.7

Cancer

Smoker	True		False	
Pollution	Low	High	Low	High
True	0.03	0.05	0.001	0.02
False	0.97	0.95	0.999	0.98

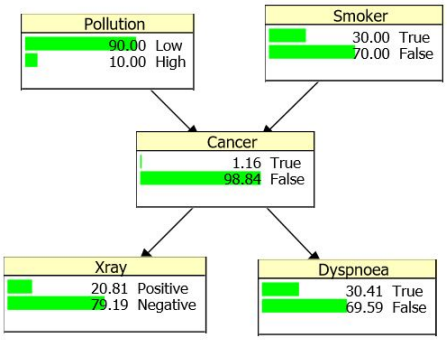
Xray

Cancer	True	False
Positive	0.9	0.2
Negative	0.1	0.8

Dyspnoea

Cancer	True	False
True	0.65	0.3
False	0.35	0.7

(b) CPT for known variables.

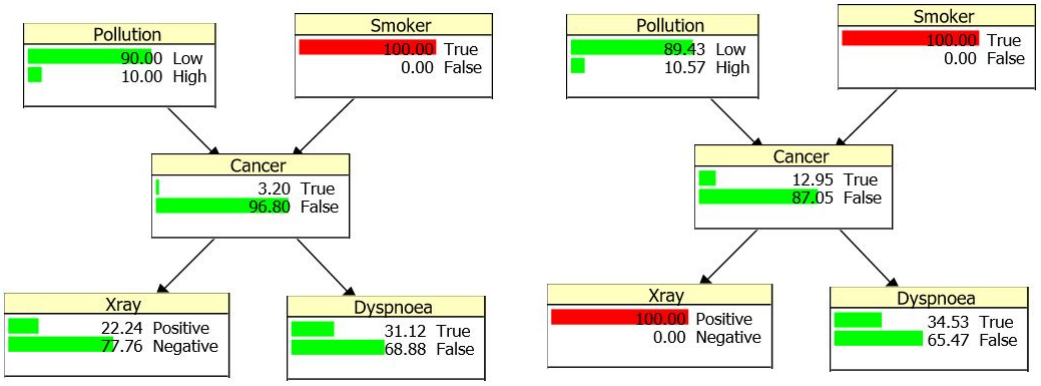


(c) Marginal probabilities for variables.

Figure 5: BN, CPT and marginal probabilities for lung cancer patient.

Figure 6 represents the 'what-if' analysis by updating beliefs in the network. For simplicity, only one to two node(s) will receive prior evidence. Figure 6a illustrates the effect of adding evidence to the *'X-ray'* node. Say the doctor receives confirmation that the patient is a smoker. Notice how the probabilities of the '*Cancer*' node changes. Before the evidence was added, the '*Cancer*' node showed a probability of 0.0116 to be positive and a probability of 0.9884 to be negative. As soon as the new evidence was added, the probability of getting a positive result for cancer increased to 0.032 and the probability to get a negative cancer result decreased to 0.9680. Suppose, now the doctor receives confirmation that the X-ray showed positive signs of cancer, as shown in Figure 6b. The new evidence increases the probability of having cancer to 0.1295 and decreases a false outcome to 0.8705. Therefore, the what-if analysis was done on the assumption of 'what-if the patient is a smoker' and 'what-if the patient is a smoker and has a positive X-ray result'. Belief updating is done using Bayes' theorem (see the example on Bayes' theorem

in subsection 4.2.1) Note the belief updating is done in Hugin Lite 8.4[1].



(a) Belief updating due to '*Smoker*' node.

(b) Belief updating due to '*Smoker*' and '*Xray*' nodes.

Figure 6: Belief updating given prior knowledge.

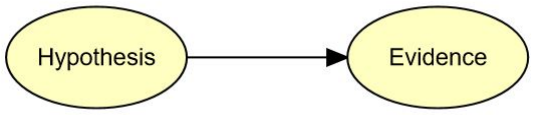
5 Bayesian networks applied to forensics

Consider any forensic investigation, during the preliminary investigation evidence is collected. This evidence relies on forensic sciences to be analyzed, and used to support the nature of a criminal act, or used to illustrate links between certain evidence elements in a formal investigation. When comparing elements in a forensic investigation, forensic scientists investigate the links between interrelated evidence pieces by analyzing various traces transferred during the act [33]. Traces found at the crime scene is known as trace evidence (see Glossary for formal definition). The interpretation of scientific evidence should be done carefully, emphasis should be on questions such as 'what do the results mean in this case?' [17]. Therefore, theory and statistical probability applications form an important foundation for evaluating scientific evidence. Probabilistic reasoning is used to explain scientific evidence in a numerically understandable way. Forensic scientists use probabilities to measure the uncertainty of evidence. Various methods of reasoning have been proposed to assist in understanding dependencies present in evidence [12]. As discussed earlier, BNs represent relationships between variables where uncertainty is present (refer back to Section 4). BNs aid the forensic analyst to describe the problem whilst communicating information about the topology of the situation, as well as calculating the effect of the new evidence on other nodes within the network [15].

5.1 Evaluation of scientific evidence using BN

Evaluation of evidence starts as soon as the forensic investigator is assigned to a case. Ideas relating to the value of evidence and propositions are made by the forensic scientist, this is known as pre-assessment.

[1]Hugin 8.4 (x64) Copyright (c) 1995-2017 Hugin Expert A/S. All Rights Reserved



(a) BN for evidence.

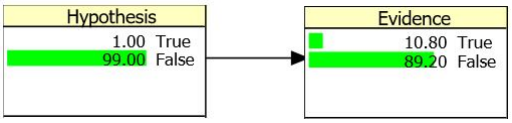
Hypothesis

True	0.01
False	0.99

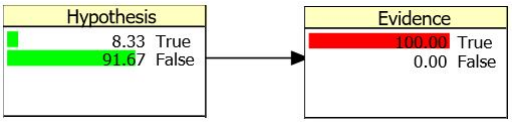
Evidence

Hypothesis	True	False
True	0.9	0.1
False	0.1	0.9

(b) CPT for Evidence.



(c) Prior probabilities for evidence.



(d) Belief updating for evidence.

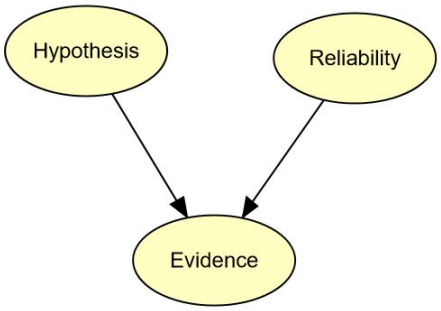
Figure 7: Evidence applied to forensic investigation example.

Pre-assessment requires an appreciation of the nature of the criminal act such that a framework can be developed for examination purposes [10], to facilitate a logical decision. The forensic investigator should consider probabilities of the evidence gathered during the investigation. Aitken and Gammerman (1989) proposed the use of a probabilistic graphical model in assessing scientific evidence [33]. BNs can be used to assess human reasoning since it provides a normative model for evidential reasoning [29].

Example 2

A generic scenario described by Lagnado, Fenton and Neil [20] will be used to illustrate the use of BNs in forensic investigations. Note that the scenario will illustrate BNs applied to evidence and evidence-reliability. By applying BNs to both evidence and evidence-reliability, probabilistic relations between hypothesis and evidence will be illustrated. Consider the hypothesis: 'Suspect committed crime', and evidence: 'Suspect identified by victim'. Evidence applied to forensic investigations involves the relationship between the stated hypothesis and the evidence given. Typically, the hypothesis is a proposition made about the case, and the evidence is an observation, such as DNA evidence and witness testimonies. The causal process is represented by the link between the hypothesis and the evidence presented [20]. To illustrate this, consider the hypothesis that a suspect committed a crime, this can result in either true or false. If the hypothesis is proven to be true, the probability of the victim identifying the suspect will increase. Now, using Bayes' rule, a positive identification increases the probability that the suspect did indeed commit the crime. This is illustrated in Figure 7. Where, Figure 7a represents the BN of cause→effect. Where the hypothesis is the cause and the evidence the effect. Figure 7b shows the CPT filled with hypothetical probabilities since this is a generic example. Figures 7c and 7d shows the prior probabilities and the updated probabilities as evidence become available, i.e., the suspect positively identified the victim.

Suppose another node is now added to the BN, i.e., the reliability of the testimony, denoted by '*Reliability*' (see Figure 8). This illustrates that the addition of an extra node can influence the evidence report. Note that measuring human testimony can be inaccurate, therefore, there is a degree of fallibility [20]. Assume '*Reliability*' is causally independent on the hypothesis. The evidence node has two parent nodes, i.e., '*Hypothesis*' and '*Reliability*', as shown in Figure 8a. Suppose the evidence node has the following two explanations: the suspect was wrongfully identified and the victim positively identified the suspect. To calculate the influence evidence has on the two states, the prior probabilities and the conditional probabilities need to be estimated, given the parent nodes. Therefore, four possible states exist: (i) hypothesis true and reliability true, (ii) hypothesis false and reliability true, (iii) hypothesis true and reliability false, and (iv) hypothesis false and reliability false.



(a) BN for evidence-reliability

Hypothesis

True	0.1
False	0.9

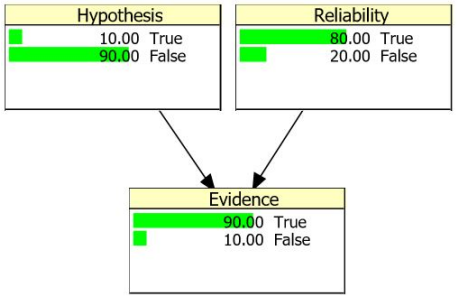
Evidence

Reliability	True		False	
Hypothesis	True	False	True	False
True	1	1	0.5	0.5
False	0	0	0.5	0.5

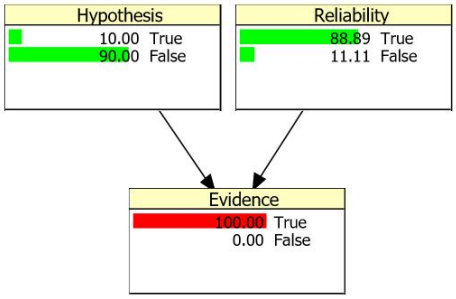
Reliability

True	0.8
False	0.2

(b) CPT for evidence-reliability.



(c) Prior probabilities for evidence-reliability.



(d) Belief updating for evidence-reliability.

Figure 8: Evidence-reliability applied to forensic investigation example.

6 Application

During any legal investigation where burned bodies are assessed, the victim's exposure to fire is the primary concern [6]. It is important to establish whether exposure to fire was before or after death, as well as to establish the circumstances surrounding the death as this can indicate suicide, homicide or accident. Heat exposure can be measured with external and internal methods, however, for this study, internal exposure is measured. Although different internal exposure methods exist, soot deposits in the respiratory tract and the percentage of carboxyhaemoglobin (COHb%) are the main focus of the comparative study.

During the period of January 2007 to December 2009, 107 burned corpses were examined at the Medico Legal Laboratory in Pretoria, South Africa. The data used in the study is obtained from Prof Herman Bernitz[2]. The tongue position was reported as either protruded, not protruded, destroyed or not known. However, for the purpose of this research report, a destroyed tongue is treated as a missing value and when no information on the position of the tongue is known, the soot deposit is used to determine the tongue position to be either a missing value or a non-protruding tongue. If soot deposit returns a positive result, it is treated as a missing value and if a negative result is returned, tongue position is reported as non-protruding. Thereafter, the missing values are taken out of the data, resulting in 74 observations to be used. That is, 33 observations returned missing values and are therefore eliminated from the dataset. The remaining observations are then converted to binary data (see subsection 6.2 for a full discussion of variables). Non-parametric statistical analysis and logistic regression will be used to test the relationship between tongue protrusion and soot deposits as well as the relationship between tongue protrusion and COHb%. Thereafter, a BN will be constructed to visually represent and examine the case study.

6.1 Comparative study

A comparative study is done to test the relationship between tongue protrusion and the other manifest variables in the dataset. That is, the association between tongue protrusion and soot presence and between tongue protrusion and COHb percentage is tested. This section will be divided into two subsections. The following descriptive statistics relate to the case study: Tongue protrusion is reported in 63 of the 107 victims, 30 out of the 107 victims showed no tongue protrusion, while 14 cases reported the tongues to be either destroyed or not known, and is therefore treated as missing values. Further investigation reported that in 96 of the 107 cases, a positive soot deposit result is reported, 8 of the cases resulted in a negative soot deposit and in three of the cases soot analysis was not possible and is therefore treated as missing values. Since a COHb threshold of > 10 is considered positive for the inhalation as a result of fire [4], 66

[2]Department of Oral Pathology and Oral Biology,
School of Dentistry, University of Pretoria,
P.O. Box 1266, Pretoria 0001, South Africa

out of the 107 cases is reported to be positive, 17 cases returned a negative COHb result and 24 cases report missing values. Note that these frequencies are computed in SAS 9.4[3] by making use of the FREQ procedure.

6.1.1 Tongue protrusion vs. the presence of soot

Internal signs of heat exposure, such as soot deposits in the respiratory tract, oesophagus and stomach indicate vital burning [4]. It is therefore important to test whether there is a relationship between tongue position and the presence of soot in the respiratory tract.

H_0 : Tongue protrusion not associated with soot presence.

H_a : Tongue protrusion associated with soot presence.

Fisher's exact test is used to test the association between tongue protrusion and soot deposits in the respiratory tract. The LOGISTIC procedure in SAS 9.4 is used to calculate the associated odds ratio and Wald Chi-square confidence interval. The odds ratio is reported to be 6.354 and the Wald Chi-square confidence interval is $(1.153, 35.008)$ with a Wald Chi-square statistic of 4.5105. The Wald Chi-square confidence interval is used to test the hypothesis. Note the $p-value$ is 0.0337. Since $1 \notin (1.153, 35.008)$, the null hypothesis is rejected. Therefore, there is a statistically significant relationship between tongue protrusion and soot deposits.

6.1.2 Tongue protrusion vs. the percentage COHb

High blood level values of carboxyhaemoglobin is also an indication of internal burning, and will be tested against the tongue position to establish whether there is a relationship between the two variables. However, unlike the previous comparison (see subsection 6.1.1), where the association between the two variables was the only hypothesis test done, this subsection will contain four different hypothesis tests. These tests will include a test for location, variation, distribution of COHb% between tongue protruded and tongue not protruded and lastly a test for association between COHb% and tongue protrusion.

H_0 : No difference in median COHb% between tongue protruded and tongue not protruded.

H_a : Significant difference in median COHb% between tongue protruded and tongue not protruded.

The first test is conducted to test whether there is a significant difference in location. A Mann-Whitney

[3]Copyright (c) 2002-2012 by SAS Institute Inc., Cary, NC, USA. All Rights Reserved

test is done in SAS 9.4 with the NPAR1WAY procedure. The Mann-Whitney test has a $p-value$ of 0.2449, which indicates that there is no statistically significant difference in location between tongue protruded and tongue not protruded in victims with COHb% concentration in blood. Figure 9a illustrates the location of median COHb% for tongue protruded and tongue not protruded.

H_0 : No difference in spread of COHb% between tongue protruded and tongue not protruded.

H_a : Significant difference in spread of COHb% between tongue protruded and tongue not protruded.

A Siegel-Tukey test is done to establish whether there is a significant difference in variance COHb% between tongue protruded and tongue not protruded. The Siegel-Tukey test has a p-value of 0.2578, thus the null hypothesis is rejected at a 10% level of significance. Therefore, indicating there is no difference in spread of COHb% between tongue protruded and tongue not protruded.

H_0 : No difference in distribution COHb% between tongue protruded and tongue not protruded..

H_a : Significant difference in distribution COHb% between tongue protruded and tongue not protruded.

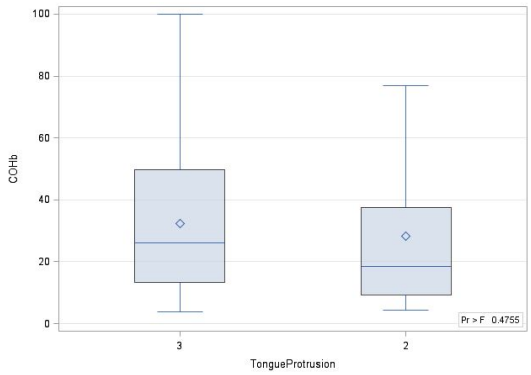
A Kolmogorov-Smirnov test is used to detect whether there is a significant difference in the distribution. The Kolmogorov-Smirnov Two-Sample Test has a $p-value$ of 0.5043, indicating no statistical significant difference in distribution between tongue protruded and tongue not protruded. The empirical distribution of COHb between tongue protruded and tongue not protruded is shown in Figure 9b.

H_0 : Tongue protrusion not associated with COHb%.

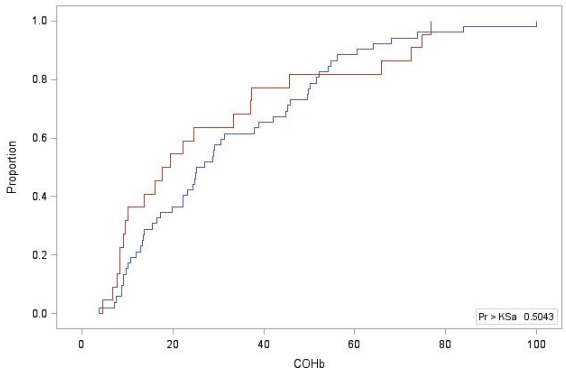
H_a : Tongue protrusion associated with COHb%.

Fisher's exact test is done to determine if victims with tongue protrusion are more likely to have COHb% concentrations more than 10 than victims with no tongue protrusion. Fisher's exact test returned a two-sided $p-value$ of 0.1242, implying that the null hypothesis is not rejected. Therefore, there is no significant association between COHb% in blood level values and tongue protrusion.

The above results correspond with the results from the article published by Bernitz *et* al [4]. On the grounds that soot deposits and a protruded tongue are often seen in fire-related deaths, the probability of both occurring simultaneously is very high. Therefore, results in a statistically significant relationship between these variables. This does not necessarily mean that the two variables are equally valid. In a



(a) Boxplot of COHb level between tongue protruded and tongue not protruded.



(b) Empirical distribution of COHb levels between tongue protruded and tongue not protruded.

Figure 9: Distribution of COHb.

letter to the editor of International Journal of Legal Medicine, Michael Bohnert [5] mentioned that there is no pathophysiological explanation for tongue protrusion to be an indicator of vital burning. Based on this feedback, an alternative approach was considered. Rather than using non-parametric statistical analysis, the focus will be on prediction. In order to do this, vital burning must be included in a predictive model. By including the latent variable to the analysis, it becomes possible to test the relationship between tongue protrusion and vital burning.

This poses a challenge as vital burning cannot be observed. A statistical technique capable of handling latent variables is needed to establish whether there is a relationship between tongue protrusion and vital burning.

In the next section, a BN will be implemented to test the relationship between tongue protrusion and vital burning.

6.2 BN applied to case study

In the previous section, statistical analysis was conducted to test if tongue protrusion is a sign of vital burning. However, since logistic regression and non-parametric tests are limited to observed variables, the relationship cannot be tested to full extent. BNs provide fundamental advantages for dealing with missing values, i.e., latent variables [9]. Aside from this, the predictive capabilities of BNs are also of interest. Therefore, a BN will be constructed to test whether tongue protrusion is a sign of vital burning.

Consider the data obtained. A BN is constructed to support the explanatory inferences supporting the evidential reasoning in the case study. The methodology discussed in Section 4.3 is used to construct the BN to represent the data.

6.2.1 Variables

The first step is to identify the variables and relevant propositions. The propositions in question are:

- Soot deposits in the respiratory tract, oesophagus and stomach (*SootPresence*),
- High blood values of carboxyhaemoglobin (*COHb*),
- Protrusion of the tongue (*TonguePosition*),
- Vital burning (*VitalBurning*)

The following variables are identified as evidence nodes: *'SootPresence'*, *'COHb'*, and *'TonguePosition'*, since the objective of the study is to determine whether the variable *'TonguePosition'* indicate vital burning. Therefore, *'VitalBurning'* is a query node. As stated previously, *'VitalBurning'* is a latent variable, as there is no recorded/observable data related to this node. The preliminary node choices for the case study is shown in Table 2.

Node name	Type	Value
SootPresence	Binary	$\{0, 1\}$
COHb	Binary	$\{0, 1\}$
TonguePosition	Binary	$\{0, 1\}$
VitalBurning	Binary	$\{0, 1\}$

Table 2: Preliminary node choices for case study .

The data is converted to binary data so that each of the evidence nodes have two states:

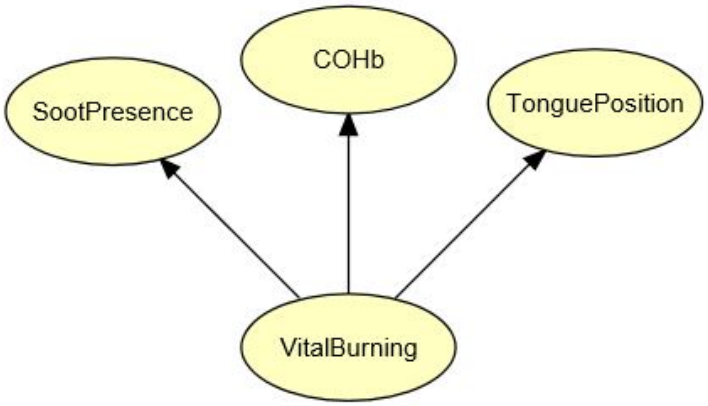
$$\text{SootPresence} = \begin{cases} 1 & \text{for Positive soot level} \\ 0 & \text{for Negative soot level} \end{cases}$$

$$\text{COHb} = \begin{cases} 1 & \text{for Percentage COHb } \geq 10\% \\ 0 & \text{for Percantage COHb } < 10\% \end{cases}$$

$$\text{TonguePosition} = \begin{cases} 1 & \text{for Tongue protruded} \\ 0 & \text{for Tongue not protruded} \end{cases}$$

6.2.2 BN structure

The next step is to determine the topology of the BN. Since, *'SootPresence'*, *'COHb'*, and *'TonguePosition'* are evidence nodes; and *'VitalBurning'* is a query node, the direction of flow will be from *'VitalBurning'* to *'SootPresence'*, *'COHb'*, and *'TonguePosition'*. This is illustrated in Figure 10a. In machine learning terminology, this is referred to as a Naive Bayes graphical structure: *'VitalBurning'* is the class node and *'SootPresence'*, *'COHb'* and *'TonguePosition'* are the features. The absence of links between the features is an indication of conditional independence [26].



(a) Preliminary topology for BN.

SootPresence

VitalBurning	Positive	Negative
Positive	0.98485	0.75
Negative	0.01515	0.25

COHb

VitalBurning	Positive	Negative
>=10%	0.87879	0.011299
<10%	0.12121	0.988701

TonguePosition

VitalBurning	Positive	Negative
Protruded	0.78788	0
NotProtruded	0.21212	1

VitalBurning

Positive	0.05102
Negative	0.94898

(b) CPT for case study.

Figure 10: Preliminary BN with associated CPTs.

6.2.3 Parameterisation

After determining the topology of the BN, the next step is to construct the CPTs associated with each node. However, since the data contains a latent variable, a structural expectation-maximization (EM) algorithm is implemented to process the values during network learning. This is executed in BayesiaLab[4] to calculate the probabilities associated with each node. The EM algorithm is applied after each new arc is added, suppressed or inversed. The observations are then used to populate the CPT via the method of Maximum Likelihood Estimation [9]. The probabilities obtained is then transferred with the BN structure in Figure 10a to Hugin Lite. The CPTs obtained are illustrated in Figure 10b.

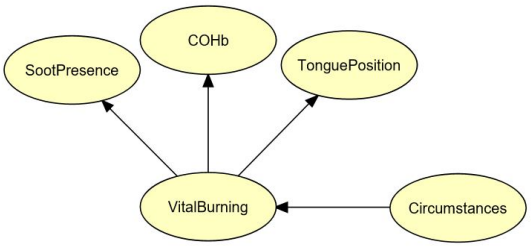
Now suppose another node is added to the BN, i.e., the circumstances surrounding the death of the victim, denoted by *'Circumstances'*. The preliminary node choices in Table 2 is now updated to include *'Circumstances'*. The *'Circumstance'* node has the following states:

$$\text{Circumstances} = \begin{cases} \text{AircraftAccident} & \text{for cause of death: Aircraft accident} \\ \text{DestroyingEvidence} & \text{for homicide cases where the corpses are burned to destroy evidence} \\ \text{Necklacing} & \text{for cause of death: Necklacing} \\ \text{ShackFire} & \text{for cause of death: Shack fire} \\ \text{Other} & \text{otherwise} \end{cases}$$

The proposition for '*Circumstances*' constitute a partition of the general class of all the possible

[4]BayesiaLab details: Copyright © 2001-2017 Bayesia S.A.S. All rights reserved. This software is protected by the international laws related to copyright

death causes and $Pr(TongueProtrusion|Circumstances_i) \neq Pr(TongueProtrusion|Circumstances_j)$ for $i \neq j$. The topology of the BN and the associated CPTs related to the updated BN is shown in Figure 11a and Figure 11b respectively. Note that the *'VitalBurning'* node CPT is updated to include the effect of the *'Circumstances'* node. The probabilities in Figure 11b is obtained during a consultation with Prof Herman Bernitz, who is an expert in the field of forensic odontology. Therefore, the BN now consists of manifest variables and latent variables generated from expert knowledge and machine learning techniques. Figure 12 shows the monitor windows associated with the BN before any belief updating due to new evidence in the network.



(a) Updated BN containing *'Circumstances'* node.

VitalBurning

Circumstances	DestroyingE	ShackFire	Other	Necklacing	AircraftAcci
Positive	0.05102	0.892245	0.5	0.813953	0.5
Negative	0.94898	0.107755	0.5	0.186047	0.5

Circumstances

DestroyingEvide	0.131135
ShackFire	0.437117
Other	0.052409
Necklacing	0.262045
AircraftAccident	0.117294

(b) CPT for node *'Circumstances'* and *'VitalBurning'*.

Figure 11: BN for vital burn victim.

6.2.4 Inference

The next step is testing the BN by computing a 'what-if' analysis scenario. New evidence is propagated through the network. For illustrative purposes, the following scenarios will be considered:

Scenario A

Consider a crime scene investigation where a severely charred body is found in an open field in a rural area near Polokwane, Limpopo. Further investigation yields that exposure to death occurred before death. The

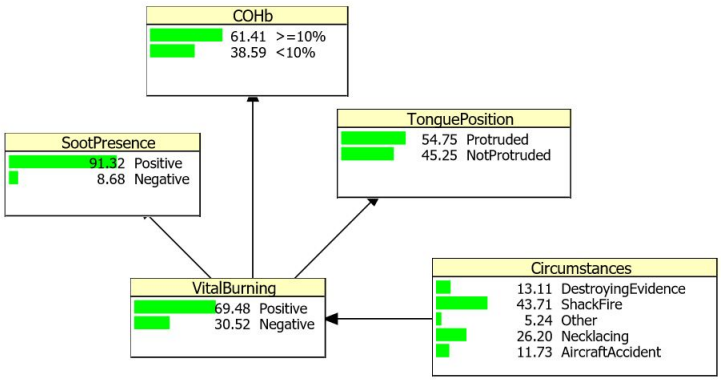
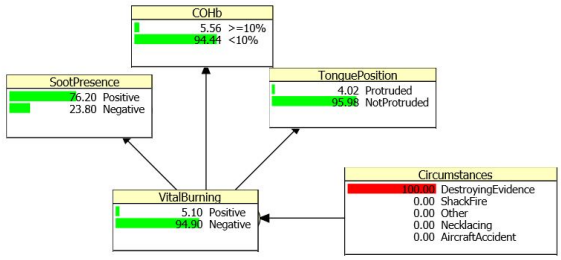
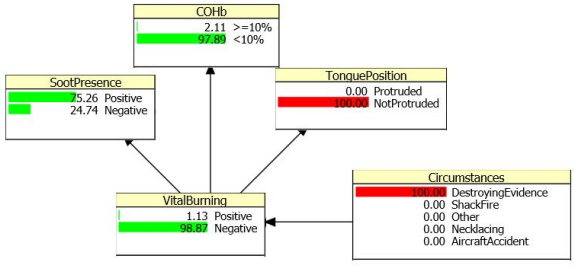


Figure 12: Marginal probabilities of variables.

cause of death is ruled to be a homicide case where the corpse is burned to destroy evidence. The what-if analysis is illustrated in Figure 13. Notice how the BN updates the information in the *'VitalBurning'* node. Initially, the '*VitalBurning*' node returned a probability of 0.6948 for a positive result and a probability of 0.3052 for a negative result. As soon as evidence becomes available on the circumstances surrounding the victim's death, the '*VitalBurning*' node updates its probabilities, the probability of the victim sustaining vital burns decreases to 0.0510. However, the probability of the victim not obtaining vital burning increases to 0.9490 (illustrated in Figure 13a). Indicating that vital burning for homicide cases where the corpses are burned to destroy evidence is less likely to happen. Suppose the forensic pathologist determines that the victim does not have a protruded tongue. The effect on '*VitalBurning*' is illustrated in Figure 13b. Notice how the probability of a positive vital burn result decreases to 0.0113 as the new evidence is added to the BN. This is a possible indication that a non-protruding tongue does not indicate vital burning. However, further investigation is needed to make a conclusion.



(a) Belief updating when *'Circumstances'* node receives evidence.

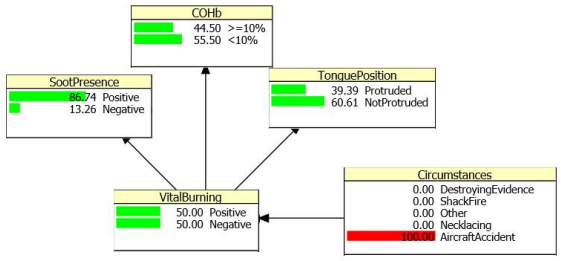


(b) Belief updating when *'TonguePosition'* receives evidence.

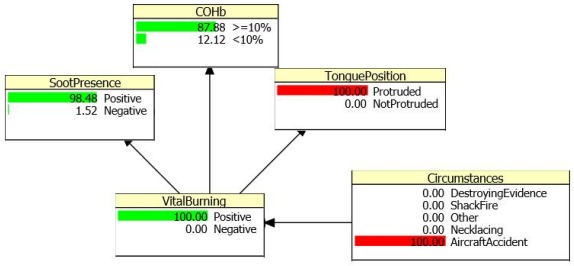
Figure 13: What-if analysis for scenario A.

Scenario B

Consider now a crime scene investigation where a severely charred body is found at an aircraft accident scene on a private farm in Alldays, Limpopo. The 'what-if' analysis is demonstrated in Figure 14. Adding evidence from the circumstances surrounding the victim's death, the probability of obtaining vital burns is decreased to 0.5 (from the initial 0.6948) as illustrated in Figure 14a. Suppose the forensic pathologist reports that the victim's tongue is protruded. From expert knowledge, if an aircraft accident victim has a protruded tongue, the victim was in the backseat of the aircraft. Since the pilot would die of initial impact, his tongue would be reported as not protruded. An interesting result is obtained from belief updating. Once the evidence is added that the victim's tongue is protruded, the '*VitalBurning*' node's probability increases to 1. Indicating that tongue protrusion is a possible indicator of vital burning.



(a) Belief updating when '*Circumstances*' node receives evidence.



(b) Belief updating when '*TonguePosition*' receives evidence.

Figure 14: What-if analysis for scenario B.

BNs for pedagogical purposes

Consider a scenario where the forensic investigator knows that vital burning is present. Note that this is a hypothetical scenario for pedagogical purposes. The belief updating is illustrated in Figure 15. Notice how the variable '*TonguePosition*' updates. The initial probability for '*TonguePosition*' Protruded is 0.5475. As soon as the network updates its beliefs, the probability of witnessing a protruded tongue increases to 0.7879. This indicates that there is a statistical relationship between vital burning and tongue protrusion.

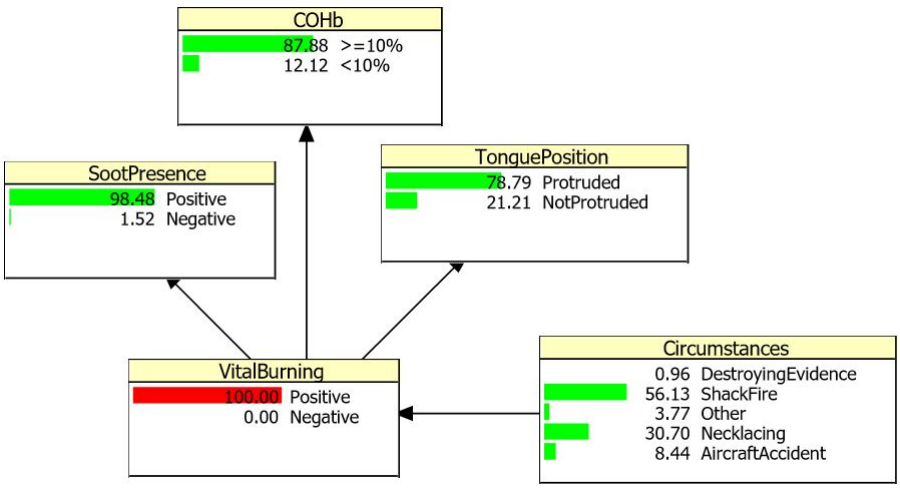


Figure 15: Belief updating for hypothetical scenario.

A 'what-if' analysis is not conducted on the effect of '*SootPresence*' and '*COHb*' on '*VitalBurning*' as both variables are possible indicators of heat exposure [6, 14].

7 Conclusion

In this report, the forensic assessment of burnt bodies is considered, whereas Taroni *et* al. considered the evaluation of evidence in the following three forensic disciplines: Transfer evidence, marks and DNA

evidence [32]. The research problem: "Is tongue protrusion an indicator of vital burning?" is first evaluated using non-parametric statistical analysis as a comparative study. In the comparative study, the association between soot deposits in the respiratory tract and tongue protrusion, and the association between the percentage of carboxyhaemoglobin in the blood and tongue protrusion is tested. The comparative study concluded that there is a statistically significant association between soot deposits in the respiratory tract and tongue protrusion. However, the comparative study concluded that there is no statistically significant association between the percentage of carboxyhaemoglobin in the blood and tongue protrusion. This result leads to the decision to evaluate the research problem using Bayesian networks. Not only does Bayesian networks provide the user to include latent variables and expert knowledge, it is capable to predict posterior probabilities from prior probabilities.

Through a 'what-if' analysis, the Bayesian network showed that once evidence is added to the network, the belief on the '*VitalBurning*' node updated to a probability of 1. Indicating that there is a statistically significant relationship between tongue protrusion and vital burning. Therefore, it can be concluded that tongue protrusion is a possible indicator of vital burning. Thus, tongue protrusion may be considered with other variables, such as soot deposits and COHb% in blood, when determining the possibility of vital burning.

Challenges to the case study include the elimination of missing values that resulted in a smaller sample size, as well as ethical limitations[5]. In future investigations, the BN can be extended to include more circumstantial variables which can facilitate autopsy reports. Aside from this, sampling for more data should be done in the future to confirm the results obtained. With a larger sample size, it will be possible to test whether a non-protruding tongue could indicate non-vital burning. Bernitz *et* al. state that the exact mechanism involved in tongue protrusion during vital burning should also be investigated [4].

[5]Inquest Act no. 58 of 1959 of the Republic of South Africa

References

[1] Colin GG Aitken and Franco Taroni. *Statistics and the Evaluation of Evidence for Forensic Scientists*, volume 16. Wiley Online Library, 2004.

[2] David Anderson, Dennis Sweeney, and Thomas Williams. *Modern Business Statistics with Microsoft Excel*. Nelson Education, 2014.

[3] Thomas Bayes, Richard Price, and John Canton. *An essay towards solving a problem in the doctrine of chances*. C. Davis, Printer to the Royal Society of London, 1763.

[4] Herman Bernitz, Paul J van Staden, Christine M Cronjé, and René Sutherland. Tongue protrusion as an indicator of vital burning. *International Journal of Legal Medicine*, 128(2):309–312, 2014.

[5] Michael Bohnert. Protrusion of the tongue in burned bodies as a vital sign? *International Journal of Legal Medicine*, 128(2):317–317, Mar 2014.

[6] Michael Bohnert, Christoph R Werner, and Stefan Pollak. Problems associated with the diagnosis of vitality in burned bodies. *Forensic Science International*, 135(3):197–205, 2003.

[7] Leonid Chindelevitch, Po-Ru Loh, Ahmed Enayetallah, Bonnie Berger, and Daniel Ziemek. Assessing statistical significance in causal graphs. *BMC Bioinformatics*, 13(1):35, 2012.

[8] Jacob Cohen, Patricia Cohen, Stephen G West, and Leona S Aiken. *Applied Multiple Regression/-Correlation Analysis for the Behavioral Sciences*. Routledge, 2013.

[9] Stefan Conrady and Lionel Jouffe. *Bayesian Networks and BayesiaLab: A Practical Introduction for Researchers*. 2015.

[10] Roger Cook, Ian W Evett, Graham Jackson, PJ Jones, and JA Lambert. A model for case assessment and interpretation. *Science & Justice*, 38(3):151–156, 1998.

[11] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–242, 1958.

[12] Philip A Dawid and Ian W Evett. Using a graphical method to assist the evaluation of complicated patterns of evidence. *Journal of Forensic Science*, 42(2):226–231, 1997.

[13] Alta de Waal, Hildegarde Koen, Pieter de Villiers, Henk Roodt, Nyalleng Moorosi, and Gregor Pavlin. Construction and evaluation of Bayesian networks with expert-defined latent variables. In *Information Fusion (FUSION), 2016 19th International Conference on*, pages 774–781. IEEE, 2016.

[14] Laurent Fanton, K Jdeed, S Tilhet-Coartet, and D Malicier. Criminal burning. *Forensic Science International*, 158(2):87–93, 2006.

[15] Paolo Garbolino and Franco Taroni. Evaluation of scientific evidence using Bayesian networks. *Forensic Science International*, 125(2):149–155, 2002.

[16] Alan Hájek. Interpretations of probability. In *The Stanford Encyclopedia of Philosophy (Zalta.* Citeseer, 2003.

[17] Graham Jackson. The scientist and the scales of justice. *Science & Justice*, 40(2):81–85, 2000.

[18] Edwin T Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[19] Kevin B Korb and Ann E Nicholson. *Bayesian Artificial Intelligence*. CRC Press, 2010.

[20] David A Lagnado, Norman Fenton, and Martin Neil. Legal idioms: A framework for evidential reasoning. *Argument & Computation*, 4(1):46–63, 2013.

[21] Dennis V Lindley. *Understanding Uncertainty*. John Wiley & Sons, 2006.

[22] Frank J Massey Jr. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.

[23] Cyrus R Mehta and Nitin R Patel. IBM SPSS exact tests. *IBM Corporation, Cambridge, MA*, 2011.

[24] Scott Menard. *Applied Logistic Regression Analysis*, volume 106. Sage, 2002.

[25] Scott Menard. *Logistic Regression: From Introductory to Advanced Concepts and Applications*. Sage, 2010.

[26] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[27] Thomas D Nielsen and Finn V Jensen. *Bayesian Networks and Decision Graphs*. Springer Science & Business Media, 2009.

[28] Judea Pearl. Causality: Models, reasoning and inference. *Econometric Theory*, 19(675-685):3,46, 2003.

[29] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 2014.

[30] Rosie Shier. Statistics: 2.3 the Mann-Whitney U Test. *Mathematics Learning Support Centre*, 15:2013, 2004.

[31] Sidney Siegel and John W Tukey. A nonparametric sum of ranks procedure for relative spread in unpaired samples. *Journal of the American Statistical Association*, 55(291):429–445, 1960.

[32] Franco Taroni, Colin GG Aitken, Paolo Garbolino, and Alex Biedermann. *Bayesian Networks and Probabilistic Inference in Forensic Science*. Wiley Online Library, 2006.

[33] Franco Taroni, Alex Biedermann, Silvia Bozza, Paolo Garbolino, and Colin Aitken. *Bayesian Networks for Probabilistic Inference and Decision Analysis in Forensic Science*. John Wiley & Sons, 2014.

[34] Jake VanderPlas. Frequentism and Bayesianism: A Python-driven primer. *arXiv preprint arXiv:1411.5018*, 2014.

[35] John Venn. *The Logic of Chance*. Courier Corporation, 2006.

[36] Sewall Wright. Correlation and causation. *Journal of Agricultural Research*, 20(7):557–585, 1921.

Appendix

Logistic regression

Since logistic regression is not the focus of this paper and is only used as a comparative study, this section will only cover essential theory related to Logistic regression. Logistic regression is a statistical method for analyzing a regression model that consist of dependent categorical variables. The theory of logistic regression was developed by David Cox during 1958 [11]. In logistic regression, the response variable may be dichotomous, nominal or ordinal, and the explanatory variable may come across as ratio, interval or dummy variables [25]. The logistic regression model is used to estimate the probability of a response based on one or more explanatory variables. Therefore, the presence of a specific risk factor is used to estimate the percentage increase in probability of a given outcome.

The Wald Chi-square statistic can be used to assess the significance of the coefficients contribution within a regression model [24]. The Wald Chi-square statistic can be written as the ratio of the squared estimate and the squared standard error:

$$W_i = \frac{\beta_i^2}{SE_{\beta_i}^2}$$

Note the following limitations of the Wald Chi-square statistic: when β_i is significantly large, the standard error is also large. Therefore, increasing the probability of obtaining a Type-II error. Another limitation is that when data is sparse, the Wald Chi-square statistic becomes biased [8]. The Wald confidence limits is the confidence interval for the proportional odds ratio given the other explanatory variables in the model. The confidence interval is equal to the Wald Chi-square statistic; if it excludes 1, the null hypothesis for the regression model would be rejected.

Non-parametric tests

Since non-parametric tests is not the main focus of this paper, this section will briefly cover essential theory linked to the comparative study.

Fisher's exact test

Sir Ronald A. Fisher developed the Fisher's exact test for a 2×2 contingency table [23]. Fisher's exact test is used when two variables are to be compared to establish an association between these variables. For the purposes of this paper, only the interpretation of Fisher's exact test is of importance. Since Fisher's exact test returns a $p-value$, hypothesis testing will follow the usual rejection criteria. For large samples, a Chi-square test can be done. However, this will only be an approximation since the sampling

distribution of the test statistic is is calculated in such a way as to approximate it to the Chi-square distribution.

Kolmogorov-Smirnov

The Kolmogorov-Smirnov test quantifies the distance between the cumulative distribution function and the empirical distribution function [22]. Therefore, it is used to test whether there is a significant difference in the distribution of the samples.

Mann-Whitney test

The Mann-Whitney test is a non-parametric test to test under the null hypothesis if two samples come from the same population, i.e., test to see if the two samples have the same median. Therefore, the Mann-Whitney test is used as a test for location [30].

Siegel-Tukey test

The Siegel-Tukey test, tests under the null hypothesis if there is a difference in spread between two samples. Therefore, the Siegel-Tukey test is a test for variability [31].

SAS Code

```
*************************************************************************
Iena Petronella Derks   13075782
Bayesian networks applied to forensic science
Year: 2017
*************************************************************************
*************************************************************************
Tongue data in Binary format
        Focus point: COHb10, Tongedit and sootedit.
AllTongueData: Used to eliminate data not used. Therefore, data such as
        when the tongue is destroyed in such a manner that it is
        impossible to determine tongue protrusion, as well as when no
        data regarding the position of the tongue was recored is
        eliminated from the dataset. This dataset contains 107
        observations.
MissingTongueData: Used to eliminate the missing values present in the
        data. This dataset contains 74 observations, meaning that 33
```

```
        observations returned missing values.
BinaryTongue: Used to convert the newly "cleaned" data to binary
        code. This dataset contains 74 values.
Binary used as follow:
TongueProtrusion        0: Tongue not Protruded 1: Tongue Protruded
SootLevel       0: Negative     1: Positive
COHbScore       0: <10%         1: >=10%
*************************************************************************;
Data AllTongueData;
        infile 'C:\Users\Ineke Derks\Desktop\UP 2017\Research Project\
                        Tongue protrusion\tongue-vs-soot.txt';
        input Obsnumber Race Gender Age TonguePosition SootLevel COHb
                        COHbScore @@;
        TongueProtrusion = TonguePosition;
        if TonguePosition = 1 then TongueProtrusion = .;
        if TonguePosition = 4 then do;
                if SootLevel = 1 then TongueProtrusion = .;
                if SootLevel = 2 then TongueProtrusion = 2;
        end;
run;
Data MissingTongueEliminate;
        set AllTongueData;
        keep TongueProtrusion SootLevel COHbScore;
                if TongueProtrusion = . then delete;
                if SootLevel = . then delete;
                if COHbScore = . then delete;
run;
Data BinaryTongue;
        set MissingTongueEliminate;
        keep TongueProtrusion SootLevel COHbScore;
                if TongueProtrusion = 2 then TongueProtrusion = 0;
                        else TongueProtrusion = 1;
                if SootLevel = 2 then SootLevel = 0;
                        else Sootlevel = 1;
```

```
                if COHbScore = 1 then COHbScore = 0;
                        else COHbScore = 1;
run;
*************************************************************************
Nonparametric regression -> All the data including missing values
        Therefore, the dataset "AllTongueData" is used to perform the
        following tests:
        -> TongueProtrusion vs SootLevels
                -> TongueProtrusion associated with SootLevels: Fisher's
                        exact test via PROC LOGISTIC.
        -> TongueProtrusion vs percentage COHb
                -> Difference in median COHb% levels between tongue
                        protrusion and tongue not protruded: Wilcoxon
                        scores via PROC NPAR1WAY.
                -> Difference in spread COHb% levels between tongue
                        protrusion and tongue not protruded: Siegel-Tukey
                        scores via PROC NPAR1WAY.
                -> Difference in distribution COHb% levels between tongue
                        protrusion and tongue not protruded:
                        Kolmogorov-Smirnov test via PROC NPAR1WAY.
                -> TongueProtrusion associated with percentage COHb:
                        Fisher's exact test via PROC LOGISTIC.
*************************************************************************;
goptions reset = all;
title1 'Descriptive statistics';
proc freq data = AllTongueData;
        tables Race Gender TongueProtrusion SootLevel COHbScore;
run;
goptions reset = all;
title1 'Tongue protrusion vs presence of soot';
proc freq data = AllTongueData;
        tables TongueProtrusion * SootLevel / chisq expected cellchi2;
        tables TongueProtrusion *COHbScore / chisq expected cellchi2;
        exact lrchi;
```

```
run;
goptions reset = all;
ods graphics on; title1 'Tongue protrusion vs presence of COHb%';
proc npar1way data = AllTongueData wilcoxon st conover edf median
        correct = no plots = anovaboxplot;
        class TongueProtrusion;
        var COHb;       exact wilcoxon;
run;
goptions reset = all;
ods graphics on; title1 'Tongue protrusion vs presence of soot';
proc logistic data = AllTongueData;
        class SootLevel / descending;
        model TongueProtrusion(event = 'Tongue protruded') = SootLevel;
run;
goptions reset = all;
title1 'Tongue protrusion vs presence of COHb%';
proc logistic data = AllTongueData;
        class COHbScore / descending;
        model TongueProtrusion(event = 'Tongue protruded') = COHbScore;
run;
ods graphics off;
quit;
```

On parameter estimation in financial time series

Vincent Dixie 14010888

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr I.J.H. Visagie

Department of Statistics, University of Pretoria

30 October 2017

Abstract

Financial markets are extremely volatile and being able to model such markets accurately would increase our understanding of them. In this study different models will be fitted to observed financial data using the method of maximum likelihood. This is done in an attempt to identify the model, and its corresponding parameters, that best fits the time series observed data. The financial data used will be the historical S&P500 stock price.

Declaration

I, Vincent Eugene Dixie, declare that this essay, submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics, at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Vincent Eugene Dixie

_ _

Dr I.J.H Visagie

_ _

Date

Acknowledgements

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

Contents

List of Figures

List of Tables

1 Introduction

This study is concerned with fitting various models to observed time series data. Each model is fitted to the historical prices of the Standard & Poor 500 (S&P500) index. The S&P500 is a capitalization weighted index which is comprised of 500 companies.

There are many methods that one can use to estimate parameters, some of the most popular being maximum likelihood estimation (MLE), the method of moments estimation, and least-square estimation. MLE is used in order to obtain parameter estimates throughout this study. When using MLE, the parameter set that maximizes the likelihood of observing the given data is chosen as the parameter estimates. We are using MLE due to the desirable characteristics of the estimates, such as the consistency and asymptotic normality of the estimates when models are fitted to independent and identically distributed data. The vast amount of literature available on MLE adds to the desirability of this estimation method. Furthermore, many software packages include algorithms for maximum likelihood estimation.

Financial models are often of a geometric form, i.e., the stock price at time t is often modelled as:

$$S_t = S_0 exp(X_t)$$

where S_0 is the starting price and X_t is some stochastic process. X_t is known as the log-return process. In this study we model the log-return process using various models.

The following models will be fitted to observed financial time series data using MLE.

- The Black-Scholes model. This model has had a major influence in the way traders hedge and price options and has two parameters that need to be estimated. Under this model X_t follows a Brownian motion.
- The geometric normal inverse Gaussian (N$\circ$IG) model. This model has four parameters that interact with each other in such a way that it can model a myriad of distributions and is a popular model for log-returns, see [3].
- The geometric lognormal-normal model. This is a time changed model; the evolution of time is not assumed to be constant. Rather time itself is modelled as evolving according to a lognormal process. This is explained in more detail in section 4.2. This model is often used to accurately model data with heavy tails. The lognormal-normal model has 4 parameters that require estimation.
- The ARCH (autoregressive conditional heteroscedasticity) model. This model is used to describe time-varying volatility in financial markets.
- The GARCH (generalized autoregressive conditional heteroscedasticity) model. This model, which is used to describe volatility in financial markets, is a generalization of the ARCH model.

2 Empirical properties of log-returns

Cont [2] presents a set of stylized empirical facts based on the statistical analysis of price variations in different types of financial markets. Stylized empirical facts refer to findings in empirical data that are so consistent that it is accepted as fact. The author discusses some general issues relating to the statistical studies of financial time series and then describes the various statistical properties of asset returns.
The mentioned properties include an absence of significant autocorrelations of asset returns, except in small intraday time periods of approximately 20 minutes where market microstructure effects are taken into consideration. Market microstructure refers to the processes observed only when considering very small time scales. Gain/loss asymmetry, where large downward movements in stock prices are observed but not equally large upward movements, i.e., frequent smaller upward movements are observed along with the occasional large downward movement. Aggregational Gaussianity is also observed in financial returns. This means that the distribution of returns appears to be more normally distributed at greater time scales. Therefore the shape of the distribution is dependent on the time scale used. Volatility clustering, where high-volatility events tend to cluster in time, this means that a small(large) price change is typically proceeded by another small(large) change in price. The returns distribution exhibit heavy tails even after correcting for volatility clustering by using GARCH models. The absence of autocorrelations in the asset returns lends support for random walk models of prices where the returns are considered to be independent random variables. However, the lack of serial correlation does not mean that the increments of the asset returns are independent. It can be shown that some nonlinear functions of asset returns, such as the squared returns and absolute returns, actually exhibit significant positive autocorrelation which implies that there is nonlinear dependence. This serves as a motivation for considering ARCH and GARCH models.
In [2], the author emphasizes statistical properties common to many of the popular markets and instruments. The author then shows how many of the popular statistical approaches used to study financial data sets are invalidated by the mentioned statistical properties. An example of this is how the absence of autocorrelations makes creating a simple statistical trading strategy impossible. If price changes did exhibit significant correlations, this would result in a strategy with positive expected profit which would result in statistical arbitrage.

2.1 Describing the log-return data

The programming language R, see [6], is used throughout this study in order to estimate the parameters of the models. We input the historical stock prices of the S&P500 into R and calculated the log returns.

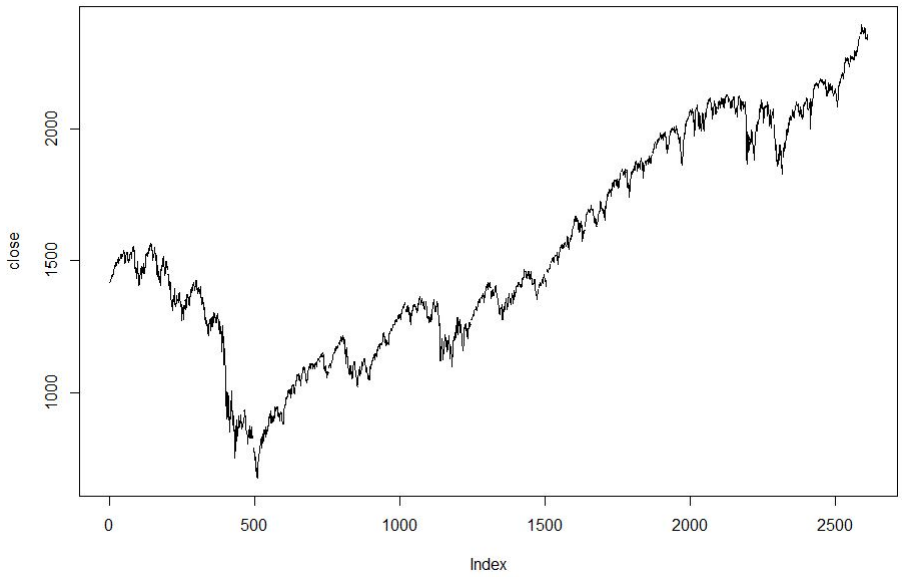


Figure 1: Plot of the S&P500 prices

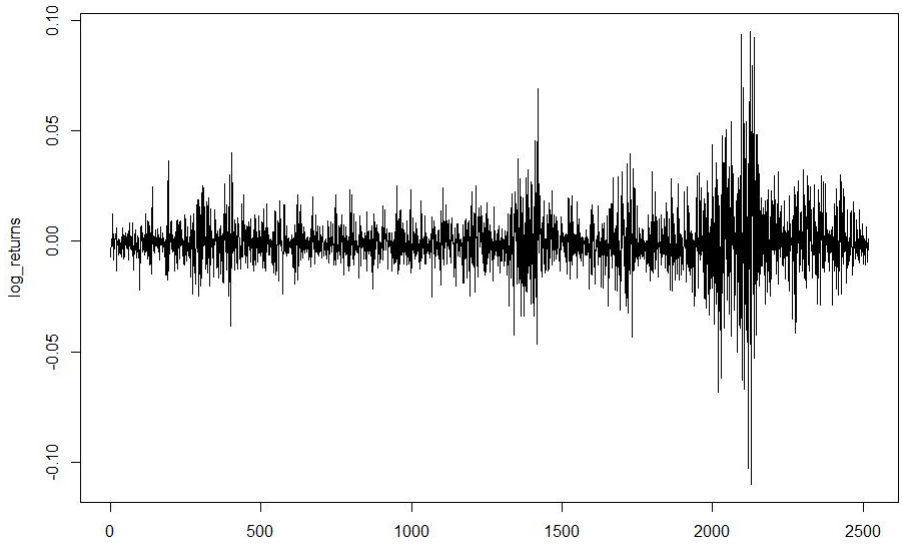


Figure 2: Plot of the S&P500 log-returns

The empirical kernel density of the log-return data is given in Figure 3.

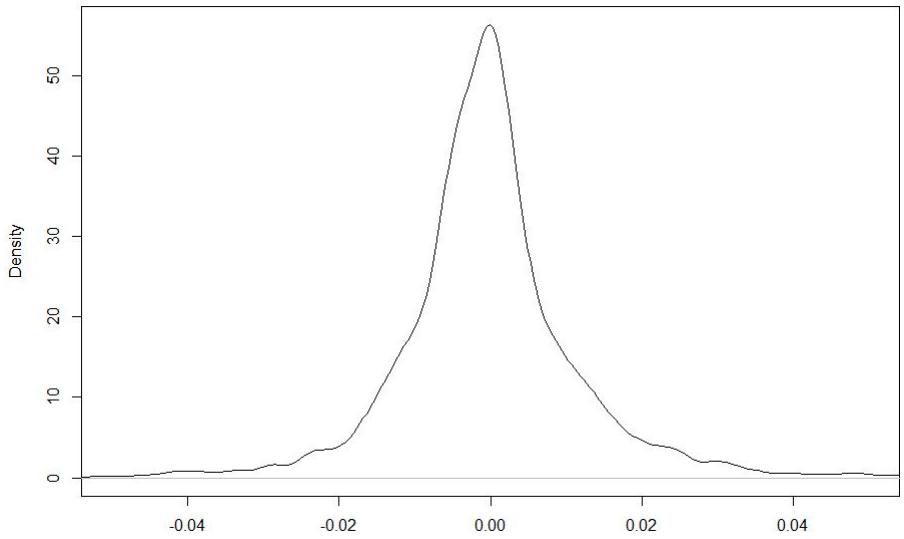


Figure 3: Empirical kernel density plot of the log-returns

The following descriptive statistics were taken from the data:

- Mean = -0.0002023.
- Variance = 0.0001732.
- Skewness = 0.3243.
- Kurtosis = 10.0278.

We use the S&P500 historical stock price data because it is a closely monitored stock index that is also highly popular with both market traders and statisticians alike due to it being comprised of 500 companies which issue 505 common stocks that cover approximately 80 percent of the American equity market. The S&P500 is traded on American stock exchanges which make it a highly traded stock. The data set used has 2519 observations of historical S&P500 daily closing prices which start from 28 March 2007 to 28 March 2017.
In what follows we fit various models to the observed log-returns. Section 3 discusses the Black-Scholes model while Section 4 is concerned with model advanced financial models.

3 The Black-Scholes model

The Black-Scholes model was developed in 1973 by Fischer Black, Myron Scholes and Robert Merton. Under the Black-Scholes model, stock prices are assumed to follow a geometric Brownian motion. As a result, the marginal distribution on the log-returns is normal under this model, see [8]. In a Black-Scholes market, the stock price at time t is given by:

$$S_t = S_0 \exp(\mu t + \sigma W_t),$$

where S_0 is the stock price at time 0, μ is the drift parameter, σ is the volatility of the stock and W_t denotes a standard Brownian motion at time t.

The book "Financial Modelling with Jump Processes" [9] provides a detailed discussion of the Black-Scholes model is provided in [9]. Thereafter the authors highlight issues where the Black-Scholes model fails to model real prices over various time-scales of interest. Many of the properties and assumptions used in the Black-Scholes model conflict with the stylized empirical facts regarding financial data sets discussed in the previous section. Examples of these include the continuity of the stock price; it is well known that stock prices contain jump discontinuities, see [7].

The only advantage of this model over the others discussed in this study is its simplicity, see [8]. Based on stylized facts discussed above, we note that the assumptions of the Black-Scholes models do not hold in practice due to the following observed facts:

- Log-returns do not follow the normal distribution.
- The volatilities are clustered, meaning that large movements tend to be followed by large movements and small movements tend to be followed by small movements.
- The volatilities change stochastically over time.

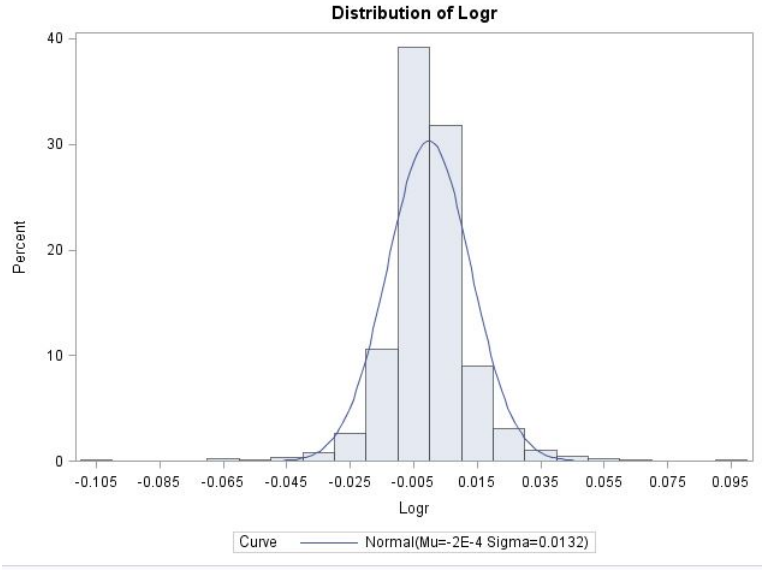


Figure 4: SAS plot of the log-returns

Tests for Normality				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.111795	Pr > D	<0.0100
Cramer-von Mises	W-Sq	11.72681	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	63.9827	Pr > A-Sq	<0.0050

Table 1: SAS test for normality

Based on the SAS PROC UNIVARIATE test for normality in table 1, the observed log-returns are not normally distributed and therefore we can estimate that the Black-Scholes model will not fit the estimated empirical density well.

4 Advanced financial models

Below we consider various types of models. We discuss the estimation of the parameters of these models in each case. Each of the models that follow are generalizations of the Black-Scholes model. Where the Black-Scholes is modeled by $S_t = S_0 exp(X_t)$ where $X_t \sim N(\mu_t, \sigma_t^2)$, the following models assume more flexible processes for X_t.

4.1 The geometric normal inverse Gaussian model

Under this model the log-returns of a stock price process are assumed to follow a normal inverse Gaussian (N◦IG) distribution. The N◦IG distribution is a flexible distribution and is a popular choice in the modeling of heavy-tailed processes. The N◦IG distribution was first developed by Barndorff-Nielsen and is fully described by four real valued parameters. Different parameter values allow for a wide range of distributional shapes. The N◦IG distribution is desirable because it is able to mimic the heavy tailed nature of the observed return data as well as the observed skewness, see [4] and [11].
The N◦IG density function is given by:

$$f(x, \alpha, \beta, \mu, \delta) = \frac{\alpha\delta}{\pi} \exp(\delta\sqrt{\alpha^2 - \beta^2} + \beta(x - \mu))\phi(x)^{-\frac{1}{2}} K_1(\alpha\phi(x)^{\frac{1}{2}}),$$

where $\phi(x) = \delta^2 + (x - \mu)^2$, $K_1(x)$ denotes the modified Bessel function of the third kind of order 1 with $\alpha > 0$, $|\beta| < \alpha$ and $\delta > 0$, see [4].
Under the geometric N◦IG model, the stock price is modelled as:

$$S_t = S_0 exp(X_t),$$

where X_t follows a N∘IG process.

The N∘IG process is defined in terms of the increments. Let each daily log-return follow a N∘IG distribution, independent of the other log-returns.

$$Y_j \sim N \circ IG(\alpha, \beta, \mu, \delta),$$

then the log-return process is defined as

$$X_t = \sum_{j=1}^{t} Y_j.$$

The marginal distribution of the log-return process is

$$X_t \sim N \circ IG(\alpha, \beta, \mu t, \delta t).$$

Consider the moments of the N∘IG distribution. Using the notation $\gamma = \sqrt{\alpha^2 - \beta^2}$, the first four moments of the $N \circ IG(\alpha, \beta, \mu t, \delta t)$ distribution are as follows:

- Mean: $\mu + \delta\beta/\gamma$
- Variance: $\delta\alpha^2/\gamma^3$
- Skewness: $3\beta/\left(\alpha\sqrt{\delta\gamma}\right)$
- Kurtosis: $3(1 + 4\beta^2/\alpha^2)/(\delta\gamma)$

The N∘IG distribution is more flexible than the normal distribution and therefore, we expect this model to fit the observed log-returns better than the case with the normal distribution. The shortfall of this model is the complications that arise in estimating the parameters due to the Bessel functions that make calculating the partial derivatives of the density function a complicated procedure.

4.2 The geometric lognormal-normal process model

The lognormal-normal model is obtained by making time evolve at a stochastic rate. I.e., time is modelled by a lognormal process. This model has an interesting trait which allows for the variance of the distribution to remain constant while the kurtosis is changed as desired, see [1]. Consider a process $X(t)$, where the difference between $X(t)$ and $X(t-1)$ (denoted by $\triangle X(t)$) are independent and normally distributed and directed by a process $T(t)$ where the difference between $T(t)$ and $T(t-1)$ (denoted by $\triangle T(t)$) are

also independent but lognormally distributed, see [1]. If $\triangle X(t)$ is normally distributed with a mean of 0 and a variance of σ^2, then $\triangle X(T(t))$ is a lognonrmal-normal distribution. The lognormal-normal model has 4 parameters that require estimation and the density of the distribution is given by:

$$f(x, \alpha, \beta, \mu, \sigma) = \frac{1}{2\pi\sigma\beta} \int_0^\infty y^{-\frac{3}{2}} exp(-\frac{(x-\mu y)^2}{2\sigma^2 y} - \frac{(log(y)-\alpha)^2}{2\beta^2})dy,$$

where $\beta > 0$, $\sigma > 0$, $\mu \in \mathbb{R}$, $\alpha \in \mathbb{R}$.

This model is desirable because of its characteristic that allows for the kurtosis, which controls the intensity of the peakedness of the distribution, to be changed as needed while maintaining a constant variance.

In Figure 4, we see that a normal approximation to the data does not fit well and that a model with larger kurtosis would be able to model the underlying process more accurately. We show that the lognormal-normal distribution, being able to change its shape, will be a more suitable choice for modelling log-return data.

4.3 Time-varying volatility models

Below we consider the ARCH(1) and GARCH(1,1) model.

4.3.1 The ARCH model

The ARCH (autoregressive conditional heteroscedasticity) model is used to model time-changing volatility. Such movements were not until recently considered important, see [5]. The ARCH(q) model is a discrete-time stochastic volatility model and is defined as:

$$Y_t = \mu + \varepsilon_t,$$

where Y_t is the log-return on day t, with

$$\varepsilon_t = z_t \sqrt{h_t},$$

where

$$h_t = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2,$$

μ, ω, α_i are constants for all i$\in 1, 2, ..., q$ with $\mu \in \mathbb{R}$, $\omega > 0$, *and* $\sum_{i=1}^{q} \alpha_i < 1$.

In this model μ represents the mean of the process, h_t represents the volatility of the process and is calculated as a function of the previous error terms added to a base volatility. ε_t represents the error term added to the mean return and is calculated as the square root of the volatility (h_t) multiplied by an innovation term (z_t). A standard normal distribution is the most popular choice for the innovation

term. ω is a constant term serving as a baseline for the volatility process. The mean of the process is often modelled using an autoregressive process, see [7] and [10].
In this study only the ARCH(1) model is considered.

4.3.2 The GARCH model

The GARCH (generalized autoregressive conditional heteroscedasticity) model, being autoregressive, uses historic variance to model the current variance of log-returns. This model often uses the normal distribution as a model for the innovation term. For an account of the use of these models in finance, see [5].
The GARCH(p, q) model is given by:

$$Y_t = \mu + \varepsilon_t,$$
$$\varepsilon_t = z_t \sqrt{h_t},$$

where

$$h_t = \omega + \sum_{j=1}^{p} \alpha_j h_{j-1}^2 + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i}^2,$$

μ, ω, α_i are constants for all i$\in 1, 2, ..., q$ and $j \in 1, 2, ..., p$ with $\mu \in \mathbb{R}$, $\omega > 0$, $\sum_{i=1}^{q} \alpha_i < 1 \, and \, \sum_{j=1}^{p} \beta_j < 1$.

In this model μ represents the mean of the process, h_t represents the volatility of the process and is calculated as a function of the previous error terms and previous volatility terms added to a base volatility. ε_t represents the error term added to the mean return and is calculated as the square root of the volatility (h_t) multiplied by an innovation term (z_t) which is assumed to be standard normal throughout this study. ω is a constant term serving as a baseline for the volatility process, see [7] and [10].
In this study only the GARCH(1,1) is considered.

5 Parameter estimation

Below we fit each of the models discussed in the previous section to the observed log-returns of the S&P500.

5.1 The Black-Scholes model

Method of estimation

Consider the definition of the Black-Scholes model:

$$S_t = S_0 exp(\mu t + \sigma W_t),$$

This model contains only two parameters that require estimation, namely μ and σ. Under this model:

$$\begin{aligned}\frac{S_{t+1}}{S_t} &= \frac{S_0 exp(\mu(t+1) + \sigma W_{t+1})}{S_0 exp(\mu t + \sigma W_t)} \\ &= exp(\mu + \sigma(W_{t+1} - W_t)).\end{aligned}$$

Since W_t denotes a standard Brownian motion, $(W_{t+1} - W_t) \sim$ Normal(0, 1). And therefore:

$$Log\left(\frac{S_{t+1}}{S_t}\right) = \mu + \sigma X,$$

where $X \sim N(0, 1)$.

This equation can also be written as

$$Log\left(\frac{S_{t+1}}{S_t}\right) = Y,$$

where $Y \sim N(\mu, \sigma^2)$.

It is well known that the maximum likelihood estimators for a normal distribution are the sample mean and sample variance of the data.

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i.$$
$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2.$$

Therefore, once the log returns of the data have been calculated, it is a trivial matter to calculate the sample mean and sample variance using any statistical package or software.

Estimation results

When using the maximum likelihood estimation we obtained the following estimates:

$$\hat{\mu} = -0.0002.$$

$$\hat{\sigma} = 0.0132.$$

As a means of visually evaluating the fit of the model to the data, we compare the density of the log-returns under the model (with the estimated parameters) to a kernel density estimate of the observed log-returns. We use the plot function in R to do this comparison [6].

The following is the graphical representation of the estimated kernel density estimate of the observed log-returns (black line) and the estimated normal distribution (blue line).

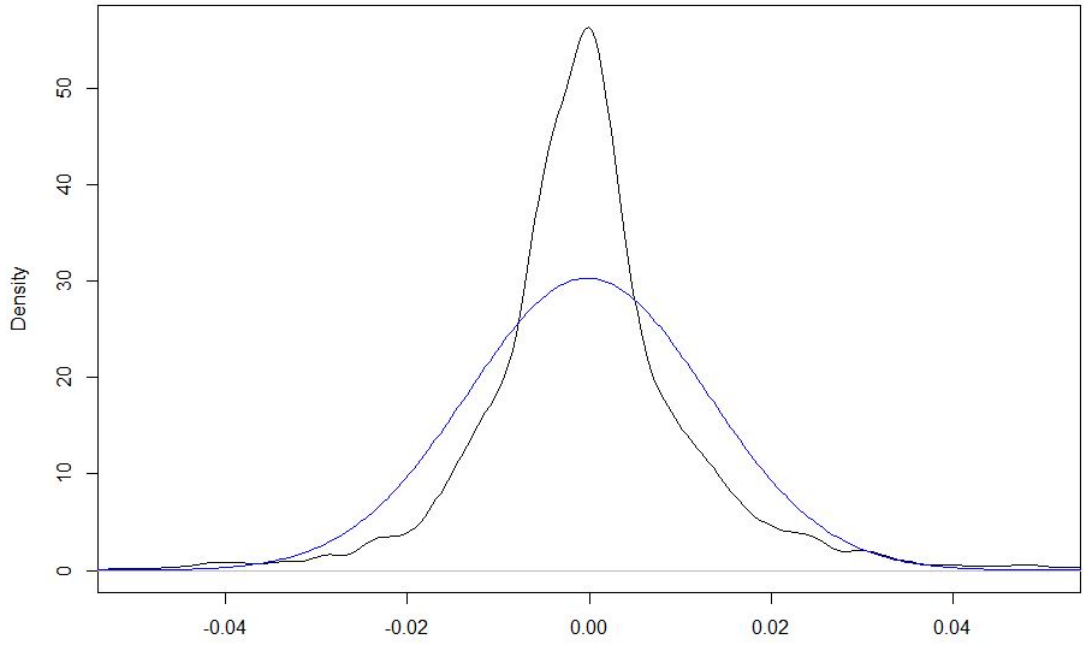


Figure 5: Black-Scholes model

Figure 6 provides some evidence against the assumption that log-returns follow a normal distribution.

5.2 The N∘IG model

The N∘IG model is a generalization of the Black-Scholes model. Under the Black-Scholes model: $S_t = S_0 exp(X_t)$, with $X_t \sim N(\mu t, \sigma^2 t)$. Under the N∘IG model, $X_t \sim N \circ IG(\alpha, \beta, \mu t, \delta t)$.

Method of estimation

In order to obtain starting values for the optimization algorithm, parameter sets are generated randomly and the log likelihood function of the data is calculated for each parameter set. We used two methods to

generate the starting parameter sets, the parameter set of each method which minimized the negative of the log likelihood was then optimized in order to obtain a local minimum at two different locations. The first method of generating parameters used nested loops to generate all possible combinations of whole numbers in a range. The second method generated one thousand random values for each of α, β, μand δ from a with specified ranges uniform distribution. The parameter set that maximizes the log likelihood is used as the starting values. The optimization is done as follows. A function was created in R that calculates the N$\circ$IG density at a data point with the parameters of the distribution given as parameters to the function. This essentially calculated the density at a data point for a specific set of parameters. Another function was then created that made use of the first function to calculate the density of the parameters at all the data points, and then multiplied the evaluated densities together and calculated the negative log of this value. We use the negative of the log likelihood since the optimization packages in base R optimize to a local minimum by default. The optimization function used is the optim function, see [6]. In short, these functions are used to output a single value as the negative log likelihood of a density with a defined set of parameters.

Estimation results

When using the maximum likelihood estimates from the method detailed above, we obtain the following estimates:

$$\hat{\alpha} = 40.8557.$$
$$\hat{\beta} = 5.8282.$$
$$\hat{\mu} = -0.0012.$$
$$\hat{\delta} = 0.007.$$

Figure 7 shows the kernel density estimate of the observed log-returns (black line). The estimated N$\circ$IG density is superimposed (in blue).

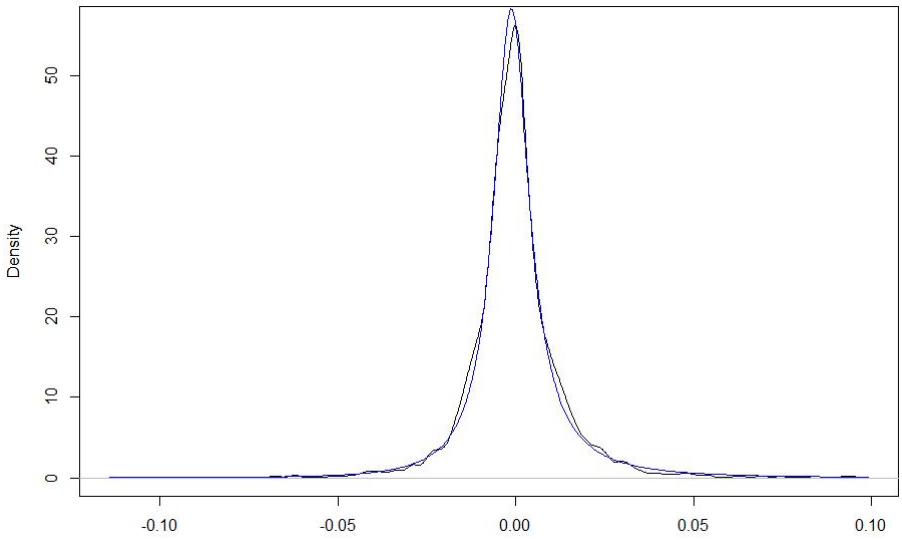


Figure 6: N*o*IG model

When the estimated distribution is graphically represented and overlayed with the empirical data, we can see how well the model fits. In section 6 we consider a quantitative measure of the closeness of the densities shown above.

5.3 The geometric lognormal-normal process model

The geometric lognormal-normal process model is also a generalization of the Black-Scholes model. Under the geometric lognormal-normal model, $X_t \sim LNN(\alpha, \beta, \mu t, \sigma t)$.

Method of estimation

The approach used to estimate the parameters of the geometric lognormal-normal distribution is similar to the approach used to fit the N*o*IG distribution. Possible starting values for the optimization algorithm are generated and the log likelihood of the data using the generated parameters is calculated in order find the parameter set that minimizes the negative log likelihood. We use the negative of the log likelihood since the optimization packages in base R optimize to a local minimum by default. The optimization function used is the optim function, see [6]. The method used to generated starting values involved generating one thousand random values for each of α, β, μ *and* δ from a with specified ranges uniform distribution. The parameter set that maximizes the log likelihood is used as the starting values. A function was created in R that calculates the geometric lognormal-normal density at a data point with the parameters of the distribution given as parameters to the function. This essentially calculated the density at a data point for a specific set of parameters. Another function was then created, that made use of the first function which calculated the density of the parameters at all the data points, and this function then multiplied the densities together and calculated the negative log of this value.

Estimation results

The parameter estimates returned by the program are:

$$\hat{\alpha} = 0.743.$$

$$\hat{\beta} = 1.0761.$$

$$\hat{\mu} = -5 \times 10^{-5}.$$

$$\hat{\sigma} = 0.0062.$$

The plot of the kernel smoothed density estimate of the data (black line) as well as the plot of the lognormal-normal density (blue line) using the estimated parameters are provided in figure 8 with the plot of the lognormal-normal density using the estimated parameters is given as:

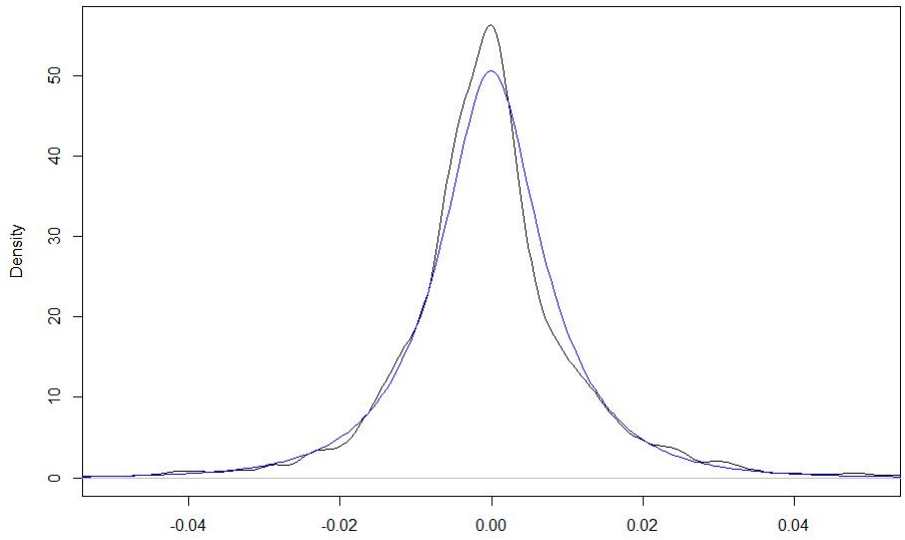


Figure 7: lognormal-normal model

The lognormal-normal distribution visually appears to model the log-returns quite well.

5.4 The ARCH model

Method of estimation

One thousand random uniform values are generated for each parameter and the log-likelihood of all these parameters are calculated. The parameter set that generates the lowest log-likelihood, since we are calculating the negative log-likelihood in our log-likelihood function, will be used as the starting parameter set. The optimization function built into R optimizes to a local minimum by default which is

the reason for calculating the negative log-likelihood. Once the optimization function in R has run, we then have the maximum likelihood estimates for the parameter values.

The stationary distribution of the ARCH(1) model is not known in closed form. In order to estimate this density we proceed as follows. Using the estimated parameters we generate a time series. Because each data point is generated using the previous data point there is some dependency between the generated data points and the randomly generated data points but that dependency is reduced the further away the data points are from one another. Therefore we assume that the thousandth data point is approximately independent of the first data point. This step is repeated 200 times and the thousandth data point is recorded each time in order to obtain a sample of 200 independent data points from which to draw a kernel density estimate. The kernel density estimate created from the simulated data is then compared to the kernel density estimate of the observed log-returns data.

Estimation results

The parameter estimates returned by the program are:

$$\hat{\alpha} = 0.3921.$$

$$\hat{\mu} = -0.0007.$$

$$\hat{\delta} = 0.0001.$$

Figure 9 shows the kernel density estimates of the observed log-returns data (black line) and the kernel density estimate of the simulated log-returns data (blue line).

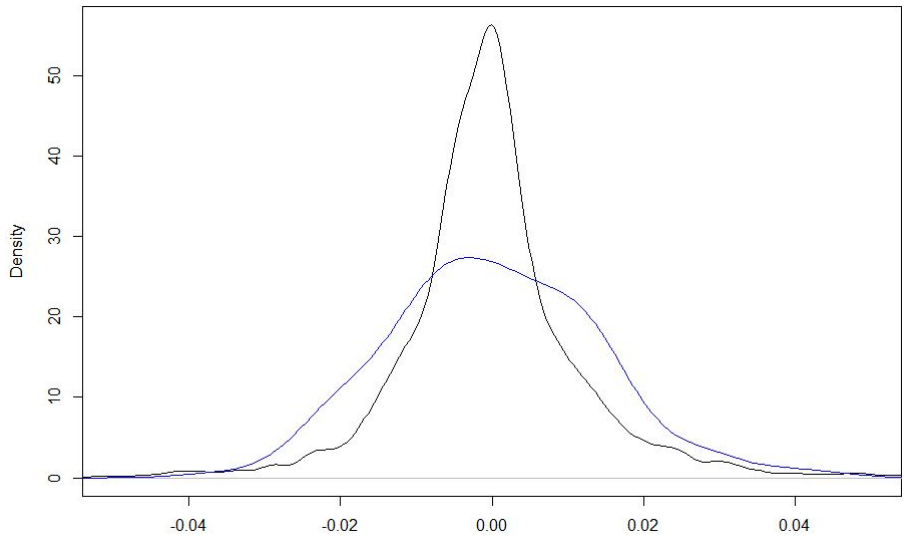


Figure 8: ARCH model

As we can see that the ARCH(1) model is not an idea fit to the observed data.

5.5 The GARCH model

Method of estimation

The method used to estimate the parameters for the GARCH model was similar to that used to estimate parameters for the ARCH model except with an alteration to the likelihood function that incorporates the added variable that requires estimation. One thousand random uniform values are generated for each parameter and the log-likelihood of all these parameters are calculated. The parameter set that generates the lowest log-likelihood, since we are calculating the negative log-likelihood in our log-likelihood function, will be used as the starting parameter set. The optimization function built into R optimizes to a local minimum by default which is the reason for calculating the negative log-likelihood. Once the optimization function in R has run, we then have the maximum likelihood estimates for the parameter values.
The stationary distribution of the GARCH(1,1) model is not known in closed form. In order to estimate this density we proceed as follows. Using the estimated parameters we generate a time series. Because each data point is generated using the previous data point there is some dependency between the generated data points and the randomly generated data points, more dependency than the ARCH(1) model due to part of the previous variation being used to estimate the current variation, but that dependency is reduced the further away the data points are from one another. Therefore we assume that the thousandth data point is approximately independent of the first data point. This step is repeated 200 times and the thousandth data point is recorded each time in order to obtain a sample of 200 independent data points from which to draw a kernel density estimate. The kernel density estimate created from the simulated

data is then compared to the kernel density estimate of the observed log-returns data.

Estimation results

The parameter estimates returned by the program were:

$$\hat{\alpha} = 0.1771.$$

$$\hat{\beta} = 0.7675.$$

$$\hat{\mu} = -0.0004.$$

$$\hat{\delta} = 6 \times 10^{-6}.$$

Figure 10 shows the kernel density estimates of the observed log-returns data (black line) and the kernel density estimate of the simulated log-returns data (blue line):

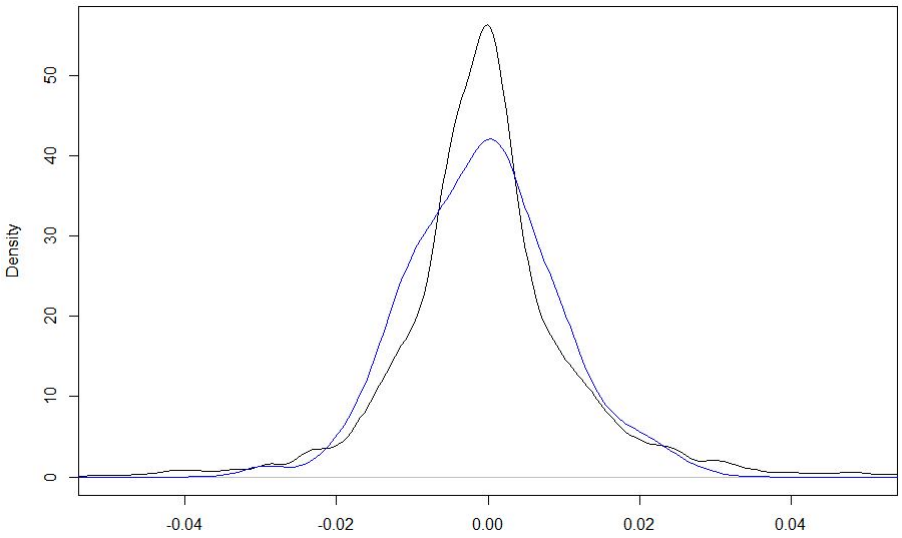


Figure 9: GARCH model

We see that the GARCH(1,1) is a better fit than the ARCH(1) but this model is still not ideal.

6 Evaluation of the models

For each of the models used we calculate a quantitative measure of how well the model fits the empirical data. In this study we use the integrated squared error (ISE) as the mentioned measure of fit. This function measures the squared distance between the empirical density estimate and the modeled density,

Model	Integrated squared error	Computation time (s)
The Black-Scholes model	4.82	0.06
The N∘IG model	0.10	34.7
The Lognormal-normal model	0.50	24.1
The ARCH(1) model	4.81	23.3
The GARCH(1,1) model	1.98	21.6

Table 2: Model accuracy table

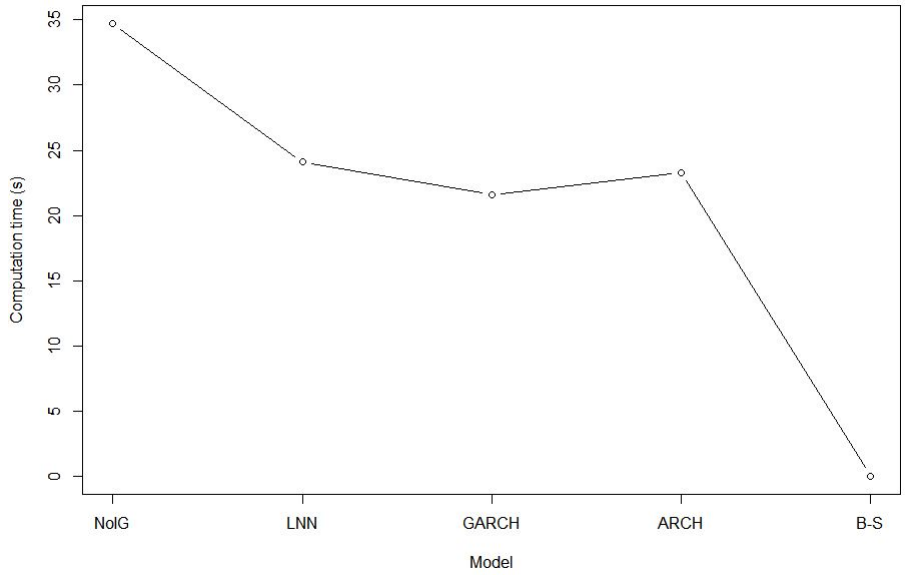


Figure 10: Model computation time

and integrates this squared distance over the support of the estimated density. The integrated squared error is calculated as follows:

$$ISE = \int (f(x) - f_e(x))^2 dx,$$

where $f(x)$ is the density of the fitted model at the point x and $f_e(x)$ is the estimated empirical density at the point x. f_e is estimated using kernel estimation and the bandwidth h is selected using Silverman's rule of thumb.

An ideal fitting model will have $f(x) = f_e(x)$ for all x, this will mean that the theoretical model fits the empirical data perfectly and that there will be no errors. When evaluating the equation, the closer the values of $f(x)$ and $f_e(x)$ are to each other, the smaller the ISE value will become. Therefore the model that returns the ISE value closest to 0 will be the model that best fits the empirical data.

Below we table the integrated squared error and computation time for each model fitted to the observed log-returns:

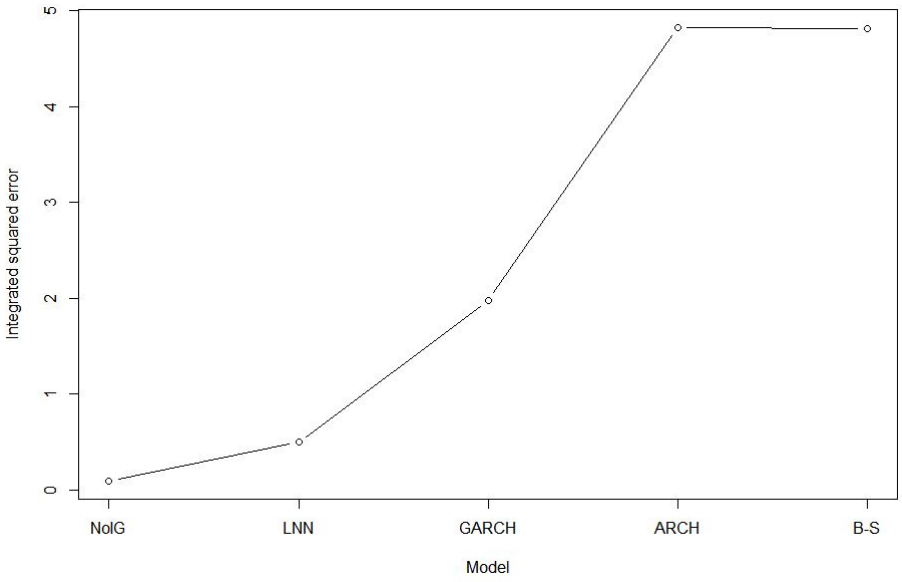


Figure 11: ISE by model

7 Conclusion

In this study we fitted 5 models to the observed log-returns of the S&P500. The Black-Scholes model, the N∘IG model, the lognormal-normal model, a ARCH(1) model and a GARCH(1,1) model. Based on the observed relationship between accuracy and program runtime, we can note that the cost of improved accuracy is an increase in the amount of time it takes to run the program. I.e., we observe that the N∘IG model is highly accurate with an integrated squared error of 0.101 and a runtime of 37.7 seconds while the Black-Scholes model takes a mere 0.06 seconds to run but the accuracy, with an integrated squared error of 4.82, is the worst of all the models fitted.

References

[1] P.K. Clark. A subordinated stochastic process model with finite variance for speculative prices. *Econometrica*, 41(1):135–155, 1973.

[2] R. Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance Volume 1, England*, 2001.

[3] J. Fernandes. Fitting the normal inverse Gaussian distribution to the S&P500 stock return data. Master's thesis, University of Massachusetts, 2012.

[4] A. Hanssen and T. Oigard. The normal inverse Gaussian distribution: a versatile model for heavy-tailed stochastic processes. *Proceedings - ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing 6:3985-3988 Â· May*, 2001.

[5] K.F. Kroner, T. Bollerslev, and R.Y. Chou. (ARCH) modeling in finance. *Journal of Econometrics*, 52(1):5 – 59, 1992.

[6] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[7] D.S. Ruppert, D. Matteson. *Statistics and Data Analysis for Financial Engineering: with R examples*. Springer Texts in Statistics. Springer New York, 2015.

[8] W. Schoutens. *Levy Processes in Finance: Pricing Financial Derivatives*. Wiley Series in Probability and Statistics. Wiley, 2003.

[9] P. Tankov. *Financial Modelling with Jump Processes*. Chapman and Hall/CRC Financial Mathematics Series. Taylor & Francis, 2003.

[10] I.J.H. Visagie. Development of an advanced volatility model for generating market consistent asset price paths for south african equities. Master's thesis, North-West University, 2010.

[11] I.J.H. Visagie. *On the calibration of Levy option pricing models*. PhD thesis, North-West University, 2015.

Appendix

7.1 Black-Scholes code

```
#Start monitoring time taken to run the code
time = proc.time()
#Set working directory, import libraries
setwd("C:/Users/Vincent/Documents/2017/Semester 1/Reseach Project - WST 795/Data sets")
#Input my data data = read.table("SP500.csv", sep = ",")
close = data[,2]
close = na.omit(close)
close = close[length(close):1]
#Get the log returns
m = length(close)-1
logr = 1:m*0
for (i in 1:m)
{
logr[i] = log(close[i+1]/close[i])
}
#Plot desity of log returns
plot(density(logr))
#Getting the MLE paramter estimates
m = mean(logr)
s = sqrt(var(logr))
#Getting the plotting points
den = density(logr)
x = den$x
y = (1/sqrt(2*pi*s^2))*exp(-((x-m)^2)/(2*s^2))
plot(density(logr), xlim=c(-0.05, 0.05))
lines(x, y, col = "Blue", type = "l")
#Calculation of the estimation accuracy value
ise = (x[2]-x[1])*sum((y-den$y)^2)
totaltime = proc.time() - time
totaltime = totaltime[3]
```

7.2 N◦IG code

```
#Start monitoring time taken to run the code
time = proc.time()
#Set working directory, import libraries
setwd("C:/Users/Vincent/Documents/2017/Semester 1/Reseach Project - WST 795/Data sets")
library("ghyp", lib.loc="~/R/win-library/3.3")
library("Bessel", lib.loc="~/R/win-library/3.3")
#Input my data
data = read.table("SP500.csv", sep = ",")
close = data[,2]
close = na.omit(close)
close = close[length(close):1]
#Get the log returns
m = length(close)-1 logr = 1:m*0
for (i in 1:m)
{
logr[i] = log(close[i+1]/close[i])
}
#Plot desity of log returns
plot(density(logr), xlim=c(-0.05, 0.05))
#Calculate density function of log returns with certain parameters
NoIGDens = function(al, be, mu, del, data)
{
dens = 0 if (al > 0 && del > 0 && abs(be) < al)
{
phi = sqrt(del^2 + (data-mu)^2)
dens = del*al/pi*exp(del*sqrt(al^2-be^2)+be*(data-mu))/phi*BesselK(al*phi, 1)
}
return(dens)
}
#Calculate the negative log likelihood function
logDens = function(param)
{
logdens = log(NoIGDens(param[1], param[2], param[3], param[4], logr))
```

```
logdens = sum(logdens)
logdens = -1*logdens
return(logdens)
}
#Method 1 for finding starting values for parameters
bot1 = 9999999999
count = 1
test = 1:1000*bot1
for (a in 1:6)
{
for (b in 1:a-1)
{
for (m in -1:1)
{
for (s in 1:2)
{
par = c(a, b, m, s)
test[count] = logDens(par)
if (test[count] < bot1) #Finding the index value for the lowest log likelihood
{
bot1 = test[count]
indBot1 = par
}
count = count + 1
}
}
}
}
startParam1 = indBot1
#Method 2 for finding starting values for parameters
n = 1000
alp = runif(n, 1, 150)
bet = runif(n, -120, 120)
muu = runif(n, -5, 5)
```

```
del = runif(n, 1, 100)
test2 = 1:n*0
bot2 = 9999999999
for (i in 1:n)
{
par2 = c(alp[i], bet[i], muu[i], del[i])
test2[i] = logDens(par2)
if (is.finite(test2[i])) #Finding the index value for the lowest log likelihood
{
if (test2[i] < bot2)
{
bot2 = test2[i]
indBot2 = i
}
}
}
startParam2 = c(alp[indBot2], bet[indBot2], muu[indBot2], del[indBot2])
#Optimizing (finding the minimum)
optimum1 = optim(startParam1, logDens)
optimum2 = optim(startParam2, logDens)
#Parameter estimates
alpha1 = optimum1$par[1]
beta1 = optimum1$par[2]
mu1 = optimum1$par[3]
delta1 = optimum1$par[4]
alpha2 = optimum2$par[1]
beta2 = optimum2$par[2]
mu2 = optimum2$par[3]
delta2 = optimum2$par[4]
optParam1 = c(alpha1, beta1, mu1, delta1)
optParam2 = c(alpha2, beta2, mu2, delta2)
logL1=logDens(optParam1)
logL2=logDens(optParam2)
optParam = optParam2
```

```
logL = logL2 if (logL1 < logL2)
{
optParam = optParam1
logL = logL1
}
cat("Maximum likelihood estimator: ", optParam) cat("Corresponding likelihood: ", -1*logL)
#Fitting the NoIG using the estimated parameters
den = density(logr)
x = den$x
y1 = NoIGDens(alpha1, beta1, mu1, delta1, x)
y2 = NoIGDens(alpha2, beta2, mu2, delta2, x)
y = NoIGDens(optParam[1], optParam[2], optParam[3], optParam[4], x)
plot(x, y1, col = "Blue", type = "l")
lines(x, y2, col = "Pink", type = "l")
lines(density(logr))
plot(density(logr))
lines(x, y, col = "Blue", type = "l")
#Calculation of the estimation accuracy value
ise = (x[2]-x[1])*sum((y-den$y)^2)
totaltime = proc.time() - time
totaltime = totaltime[3]
```

7.3 Lognormal-normal code

```
#Start monitoring time taken to run the code
time = proc.time()
#Set working directory, import libraries
setwd("C:/Users/Vincent/Documents/2017/Semester 1/Reseach Project - WST 795/Data sets")
library("ghyp", lib.loc="~/R/win-library/3.3")
library("Bessel", lib.loc="~/R/win-library/3.3")
#Input my data
data = read.table("SP500.csv", sep = ",")
close = data[,2]
close = na.omit(close)
close = close[length(close):1]
```

```
#Get the log returns
m = length(close)-1 logr = 1:m*0
for (i in 1:m)
{
logr[i] = log(close[i+1]/close[i])
}
#Plot desity of log returns
plot(density(logr), xlim=c(-0.05, 0.05))
#Calculate lognormal-normal density function of log returns given certain parameters
LnnDens = function(al, be, mu, sig, data)
{
dens = 0
y = 0
if (sig > 0 && be > 0)
{
int = 0; for(i in 1:200)
{
y = i*0.25
int = int + (y^(-3/2))*exp(-1*((data-mu*y)^2)/(2*y*sig^2)-((log(y)-al)^2)/(2*be^2))*0.25 }
dens = 1/(2*pi*sig*be)*int
}
return(dens)
}
logDens = function(param)
{
logdens = log(LnnDens(param[1], param[2], param[3], param[4], logr))
logdens = sum(logdens)
logdens = -1*logdens
return(logdens)
}
n = 500
alp = runif(n, -1, 10)
bet = runif(n, 0, 5)
muu = runif(n, -0.5, 0.5)
```

```
sigm = runif(n, -2, 2)
test = 1:n*0
bot = 9999999999
for (i in 1:n)
{
par = c(alp[i], bet[i], muu[i], sigm[i])
test[i] = logDens(par)
if (is.finite(test2[i])) #Finding the index value for the lowest log likelihood
{
if (test2[i] < bot2)
{
bot2 = test2[i]
indBot2 = i
}
}
}
startParam = c(alp[indBot], bet[indBot], muu[indBot], sigm[indBot])
optimum = optim(startParam, logDens)
optParam = optimum$par
#Parameter estimates
alpha = optParam[1]
beta = optParam[2]
mu = optParam[3]
sig = optParam[4]
#Plotting of the graphs
den = density(logr)
x = den$x
y = LnnDens(alpha2, beta2, mu2, sig2, x)
plot(density(logr))
lines(x, y, col = "Blue", type = "l")
#Calculation of the estimation accuracy value
ise = (x[2]-x[1])*sum((y-den$y)^2)
#End monitoring time taken to run the code
totaltime = proc.time() - time
```

```
totaltime = totaltime[3]
```

7.4 ARCH code

```
#Start monitoring time taken to run the code
time = proc.time()
#Set working directory
setwd("C:/Users/Vincent/Documents/2017/Semester 1/Reseach Project - WST 795/Data sets")
#Input my data
data = read.table("SP500.csv", sep = ",")
close = data[,2]
close = na.omit(close)
close = close[length(close):1]
#Get the log returns
m = length(close)-1
logr = 1:m*0
for (i in 1:m)
{
logr[i] = log(close[i+1]/close[i])
}
plot(density(logr))
#Calculation of the likelihood
like = function(par)
{
d = 1:(m-1)*0 h = 1:m*0
if(par[1] > 0 && par[3] > 0 && par[1] < 1)
{
h[1] = par[3]/(1-par[1])
e = logr-par[2]
for (i in 2:m)
{
d[i-1] = dnorm(logr[i-1], mean=par[2], sd=sqrt(h[i-1]))
h[i] = par[3] + par[1]*e[i-1]^2
}
}
```

```
return(d)
} #Calculation of the log-likelihood
loglike = function(par)
{
ll = like(par)
ll = log(ll)
ll = sum(ll)
ll = -1*ll return(ll)
}
#Estimating starting parameters
n = 1000
al = runif(n, 0.0000001, 1)
mu = runif(n, -0.001, 0.001)
del = runif(n, 0.0000001, 1)
test = 1:n*0
bot = 9999999999
for (i in 1:n)
{
par = c(al[i], mu[i], del[i])
test = loglike(par)
if (is.finite(test)) #Finding the index value for the lowest log likelihood
{
if (test < bot)
{
bot = test
indBot = i
}
}
}
#Optimization of the starting parameter estimates
sPar = c(al[indBot], mu[indBot], del[indBot])
optimum = optim(sPar, loglike)
oPar = optimum$par
#Simulation of the model to obtain independent data points
```

```
m = 200
dist = 500
h = 1:dist*0
err = 1:dist*0
simr = 1:dist*0
use = 1:m*0
h[1] = oPar[3]/(1-oPar[1])
for (j in 1:m)
{
rand = rnorm(dist, mean = 0, sd = 1)
err[1] = rand[1]*sqrt(h[1])
simr[1] = oPar[2] + err[1]
for (i in 2:dist)
{
h[i] = oPar[3] + oPar[1]*err[i-1]^2
err[i] = rand[i]*sqrt(h[i])
simr[i] = oPar[2] + err[i]
}
use[j] = simr[dist]
}
#Finding the graph plotting points
denl = density(logr)
x = denl$x
denu = density(use, from=min(x), to=max(x))
#Plotting the kernel density estimates
plot(denl, xlim=c(-0.05, 0.05))
lines(denu, xlim=c(-0.05, 0.05))
#Calculation of the estimation accuracy value
ise = (x[2]-x[1])*sum((denl$y-denu$y)^2)
#End monitoring time taken to run the code
totaltime = proc.time() - time
totaltime = totaltime[3]
```

7.5 GARCH code

```
#Start monitoring time taken to run the code
time = proc.time()
#Set working directory
setwd("C:/Users/Vincent/Documents/2017/Semester 1/Reseach Project - WST 795/Data sets")
#Input my data
data = read.table("SP500.csv", sep = ",")
close = data[,2]
close = na.omit(close)
close = close[length(close):1]
#Get the log returns
m = length(close)-1
logr = 1:m*0
for (i in 1:m)
{
logr[i] = log(close[i+1]/close[i])
}
plot(density(logr))
#Calculation of the likelihood
like = function(par)
{
d = 1:(m-1)*0
h = 1:m*0
if(par[1] > 0 && par[2] > 0 && par[4] > 0 && (par[1] + par[2]) < 1)
{
h[1] = par[4]/(1-par[1]-par[2])
e = logr-par[3] for (i in 2:m)
{
d[i-1] = dnorm(logr[i-1], mean=par[3], sd=sqrt(h[i-1]))
h[i] = par[4] + par[2]*h[i-1] + par[1]*e[i-1]^2
}
}
return(d)
}
```

```
#Calculation of the log-likelihood
loglike = function(par)
{
ll = like(par)
ll = log(ll)
ll = sum(ll)
ll = -1*ll
return(ll)
}
#Estimating starting parameters
n = 1000
al = runif(n, 0.0000001, 1)
be = runif(n, 0.0000001, 1)
mu = runif(n, -0.001, 0.001)
del = runif(n, 0.0000001, 1)
bot = 9999999999
for (i in 1:n)
{
par = c(al[i], be[i], mu[i], del[i])
test = loglike(par)
if (is.finite(test)) #Finding the index value for the lowest log likelihood
{
if (test < bot)
{
bot = test
indBot = i
}
}
}
#Optimization of the starting parameter estimates
sPar = c(al[indBot], be[indBot], mu[indBot], del[indBot])
optimum = optim(sPar, loglike)
oPar = optimum$par
#Simulation of the model to obtain independent data points
```

```
it = 200
dist = 1000
h = 1:dist*0
err = 1:dist*0
simr = 1:dist*0
use = 1:it*0
h[1] = oPar[4]/(1-oPar[1]-oPar[2])
for (j in 1:it)
{
rand = rnorm(dist, mean = 0, sd = 1)
err[1] = rand[1]*sqrt(h[1])
simr[1] = oPar[3] + err[1]
for (i in 2:dist)
{
h[i] = oPar[4] + oPar[2]*h[i-1] + oPar[1]*err[i-1]^2
err[i] = rand[i]*sqrt(h[i])
simr[i] = oPar[3] + err[i]
}
use[j] = simr[dist]
}
#Finding the graph plotting points
denl = density(logr)
x = denl$x
denu = density(use, from=min(x), to=max(x))
#Plotting the kernel density estimates
plot(denl, xlim=c(-0.05, 0.05))
lines(denu, xlim=c(-0.05, 0.05))
#Calculation of the estimation accuracy value
ise = (x[2]-x[1])*sum((denl$y-denu$y)^2)
#Calculation of the time taken to run the program
totaltime = proc.time() - time
totaltime = totaltime[3]
```

7.6 SAS Code

```
proc import datafile="C:\Users\Vincent\Documents\2017\Semester 1\Reseach Project - WST 795\Data
sets\Sp500 - logr.csv"
dbms=csv
out=sasuser.SP500
replace;
proc univariate data = sasuser.SP500 normal;
histogram logr /normal;
run;
```

Latent Dirichlet allocation applied to forensic data

Brent Albert Dreyer 12092942

STK795 Research Report

Submitted in partial fulfillment of the degree BCom(Hons) Statistics

Supervisor: Ms J. Mazarura, Co-supervisor: Dr. A. de Waal

Department of Statistics, University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

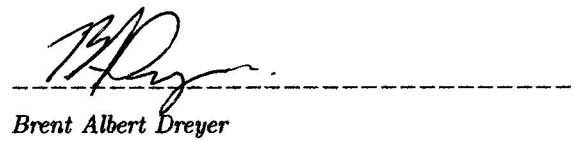
30 October 2017

Abstract

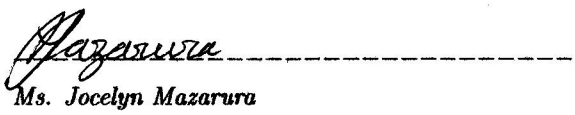
In the modern world we have the ability to store vast amounts of digitized data. Data, however, means nothing if it cannot be utilised. Large amounts of unlabelled data can be very problematic if one wants to extract relevant information from it. Topic modelling is a tool that has been created to address this problem by allocating similar documents to a certain topic. This makes extracting information much easier, since the documents are now clustered according to topics. In this research project we will describe and make use of latent Dirichlet allocation (LDA) to allocate a large amount of forensic data to topics. LDA is arguably the most popular topic model and uses sophisticated methods to enable a computer to identify topics in a manor that humans would. This will enable us to use the data more efficiently and extract meaningful information much faster. Inference will be done using the collapsed Gibbs sampler.

Declaration

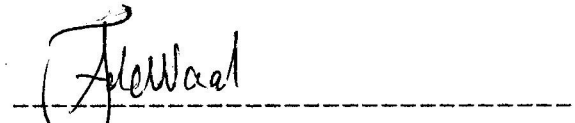
I, *Brent Albert Dreyer,* declare that this essay, submitted in partial fulfillment of the degree *BCom(Hons) Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.



Brent Albert Dreyer



Ms. Jocelyn Mazarura



Dr. Alta de Waal

20/10/2017

Date

Acknowledgements

I would like to express my profound gratitude to my supervisor, Jocelyn Mazarura, for her support, guidance and immense knowledge. I am ever grateful for her mentorship and could not have hoped for a better supervisor. I would also like to thank my co-supervisor, Dr Alta de Waal for her support.

The financial assistance of STATOMET and the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF. I would also like to thank the Department of Statistics at the University of Pretoria for giving me the opportunity to do research.

I thank my mother Rene Dreyer and my brothers Bernard and Devin Dreyer for supporting me through all the hard work. A special thanks to my significant other, Megan Lombard, for keeping me company while working late nights and providing me with insightful ideas.

Contents

List of Figures

List of Tables

1 Introduction

Wisdom, knowledge, information and data are some of the most valuable assets we possess. We have travelled into space and achieved incredible things. Luckily the innovators and scientists of today have the privilege of standing on the shoulders of giants. According to a study done by Buckminster Fuller[1], human knowledge doubled every century up until 1900. With the competitive and urgent environment created by the first and second world wars, knowledge doubled every twenty five years. Today knowledge is doubling every twelve months. The good news is that with this accumulated knowledge we can achieve things beyond our wildest imaginations. However, it is not always easy to access information, because a lot of it still lies hidden in vast amounts of data. The world has become a digital domain of data which exists in the form of documents, social networks, and any form of digital memory. Most of the time the data is unstructured and unlabelled. It becomes increasingly difficult to pinpoint what we are looking for when we are faced with massive collections of unlabelled data. Presently, search engines use key words to find relevant information. Topic modelling algorithms use statistical methods to discover themes and how those themes change over time. They do this by analysing the words in those texts through complex methods. With this new technology organising and utilising large amounts of raw data has become a reality.

The main goal of topic modelling is to automatically discover topics in documents by looking at the distribution of words in each document. There are many topic models that have been developed, but we will focus on latent Dirichlet allocation (LDA) [2] which is arguably the most popular topic model. In this research project, the LDA model will be explained in detail. As a practical example, it will be used to extract topics from forensic data. The collapsed Gibbs sampler will be used to do inference on the LDA model.

Our colleagues at the forensic science department approached us to help organise their Autopsy reports. Over the years this data have been accumulating and as a result, has become problematic to filter through and find document with certain traits. We received one thousand six hundred and sixty nine post mortem examination reports to analyse. We applied LDA to find latent topics embedded in the documents. The popular Python package*gensim* was used to preprocess the data and implent the model. The output is a comprehensive understanding of semantically similar documents which enables navigation through documents.

2 Background Theory

In this section, concepts relevant to this paper will be explained in order to understand the work more clearly.

[1] http://www.industrytap.com/knowledge-doubling-every-12-months-soon-to-be-every-12-hours/3950

2.1 Independent and identically distributed

Random variables are said to be independent and identically distributed (iid), if they follow the same probability distribution and are mutually independent. In other words, two events are said to be iid if the occurrence of one does not give any information as to whether the second event occurred or not. In particular, the probability we ascribed to the second event is not affected by the knowledge that the first event has occurred. The assumption of iid variables is the core of many statistical theorems, such as the central limit theorem, which states that the probability distribution of the mean of iid variables approaches a normal distribution.

2.2 Conditional independence

Figure 1 is used to explain the concept of conditional independence. The outcome of the stochastic variables X_1 and X_2 are dependent on variable A. By definition, if X_1 and X_2 are conditionally independent $P(X_1, X_2|A) = P(X_1|A)$. In other words, if A is assumed to be known then the outcome of X_1 does not influence the outcome of X_2, similarly, the outcome of X_2 does not influence the outcome of X_1. For instance if $P(X_1|A, X_2) = P(X_1|A)$ it can be said that X_1 and X_2 are conditionally independent given that A is known.

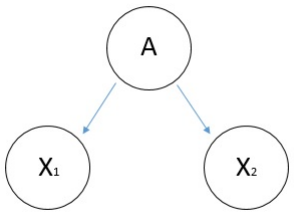


Figure 1: Illustration of independent and identically distributed variables

2.3 Exchangeability

Exchangeability is an independence relation stronger than conditional independence. The set of random variables $Y_1, Y_2, ..., Y_n$ is exchangeable if their joint probability, $p(y_1, ..., y_n)$, is invariant to permutation of the indices. That is, for any permutation π, $p(y_1, ..., y_n) = p(y_{\pi(1)}, ..., y_{\pi(n)})$.

2.4 Sufficient statistics

This definition can be explained heuristically as follows. Suppose there are two persons, namely A and B. Person A knows the entire random sample and person B only knows the value of T. It can be said that T is

a sufficient statistic if person B can estimate the unknown parameter θ just as good as person A. According to the mathematical definition the statistic $T = r(X_1, X_2, ..., X_n)$ is a sufficient statistic if, for each t, the conditional distribution of $X_1, X_2, ..., X_n$ given $T = t$ and θ, does not depend on θ.[2]

2.5 The exponential family

The exponential family, as explained in [8], is a range of distributions that can be manipulated into a pdf $p(x|\theta)$, for $x = (x_1, ..., x_m) \in X^m$ and $\theta \in R^d$ that is in the form:

$$p(x|\theta) = \frac{1}{Z(\theta)} h(x) exp[\theta^T \phi(x)] \tag{1}$$

$$= h(x) exp[\theta^T \phi(x) - A(\theta)] \tag{2}$$

where

$$Z(\theta) = \int_{X^m} h(x) exp[\theta^T \phi(x)] dx \tag{3}$$

$$A(\theta) = logZ(\theta) \tag{4}$$

The $\vec{\theta}$ are the natural parameters and $\phi \in R^d$ is a vector of sufficient statistics. The partitioning function and the log partitioning function are denoted by $Z(\theta)$ and $A(\theta)$ respectively, and $h(x)$ is the scaling constant. One important property of the exponential family is that all members have a conjugate priors. Given the data D, a family F of prior distributions $p(\theta)$ is conjugate to a likelihood $p(D|\theta)$ if the posterior $p(\theta|D)$ is in F. Having a conjugate prior simplifies the computation of the posterior. This will prove to be helpful in LDA.

2.6 The Gibbs sampler

The Gibbs sampler uses a method called Markov chain Monte Carlo (MCMC). The idea is that we sample one variable at a time conditioned on all the other variables. Consider that we want to sample from the distribution $p(z) = p(z_1, ..., z_M)$, and suppose that the initial state for the Markov chain has been chosen. In each step of the procedure, the value z_i will be replaced by a value drawn from the distribution based on all the remaining values $p(z_i|Z_{(i)})$, where z_i is the i^{th} value of Z and $Z_{(i)}$ denotes $z_1, ..., z_M$ but with z_i omitted.

We will now use a distribution $p(z_1, z_2, z_3)$ over three variables as an example. At step τ in the algorithm

[2] http://math.arizona.edu/ tgk/466/sufficient.pdf.

we have selected values $z_1^{(\tau)}, z_2^{(\tau)}$ and $z_3^{(\tau)}$. First we sample from the conditional distribution

$$p(z_1|z_2^{(\tau)}, z_3^{(\tau)}). \tag{5}$$

We have now obtained a new value $z_1^{(\tau+1)}$ that will replace the old value $z_1^{(\tau)}$ in future sampling. Next we will sample a new value for $z_2^{(\tau)}$ from the conditional distribution

$$p(z_2|z_1^{(\tau+1)}, z_3^{(\tau)}). \tag{6}$$

Now that we have $z_1^{(\tau+1)}$ and $z_2^{(\tau+1)}$, we will use them in the conditional distribution

$$p(z_3|z_1^{(\tau+1)}, z_2^{(\tau+1)}) \tag{7}$$

to obtain a replacement value $z_3^{(\tau+1)}$ for $z_3^{(\tau)}$. This process of cycling through and sampling continues until a steady state has been reached.

According to [1] the algorithm for Gibbs sampling, in general, can be summarised as follows:

Algorithm 1 The Gibbs sampler

1: Initialise $(\mu_i : i = 1, ..., M)$

2: For $\tau = 1, ..., T$:

- Sample $\mu_1^{(\tau+1)} \sim p(\mu_1|\mu_2^{(\tau)}, \mu_3^{(\tau)}, ..., \mu_M^{(\tau)})$.
- Sample $\mu_2^{(\tau+1)} \sim p(\mu_2|\mu_1^{(\tau+1)}, \mu_3^{(\tau)}, ..., \mu_M^{(\tau)})$.
- $\vdots$
- Sample $\mu_j^{(\tau+1)} \sim p(\mu_j|\mu_1^{(\tau+1)}, ..., \mu_{j-1}^{(\tau+1)}, \mu_{j+1}^{(\tau)} ..., \mu_M^{(\tau)})$.
- $\vdots$
- Sample $\mu_M^{(\tau+1)} \sim p(z_M|\mu_1^{(\tau+1)}, \mu_2^{(\tau+1)}, ..., \mu_{M-1}^{(\tau+1)})$.

Experiment:

Table 1 illustrates the the mean values as estimated by the Gibbs sampler against the true initial mean values. It is clearly shown that the estimated mean values is very close to the actual mean values. The algorithm was implemented using SAS version 9.4 and the code can be found in Appendix A.

i	$\hat{\mu_i}$	μ_i
1	29.6016	30
2	-9.99691	-10
3	4.97329	5

Table 1: Mean values estimated by the Gibbs sampler against the true mean values as initialised

2.7 The Dirichlet distribution

The Dirichlet distribution is a distribution on probability distributions. Suppose $\theta \sim Dir(\alpha)$, where the density function of θ is given by:

$$p(\theta) = \frac{1}{\beta(\alpha)} \prod_{i=1}^{n} \theta_i^{\alpha_i - 1}, \quad I(\theta \in S) \tag{8}$$

where

- $\theta = (\theta_1, ..., \theta_n)$ and $\sum_{i=1}^{n} \theta_i = 1$
- $\alpha = (\alpha_1, ..., \alpha_n)$, $\alpha_i > 0$ and $\alpha_0 = \sum_{i=1}^{n} \alpha_i$.
- The probability simplex is given by $S = (X \in R^n : x_i \geq 0, \quad \sum_{i=1}^{n} x_i = 1)$

We can rewrite the β function as $\frac{1}{\beta(\alpha)} = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{S} \Gamma(\alpha_i)}$ and thus the Dirichlet distribution can be rewritten as:

$$p(\theta) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{S} \Gamma(\alpha_i)} \prod_{i=1}^{n} \theta_i^{\alpha_i - 1}, \quad I(\theta \in S) \tag{9}$$

The Dirichlet distribution is part of the exponential family. As previously stated, an advantageous feature of the exponential family is that the distributions always have a conjugate prior. This will be useful, since the Dirichlet distribution is a conjugate prior to the multinomial distribution. For a detailed proof the reader is referred to [5].

3 Notation

- The vocabulary is the body of words in the corpus that we will be examining through the LDA process. It is represented by a V-vector.

- Words are units of discrete data and are represented in vector form, making up the entire vocabulary.
- N is the total number of words in a document. The words are denoted by $W = (w_1, w_2, ..., w_N)$, where w_n is the n^{th} word in the document.
- M is the total number of documents in a corpus. The documents are donated by $D = (W_1, W_2, ...W_M)$, where W_m is the m^{th} document in the corpus.
- K is the number of topics.
- $\vec{\alpha}$ is the hyperparameter on the mixing proportions (K-vector).
- $\vec{\beta}$ is the hyperparameter on the mixing components (V-vector).
- $\vec{\theta}_m$ is the topic proportion for document m.
- $\vec{\phi}_k$ is the mixture component for topic k.

4 Literature review

In this section the literature of the development of LDA and and other text analysing models will be explored. This will serve as a good foundation in order to understand key concepts in text analysis.

We have the ability to store text corpora electronically, but it is useless if we cannot analise and utilise the massive amount of data stored. The aim is to reduce text corpora (in document form) to short descriptions of the documents so that efficient processing of the large collection of documents can be performed without losing the essential statistical relationships that are useful for relevant judgments[3]. Methods have been developed to address this problem and it has proven to be a challenging, but rewarding process. At first the term frequency-inverse document frequency (*tf-idf*) [11] scheme was introduced. The *tf-idf* scheme is a weighting factor for variables. The weight increases as the word frequency in a document increases, but that is offset by the number of times the word appears in the entire corpus. The scheme is illustrated in Table 2. We have to get a vocabulary from all the documents. Thereafter, the number of times each word appears in a document has to be counted, this is the word count. We then use the word count to calculate the term frequency for every word by dividing the word count for each individual word by the word total in that document. Given the total number of documents collected, D, and the number of documents containing the word, $f_{w,D}$, an inverse document frequency (*idf*) term is calculated using $idf = \ln\left(\frac{D}{f_{w,D}}\right)$ [10]. We then multiply the *idf* with the term frequency to obtain a *tf-idf* vector for each document.

	Word Count		Term Frequency				
	Document1	Document2	Document1	Document2	idf	tf-idf1	tf-idf2
Word 1	0	1	0	0.125	0.69	0	0.08625
Word 2	1	1	0.125	0.125	0	0	0
Word 3	2	1	0.25	0.125	0	0	0
Word 4	1	0	0.125	0	0.69	0.08625	0
Word 5	1	1	0.125	0.125	0	0	0
Word 6	2	2	0.25	0.25	0	0	0
Word 7	0	1	0	0.125	0.69	0	0.08625
Word 8	1	1	0.125	0.125	0	0	0
Total	8	8	1	1			

Table 2: An illustration of the *tf-idf* scheme with two documents containing eight words each

Thus we are left with a term by document matrix, denoted X, where the columns contain the *tf-idf* values for each document. The more frequent a word is observed in different documents, the less significant it becomes. This can be seen in the *tf-idf* columns, as the words that are observed in both documents have a weight of zero. The scheme has proven to reduce documents of arbitrary length to a fixed length vector of numbers. Searching for a relevant document using key words will be much more efficient after the *tf-idf* values for each word in each document has been obtained. The sum of the *tf-idf* values for each document, relative to keywords[3], will determine how relevant each document is. The document with the highest $\sum tf - idf$, will be of greatest significance. The scheme was compelling, because it was simple and efficient for matching keywords to documents. However, it lacked reduction in description length and did not reveal much statistical structure between or inside documents. This problem was addressed by *latent semantic indexing* (LSI) [4] which achieved significant compression of big data collections. This was accomplished by breaking down the X matrix to a linear subspace which captures a high percentage of the variance in the data set. In effect LSI does not only look at the frequency of words appearing in the same document, but it also compares how often those occurrences happen in all of the documents in the corpus. Hofmann [7] improved LSI by introducing the *probabilistic LSI* (pLSI) model as an alternative to LSI. Before understanding pLSI, the unigram model and mixture of unigrams will be explained for clarification. As illustrated in Figure 2(a), the unigram model draws every word independently. The words are all drawn from a single multinomial distribution:

$$p(w) = \prod_{n=1}^{N} p(w_n) \tag{10}$$

[3]Keywords are words with great significance to the subject.

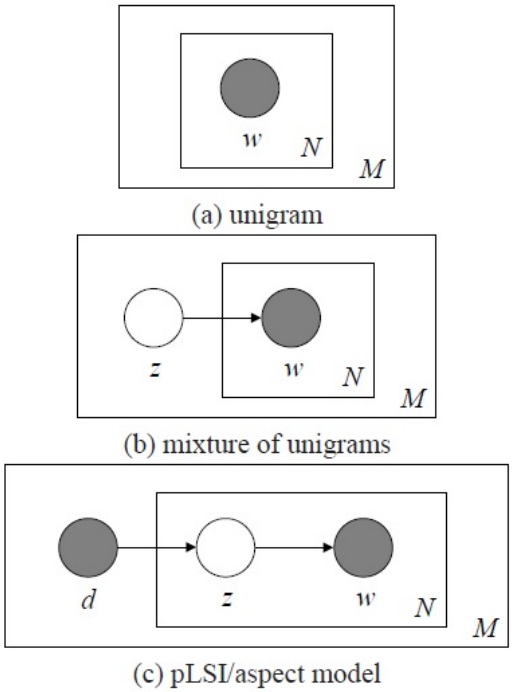


Figure 2: A Graphical illustration of the different models explained. The plates (rectangles) represents replicated structure, observed variables are represented by the shaded nodes and the unshaded nodes represent latent or hidden variables. The arrows between the nodes represent dependency between variables. In (b) it can be said that w is dependent on z. [3]

In the unigram model a topic assignment variable is still irrelevant since all the documents are assumed to have the same topic.

As illustrated in Figure 2(b), the mixture of unigrams model can be obtained by augmenting the unigram model with a discrete random variable z [9] . Under this model a document is generated using a slight different approach as the unigram model. We start by choosing a topic z. Thereafter, N words are independently generated from the conditional multinomial $p(w|z)$. Mathematically we can denote the probability of a document with the function:

$$p(w) = \sum_{z} p(z) \prod_{n=1}^{N} p(w_n|z). \tag{11}$$

This model assumes that each document has only one topic.

The pLSI model relaxes the assumption that each document is generated by only one topic. Figure 2(c)

illustrates that a document d and a word w_n are conditionally independent given an unobserved topic z:

$$p(d, w_n) = p(d) \sum_z p(w_n|z)p(z|d). \tag{12}$$

This means that a document can be modeled as a function over topics. Despite this research being a step forward in probability modeling of text, it has two weaknesses: the bigger the corpus is the more parameters are present, which leads to over-fitting, and assigning probability to documents outside of the training set is unclear [3]. This brings us to latent Dirichlet allocation which will be described in detail in section 5.

5 Latent Dirichlet allocation

LDA is traditionally used to detect underlying topics across a corpus of text documents. The basic concept of LDA is that a document exhibits multiple topics in different proportions. Words carry strong semantic information, thus documents with similar topics use a similar group of words. It is worth noting that a topic is defined as a distribution over a fixed vocabulary [2]. In other words, every topic contains a probability for every word from the fixed vocabulary. The latent topics are discovered by identifying groups of words in the corpus that frequently occur together within documents. It is important to understand that LDA is subject to the "bag of words" assumption, which assumes that only the identity and not the position of the words are relevant. This leads to an assumption of exchangeability for the words in a document and documents in a corpus. However, it does not mean that the random variables are *independent and identically distributed* (iid), instead it can be said that they are conditionally iid [3]. Due to the "bag of words" assumption, syntax's are irrelevant to the model. Only the distribution of words matter. While we would never be able to read the document, we will be able to identify the most obvious topics embedded in the document.
LDA and other topic models are part of the *probabilistic modeling* (PM) family. With generative PM, data is treated as if it is arising from hidden (latent) variables. In other words, each document can be thought of as a random mixtures over latent topics and the topics can be thought of as distributions over words.
Here follows the generative process that the LDA model assumed for every document W_m in corpus D [3].

1. A number of words N must be decided on, according to a Poisson distribution with parameter (ζ).
2. Choose a topic distribution $\vec{\theta}$ for the document with Dir($\vec{\alpha}$).
3. For each of the N words w_n:
   (a) First randomly choose a topic z_n according to the multinomial distribution with parameters ($\vec{\theta}$).

(b) Then randomly choose each word ,w_n, from a multinomial probability conditioned on the topic z_n: $p(w_n|\vec{z}_n, \vec{\phi})$.

The generative process can also be expressed graphically as in Figure 3. The only observable features are the words illustrated by the shaded node. All the other parameters are latent or hidden. The parameter denoted as z is the topic assignment of each word which makes each document a mixture of topics. ϕ

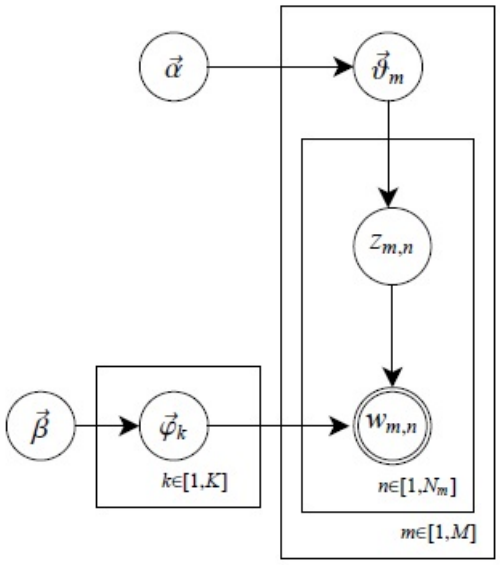


Figure 3: Graphical model representation of LDA [6]

The plates (rectangles) represent replicated structure. The outer plate represents the formation of the M documents and the inner plate represents repeated choice of topics and words in the document. The observed word is denoted by w and z is the topic for that word. The topic distributions for the documents is given by $\vec{\theta}$ and the word distribution for the topics is denoted by $\vec{\phi}$. The hyper-parameters $\vec{\alpha}$ and $\vec{\beta}$ are parameters of the Dirichlet distribution. The former is used to control topic distribution for each document and the latter is used to control the word distribution for each topic. When the α value is high each document is likely to be compiled out of most of the topics, while a low α means that the documents are more likely to be represented by just a few topics. When the β value is high each topic is more likely to be compiled out of most of the words, while a low β value means that the topics are more likely to be represented by just a few words.

To simplify the following assumtions are made. The number of topics, z, and therefore the dimensionality, k, of the Dirichlet distribution is decided on and fixed beforehand. The word probabilities are parameterised by a $k \times V$ matrix $\boldsymbol{\phi}$. The Dirichlet distribution is in the exponential family. It is a conjugate prior to the multinomial distribution and it has finite dimensional sufficient statistics [3]. The number of topics, k,

determines the dimensionality of the Dirichlet distribution where $\vec{\theta}$ can take on values in the $(k-1)$-simplex. The k-vector $\vec{\theta}$ lies in the $(k-1)$-simplex if $\theta_i \geq 0, \sum_{i=1}^{k} \theta_i = 1$. Here follows the probability density function of the Dirichlet distribution on this simplex: Here follows the probability density function of the K-dimensional Dirichlet distribution with parameters $\vec{\alpha}$:

$$p(\theta|\vec{\alpha}) = \frac{\Gamma(\sum_{i-1}^{k} \alpha_1)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}. \tag{13}$$

The parameters $\vec{\alpha}$ is a k-vector with $\alpha_i > 0$ and $\Gamma(x)$ is the Gamma function. The joint distribution of a topic mixture $\vec{\theta}$, a set of N topics , z, and a set of N words , w, given the parameters $\vec{\alpha}$ and $\vec{\phi}$ is given by:

$$p(\vec{\theta}, \vec{z}, \vec{w}|\vec{\alpha}, \vec{\phi}) = p(\vec{\theta}|\vec{\alpha}) \prod_{n=1}^{N} p(z_n|\vec{\theta}) p(w_n|z_n, \vec{\phi}). \tag{14}$$

The marginal distribution of a document is obtained by integrating over θ and summing over z, where $p(z_n|\theta)$ is simply θ for the unique i such that $z_n^i = 1$:

$$p(w|\vec{\alpha}, \vec{\phi}) = \int p(\theta|\vec{\alpha}) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \vec{\phi}) \right) d\theta. \tag{15}$$

To obtain the probability of a corpus, we must take the product of the marginal probabilities of single documents:

$$p(D|\vec{\alpha}, \vec{\phi}) = \prod_{d=1}^{M} \int p(\theta_d|\vec{\alpha}) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \vec{\phi}) \right) d\theta_d. \tag{16}$$

Both α and β are only sampled once in the corpus, where the variables θ_d and ϕ are sampled once for every document and topic respectively. The variables z_{dn} and w_{dn} are sampled for every word in a document.

5.1 The Gibbs sampler for latent variable models

As explained in section 2.6, the Gibbs sampler is a special case of MCMC. Therefore, it can be used to emulate high dimensional probability distributions $p(\vec{x})$ by the stationary behavior of a Markov chain. In Eq. 17 it is clear that all the dimensions, x_i, of the distribution are sampled one at a time and that they are conditioned on all of the other values of all the other dimensions denoted by $\vec{x}_{(i)}$. According to [6] the algorithm works as follows:

1. The dimension i should be chosen.

2. Using $p(x_i.\vec{x}_{(i)})$, x_i will be sampled.

The full conditions of a Gibbs sample must be found, it is possible using:

$$p(x_i|\vec{x}_{(i)}) = \frac{p(\vec{x})}{p(\vec{x}_{(i)})} \tag{17}$$

Using Eq. 17, the general formulation of a Gibbs sampler for models with hidden or latent variables $\vec{z}$ becomes:

$$p(z_i|\vec{z}_{(i)}, \vec{x}) = \frac{p(\vec{z}, \vec{x})}{p(\vec{z}_{(i)}, \vec{x})} \tag{18}$$

The concept of the Gibbs sampler for latent variable models will now be used to derive a collapsed Gibbs sampler for LDA. This will minimise uncertainty by "collapsing" away most of the unknown variables.

5.2 Inference via the collapsed Gibbs sampler

The collapsed Gibbs sampler will be described in the context of an LDA model. The idea behind the collapsed Gibbs sampler is that we can analytically marginalise over all of the uncertainty in our model parameters and just sample the word assignment variables z. We never have to sample our corpus-wide topic vocabulary distributions α and β or any of the per-document specific topic proportions θ. We just go through iterations, sampling the topic assignment for word variables z. Inference is done on the distribution $p(\vec{z}|\vec{w})$, which is proportional to the joint distribution:

$$p(\vec{z}|\vec{w}) = \frac{p(\vec{z}, \vec{w})}{p(\vec{w})} = \frac{\prod_{i=1}^{W} p(z_i, w_i)}{\prod_{i=1}^{W} \sum_{k=1}^{K} p(z_i = k, w_i)}. \tag{19}$$

This often leads to much better performance, because we are examining uncertainty in a smaller space. In essence, all of the model parameters can completely "collapse" away. All that is needed is to iteratively re-sample the word assignment variables for every word in the document and then for every word in the corpus. It is done by simulating $p(\vec{z}|\vec{w})$ using the full conditional $p(z_i|\vec{z}_{(i)}, \vec{w})$. The full conditional can be obtained by using the hidden-variable approach which requires the joint distribution from the Gibbs sampler. By assessing Eq. 18, the joint distribution in LDA can be factored as follows:

$$p(\vec{w}, \vec{z}|\vec{\alpha}, \vec{\beta}) = p(\vec{w}|\vec{z}, \vec{\beta})p(\vec{z}|\vec{\alpha}). \tag{20}$$

It can by shown that:

$$p(\vec{w}|\vec{z},\vec{\beta}) = \prod_{z=1}^{K} \frac{\Delta\vec{n}_z + \vec{\beta}}{\Delta(\vec{\beta})}, \quad \vec{n}_z = \left\{n_z^{(t)}\right\}_{t=1}^{V}. \tag{21}$$

and

$$p(\vec{z}|\vec{\alpha}) = \prod_{m=1}^{M} \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})}, \quad \vec{n}_m = \left\{n_m^{(k)}\right\}_{k=1}^{K}, \tag{22}$$

where

- $\vec{n}_z$ is the vector of word observation counts for topic z.
- $\vec{n}_m$ is the vector of topic observation counts for document m.

Thus, the joint distribution now becomes:

$$p(\vec{w},\vec{z}|\vec{\alpha}) = \prod_{z=1}^{K} \frac{\Delta\vec{n}_z + \vec{\beta}}{\Delta(\vec{\beta})} \cdot \prod_{m=1}^{M} \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})}. \tag{23}$$

Using the full conditional, each of the topics, z_i, that have been assigned to words should now sequentially be re-sampled given all the other topic assignments, $\vec{z}_{(i)}$, and all of the words, w in the corpus:

$$p(z_i = k|\vec{z}_{(i)},\vec{w}) = \frac{p(\vec{w},\vec{z})}{p(\vec{w},\vec{z}_{(i)})} = \frac{p(\vec{w}|\vec{z})}{p(\vec{w}_{(i)}|\vec{z}_{(i)})p(w_i)} \cdot \frac{p(\vec{z})}{p(\vec{z}_{(i)})} \tag{24}$$

$$\propto \frac{n_{k,(i)}^{(t)} + \beta_t}{\sum_{t=1}^{V} n_{k,(i)}^{(t)} + \beta_t} \cdot \frac{n_{m,(i)}^{(k)} + \alpha_k}{[\sum_{k=1}^{K} n_m^{(k)} + \alpha_k] - 1}, \tag{25}$$

where $n_{\cdot,(i)}^{(\cdot)}$ indicates that the i^{th} unit in the document is omitted.

The expected value multinomial parameters that correspond to the Markov chain, $\rightarrow z$, can be calculated with:

$$\phi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{t=1}^{V} n_k^{(t)} + \beta_k}, \tag{26}$$

$$\theta_{m,k} = \frac{n_m^{(k)} + \alpha_t}{\sum_{k=1}^{K} n_m^{(k)} + \alpha_k} \tag{27}$$

where $z = k$ and $\vec{n}_k$ is the number of times words have been assigned to topic k. The reader is refered to [6] for the details of the proof.

Example:

Here follows a practical example of how the collapsed Gibbs sampler works. Referring to Table 3, we have a very simple five word document denoted as document 1. The focus of this example will be the re-sampling of the topic assignment to the word "mouse" in document 1. Every word in the document needs to be assigned to a topic. This process has to be repeated for every word in all of the documents in the corpus. As illustrated in Table 3, it is clear that each of the five words have been assigned to one of three different topics. Specifically, the word "mouse" has initially been assigned to topic number two.

Document 1					
Word	Computer	Mouse	Keybord	Sky	Plane
Topic number	1	2	1	3	3
Local counts	Topic 1	Topic 2	Topic 3		
Document 1	2	1	2		
Global statistics	Topic 1	Topic 2	Topic 3		
Computer	49	0	1		
Mouse	10	7	0		
Keybord	50	0	1		
Sky	0	2	38		
Plane	1	0	42		

Table 3: Collapsed Gibbs sampler: Initialisation of random topics to words with local and global statistics

Local counts are then calculated to see how many times a certain topic has been assigned to words the document. The Global statistics is a corpus-wide count of how many times a specific word has been assigned to a given topic. According to the collapsed Gibbs sampler the re-sampling process begins by removing the topic assignment of the word and decrement the local and global counts accordingly. This can be seen in Table 4.Using Eq. 24 the topic assigned to the word "mouse" will be re-sampled. The re-assignment is based on the probability that "mouse" belongs to a topic given every other topic and all of the words in the corpus. Intuitively the re-sampling is based on how much the document "likes" each topic based on the other assignments of topics to words in the document and how much each topic "likes" the word "mouse" based on assignments in other documents in the corpus.

Document 1					
Word	Computer	Mouse	Keybord	Sky	Plane
Topic number	1	0	1	3	3
Local counts	Topic 1	Topic 2	Topic 3		
Document 1	2	0	2		
Global statistics	Topic 1	Topic 2	Topic 3		
Computer	49	0	1		
Mouse	10	6	0		
Keybord	50	0	1		
Sky	0	2	38		
Plane	1	0	42		

Table 4: Collapsed Gibbs sampler: Re-sampling

It is clear from the local counts in Table 4 that document 1 prefers topic 1 and 3 over topic 2. It can also be seen that "mouse" prefers topic 1 over the rest. These two factors are multiplied to get the conditional distribution of Eq. 25. Thus a new topic is assigned as seen in Table 5. The local counts and global statistics should be incremented accordingly.

Document 1					
Word	Computer	Mouse	Keybord	Sky	Plane
Topic number	1	1	1	3	3
Local counts	Topic 1	Topic 2	Topic 3		
Document 1	3	0	2		
Global statistics	Topic 1	Topic 2	Topic 3		
Computer	49	0	1		
Mouse	11	6	0		
Keybord	50	0	1		
Sky	0	2	38		
Plane	1	0	42		

Table 5: Collapsed Gibbs sampler: Incrementing global and local counts

This process is to be repeated until a steady state occurs.

6 Evaluation Method

6.1 Coherence

Choosing the optimal number of topics can be very difficult when applying LDA. One way is to calculate the coherence for different number of topics, k, and then choose k with the highest average coherence. For purposes of this article, the UMass measure will be used to compute coherence of topics. This measurement has proven to be the best when working with LDA [12]. The Umass Coherence measure gives each topic a score by measuring the degree of semantic similarity between high scoring words in the topic. The higher the score, the better the topic. The coherence of a topic is computed as the sum of similarity scores over all the words in the the topic.

$$coherence(V) = \sum_{(v_i, v_j) \in V} score(v_i, v_j, \in) \tag{28}$$

Where V is all the words in the topic and $\in$ is a smoothing parameter to ensure only real values are produced. The UMass metric specifically base scores on the co-occurrence documents:

$$score(v_i, v_j, \in) = log \frac{D(v_i, v_j) + \in}{D(v_j)} \tag{29}$$

Where $D(v_i, v_j)$ is the number of documents containing the i^{th} and j^th word and $D(v_j)$ is the total number of documents with the i^th word.

7 Application

In this section the LDA model will be applied to two text corpora. The programming language used is Python 3.7 and the gensim package is used to implement the LDA model. For illustration prepossess, the first application will be a simple demonstration on a small corpus (corpus 1) where the outcome can be illustrated clearly. LDA is not a model for short text analysis and works best with large corpora, but in this example LDA worked well. The second application will be on a large corpus (post mortem examination reports).

7.1 Datasets

As shown in the corpus 1 below, the documents were fabricated to have obvious topics. The corpus is split into two topics with document A and document E sharing a similar topic and documents B, C and D sharing

a similar topic. This fabrication will make it easy to evaluate the accuracy of the LDA model.

```
Corpus 1:
Document A = "My sister likes racing cars, but not my mother. Racing cars are dangerous."
Document B = "My girlfriend and I love going to the Kruger National Park to see the lions"
Document C = "Lions tend to hunt mostly by night or in the early mornings."
Document D = "My favorite predator is the leopard, but my girlfriend says the lion is the king
 of the jungle."
Document E = "My mother said she will never drive a racing car, because it is too dangerous."
```

The second corpus contains 1669 post mortem examination reports that we received from our colleagues at medical campus. The average number of words per document is 1305 and the total number of words in the corpus sums to 2178045. After cleaning the text 18918 words remained, which indicates that a lot of meaningless words were filtered out and only the most important ones remained.

7.2 Data preprocessing

The post mortem examination reports were received in pdf format and had to be converted into a suitable format that is easy to work with in Python. The first step is to extract the text from each pdf into text documents. The text documents are then converted to a single csv folder where each document is in a single cell. The code for these processes can be found in Appendix B2 and B3 respectively. The next step is to preprocess all of the text. This step is extremely important and if done correctly it can increase the quality of the output significantly. There are three main steps namely tokenizng, stopping and lemmatizing (stemming) of words [13]. Tokenizing is used to convert documents to their atomic level, which in this case are the words. Stopping is then used to remove meaningless words from the corpus which will add no significant value when identifying topics. The next step is lemmatization(stemming) which merges words that are equivalent in meaning. To increase the performance of the model, integers are also removed. Lastly, words that appeared in more that sixty percent of the documents are removed since they will not assist in indicating different topics. All of the code for tokenizing, stopping and stemming are shown in Appendix B4.

7.3 LDA applied to small corpus

The LDA model is first applied to the small corpus (corpus 1) to evaluate the quality of the outcome. After applying the model with two topics specified, the following topics with their top four words were generated and are displayed in Table 6. Words like lion and girlfriend are not normally considered to represent a similar

topic, but when examining the corpus one can see that the topics make perfect sense in this scenario.

Topic 1	Topic 2
racing	lion
car	girlfriend
sister	leopard
mother	favourite

Table 6: Top four words allocated to two topics.

The topics are then allocated to each document with a certain probabilities. It was formerly mentioned in section 5 that when using the LDA model, documents are distributions over topics. As illustrated in Table 7, each document in corpus 1 is assigned to topics with different probabilities. For example document A is assigned to topic 1 and topic 2 with a probability of 94.2 and 0.058 percent respectively and document C is assigned to topic 1 and topic 2 with a probability of 93.4 and 0.066 percent respectively. This is very accurate since document A is clearly more about a topic containing the words racing, car, sister and mother.

	Topic 1	Topic 2
Document A	0.942	0.058
Document B	0.058	0.942
Document C	0.066	0.934
Document D	0.058	0.942
Document E	0.943	0.057

Table 7: Distribution of two topics for five documents.

Documents A and E shared similar topics and were allocated with high probabilities accordingly. Documents B, C and D also shared similar topics and had a similar outcome. From the experiment it can be concluded that the model accurately assigned topic proportions to documents. The code to get these distributions using LDA is given in Appendix B1

7.4 LDA applied to post mortem examination reports

Before applying the LDA model, the number of topics, k, and hyperparameters α and β should be decided on. It was formerly mentioned in section 5 that the β value controls the per topic word distribution and the

α value controls the per document topic distribution. Determining the optimal number of topics, k, in a large corpus is very difficult. We ran the model 30 times, 10 times for 20, 30 and 40 topics each and calculated the average coherence for each of them. According to the coherence score, 30 topics were optimal. This is shown in Figure 4 where it can be seen that 30 topics has the highest coherence. Both the α and β values were kept constant relative to the number of topics at $\frac{1}{k}$, which is the default option in the *gensim* package in Python. The model ran for 100 iterations, which is both time efficient and increases accuracy.

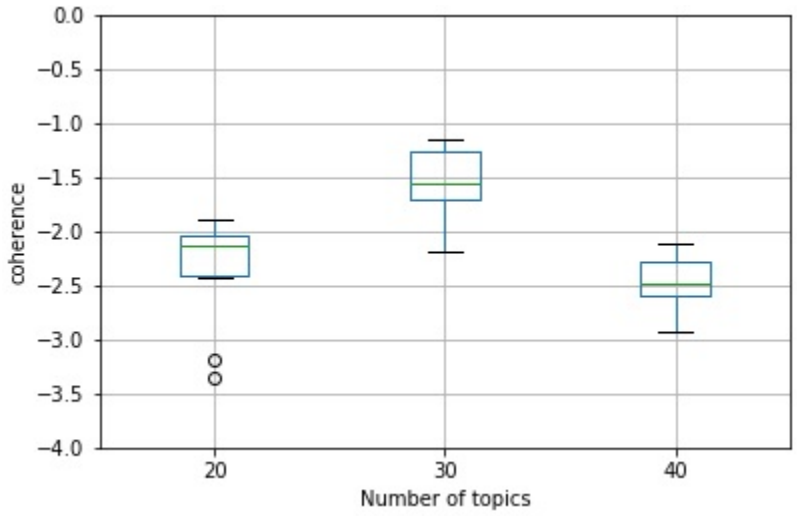


Figure 4: Boxplot grouped by number of topics.

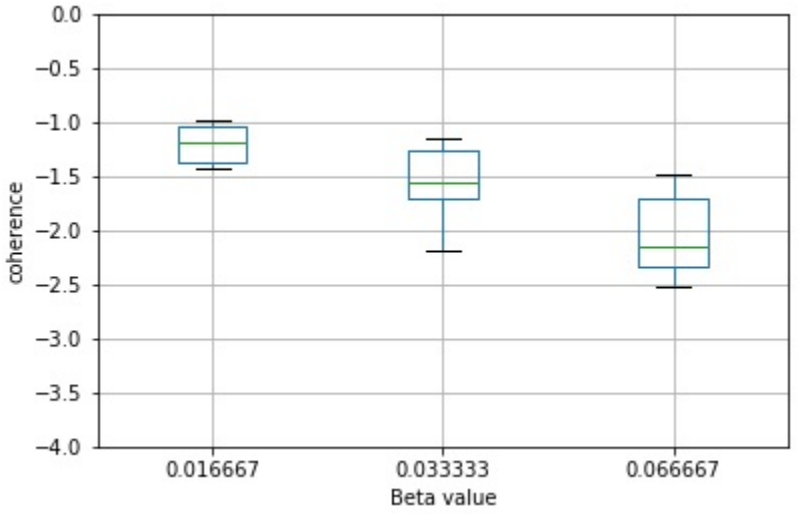


Figure 5: Boxplot grouped by beta value.

Furthermore, to test which the β value is optimal we changed its values to $\beta = \frac{0.5}{k} = 0.016667$ and then to $\beta = \frac{2}{k} = 0.66667$. The number of topics were kept constant at 30. As shown in Figure 5, a β value of

0.016667 is optimal. The α value was kept constant since changing it did not improve the document topic distribution significantly. Table 8 illustrates the first 5 topics inferred from the optimised topic model. The rest can be seen in Appendix C. It is clear that topic 1 describes someone killed by a gunshot an topic 4 describes death caused by burning.

LDA Topics				
Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
gunshot	laceration	stab	burn	foetus
projectile	multiple	incised	heat	length
tract	haematoma	length	pink	umbilical
penetrating	dislocation	penetrating	cherry	placenta
described	visible	orientated	coagulation	congenital
defect	aorta	sharp	charred	crown
entrance	contusion	edge	charring	circumference
irregular	rib	dna	soot	decomposition
exit	side	wound	muscle	matter
intact	fraction	intercostal	degree	conception

Table 8: First 5 topics from post mortem examination reports with top 10 words for each

Furthermore, the prevalence of each topic and how they relate to each other is shown in Figure 6. Topics are represented as circles and their centers are determined by computing the distance between topics and then by using multidimensional scaling to project the inter topic distances onto 2 dimensions [3]. After consulting our colleagues at the forensic science department we concluded that the topic word distributions were very useful and they could immediately recognize the cause of death from most of the topics. Some topics were a bit vague and only made sense once the documents containing a probability of those topics were examined. They found the document topic distributions very useful when allocating documents with a specific topic (cause of death). The only drawback was that some documents had a topic distribution where all the probabilities of topics were very low. Determining a cause of death where a document is allocated to 20 different topics with 5 percent probability each seems very counter productive.

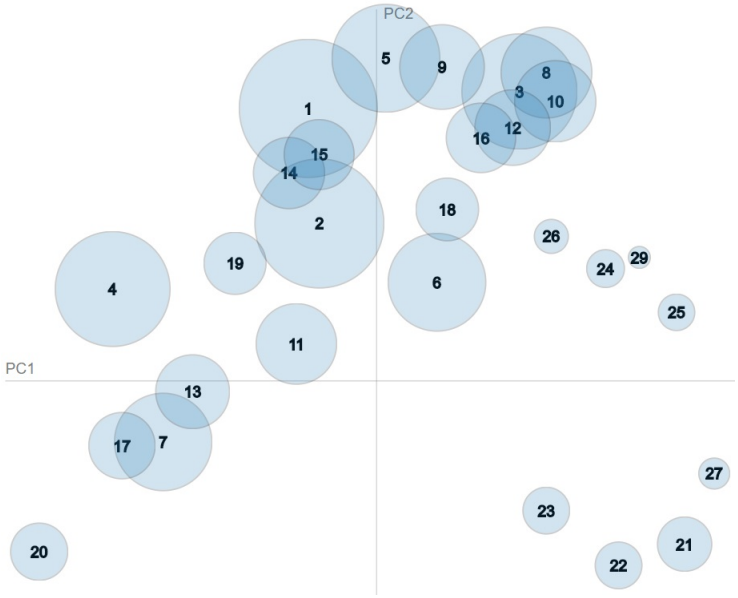


Figure 6: Distance map(via multidimensional scaling)

8 Conclusion

From the experiments above LDA produced positive results for the most part. Topics were given word distributions as expected, and most documents were given topic distributions correctly. However, some document topic distribution were difficult to interpret, since they were given low topic probabilities over a large range of topics. LDA has proven to be a useful tool to organize large corpora and enables us to extract information from unorganized data. However, there is room for improvement with regards to the document topic distributions.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. springer, 2006.

[2] David Blei, Lawrence Carin, and David Dunson. Probabilistic topic models. *IEEE Signal Processing Magazine*, 27(6):55–65, 2010.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

[4] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391, 1990.

[5] Persi Diaconis and Donald Ylvisaker. Conjugate priors for exponential families. *The Annals of statistics*, 7(2):269–281, 1979.

[6] Heinrich Gregor. Parameter estimation for text analysis. *Technical report*, 2005.

[7] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[8] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[9] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2):103–134, 2000.

[10] Juan Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[11] Gerard Salton and Michael J McGill. *Introduction to Modern Information Retrieval. Mcgraw-Hill*, 1986.

[12] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 952–961. Association for Computational Linguistics, 2012.

[13] S Vijayarani, Ms J Ilamathi, and Ms Nithya. Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, 5(1):7–16, 2015.

Appendix A

The SAS program for implementing the Gibbs sampler in estimating a mean value of 30 for a normal distribution.

```
data gibbs;
n = 10;
true_mu = 30;
do
  i = 0;
  mu = 0;
  sg2 = 1;
  output;
end;
seed1 = 1;
do i = 1 to 2500;
  retain seed1;
  call rannor(seed1,r1);
  mu = true_mu+sqrt(sg2/n)*r1;
   output;
end;
run;
data gibbs;
set gibbs;
if i <= 500;
keep i mu;
proc univariate noprint;
var mu ;
output out = AA mean=mean_mu
run;
proc print;
run;
```

Appendix B

Python version 3.6 is the only programming language used in this section.

B1. The LDA model applied to a small corpus of five documents.

```
import gensim
from nltk.tokenize import RegexpTokenizer
from stop_words import get_stop_words
from gensim import corpora, models
import numpy as np
from nltk.stem.wordnet import WordNetLemmatizer
from pprint import pprint
from gensim.models import LdaModel


tokenizer = RegexpTokenizer(r'\w+')
en_stop = get_stop_words('en')
lemmatizer = WordNetLemmatizer()

doc_a = "My sister likes racing cars, but not my mother. Racing cars are dangerous."
doc_b = "My girlfriend and I love going to the Kruger national park to see the lions"
doc_c = "Lions tend to hunt mostly by night or in the early mornings."
doc_d = "My favourite predator is the leopard, but my girlfriend says the lion is the king
 of the jungle. "
doc_e = "My mother said she will never drive a racing car, because it is too dengerous."



doc_setworking = [doc_a, doc_b, doc_c, doc_d, doc_e]
#print(doc_setworking)
print(len(doc_setworking))
#print (doc_set)

texts = []
```

```
for i in doc_setworking:
    raw = i.lower()
    print(raw)
    tokens = tokenizer.tokenize(raw)
    #print (tokens)
    stopped_tokens = [i for i in tokens if not i in en_stop]
    #print (stopped_tokens)
    docs = [lemmatizer.lemmatize(i) for i in stopped_tokens]
    #print (docs)
    texts.append(docs)

dictionary = corpora.Dictionary(texts)
print (dictionary)
#print (dictionary.token2id)
corpus = [dictionary.doc2bow(text) for text in texts]

num_topics = 2
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics=num_topics,
 id2word = dictionary, passes=20)
print (ldamodel.print_topics(num_topics=2, num_words=4))

color = []
for corpus_line in corpus[:5]:
    sorted_yopic_line = list(sorted(ldamodel[corpus_line], key=lambda x:x [1], reverse=True))
    color.append(sorted_yopic_line[0][0])
lda_output = []
for line in corpus[:5]:
    lda_output.append(ldamodel[line])
topics_data = np.zeros(shape=(5,2))
for i, line in enumerate(lda_output):
    for topic_line in line:
        topics_data[i][topic_line[0]] = topic_line[1]
```

```
print(topics_data[0])
```

B2: pdf to text

```
import os
from os import chdir, getcwd, listdir, path
import PyPDF2
from time import strftime
def check_path(prompt) :
        abs_path = input(prompt)
        while path.exists(abs_path) != True:
            print ("\nThe specified path does not exist.\n")
            abs_path = input(prompt)
        return abs_path
print ("\n")
folder = check_path("Provide absolute path for the folder: ")
list=[]
directory=folder
for root,dirs,files in os.walk(directory):
    for filename in files:
        if filename.endswith('.pdf'):
            t=os.path.join(directory,filename)
            list.append(t)
for item in list:
    path=item
    head,tail=os.path.split(path)
    var="\\"
    tail=tail.replace(".pdf",".csv")
    name=head+var+tail
    content = ""
    pdf = PyPDF2.PdfFileReader(path, "rb")
    for i in range(0, pdf.getNumPages()):
        content += pdf.getPage(i).extractText() + "\n"
```

```
    print (strftime("%H:%M:%S"), " pdf  -> csv ")
    with open(name,'a') as out:
        out.write(content)
```

B3: All text files to single .csv file.

```
import glob
import csv
read= glob.glob('f2d\\*.txt')
with open("neg2.csv", "w") as outfile:
    w=csv.writer(outfile)
    for f in read:
        with open(f, "r") as infile:
            w.writerow([" ".join([line.strip() for line in infile])])
```

B4: Lda model on post mortem examination reports.

```
import gensim
import pandas as pd
from scipy import stats
from gensim import corpora, models
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.tokenize import RegexpTokenizer
from nltk.stem.wordnet import WordNetLemmatizer
from stop_words import get_stop_words
import pyLDAvis
import pyLDAvis.gensim
import numpy as np
import csv
import json
import nltk
import re
from pprint import pprint
```

```
from gensim.models import LdaModel
from gensim.models.coherencemodel import CoherenceModel

s = set(stopwords.words('english'))
s.update(['g','cm','e','gw','republic','south','africa','post','mortem','examination'
,'19 cm','also','5cm','2nd','paragraph'])
tokenizer = RegexpTokenizer(r'\w+')
en_stop = get_stop_words  ('en')
lemmatizer = WordNetLemmatizer()
#print (s)
#print (en_stop)

data_file = pd.read_csv('Final_data.csv',encoding='cp1252')
#data_file.head
#print (data_file[0:3])
#print (data_file)
print (len(data_file))

doc_set = data_file.reset_index().astype(str).values.tolist()
for x in doc_set:
    del x[0]
#print(doc_set)
#print (len(doc_set))
docset = []
for sublist in doc_set:
    for val in sublist:
        docset.append(val)
#print (docset)
print (len(docset)
print (len(docset))
num_words = [len(sentence.split()) for sentence in docset]
#print (num_words)
```

```
print (sum(num_words))
ave_len = sum(num_words) / float(len(num_words))
print (ave_len))

texts = []
for i in docset:
    raw = i.lower()
    tokens = tokenizer.tokenize(raw)
    #print (tokens)
    stopped_tokens_1 = [i for i in tokens if not i in s]
    stopped_tokens_2 = [i for i in stopped_tokens_1 if not i in en_stop]
    no_integers = [x for x in stopped_tokens_2 if not (x.isdigit()
                                        or x[0] == '-' and x[1:].isdigit())]
    docs_1 = [i for i in no_integers if len(i) > 1]
    #print (stopped_tokens_1)
    #print (stopped_tokens_2)
    #print (no_integers)
    docs_2 = [lemmatizer.lemmatize(i) for i in docs_1]
    texts.append(docs_2)

dictionary = corpora.Dictionary(texts)
dictionary.filter_extremes(no_below=0, no_above=0.6, keep_n=100000, keep_tokens=None)
dictionary.save('dictionary.dict')
print (dictionary)
#print (dictionary.token2id)

corpus = [dictionary.doc2bow(text) for text in texts]
corpora.MmCorpus.serialize('corpus.mm', corpus)
#print (corpus)
print (len(corpus))

num_topics = 30
```

```
chunksize=5000
passes = 20
iterations = 100
eval_every = None
eta = 0.5/30

ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics=num_topics,
update_every=1, chunksize=chunksize,
id2word = dictionary, passes=passes, eval_every=eval_every,iterations = iterations,eta =eta)
print (ldamodel.print_topics(num_topics=50, num_words=10))

goodcm = CoherenceModel(model=ldamodel, corpus=corpus, dictionary=dictionary,
 coherence='u_mass')
print (goodcm)
print (goodcm.get_coherence())

for i in ldamodel.print_topics():
    for j in i: print(j)

ldamodel.save('topic.model')
loading = LdaModel.load('topic.model')
print(loading.print_topics(num_topics=2, num_words=4))



d = gensim.corpora.Dictionary.load('dictionary.dict')
c = gensim.corpora.MmCorpus('corpus.mm')
lda = gensim.models.LdaModel.load('topic.model')

data = pyLDAvis.show(pyLDAvis.gensim.prepare(lda, c, d))
data

duration = 500  # millisecond
```

```
freq = 440  # Hz
winsound.Beep(freq, duration)

color = []
for corpus_line in corpus[:10000]:
    sorted_yopic_line = list(sorted(ldamodel[corpus_line], key=lambda x:x [1], reverse=True))
    color.append(sorted_yopic_line[0][0])

lda_output = []
for line in corpus[:10000]:
    lda_output.append(ldamodel[line])

topics_data = np.zeros(shape=(10000,50))

for i, line in enumerate(lda_output):
    for topic_line in line:
        topics_data[i][topic_line[0]] = topic_line[1]
print(topics_data[300])
```

B5: Plotting of a box and whiskers for coherence.

```
import matplotlib
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data =  pd.read_csv('coherence.csv', sep=',', na_values='.')
data.head()
slice = data.iloc[:,[0, 1]]
bp = slice.boxplot(column='Coherence', by ='Number of topics')
axes = plt.gca()
axes.set_ylim([-4,0])
plt.ylabel('coherence')
plt.show
```

Appendix C

Topic word distributions for topic 6 to 30.

LDA Topics				
Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
congested	oedematous	ventricle	burn	involving
abrasion	urine	measure	heat	abrasion
white	opened	aortic	pink	focal
friction	white	valve	cherry	intact
congestion	visible	aorta	coagulation	laceration
petechial	histology	pulmonary	charred	confluent
contusion	oedema	graft	charring	contusion
ligature	official	circumference	soot	side
muscle	congestion	catheter	muscle	glistening
thyroid	congested	sutured	degree	wound

LDA Topics				
Topic 11	Topic 12	Topic 13	Topic 14	Topic 15
well	ventricle	jaundiced	foetus	intact
within	measure	jaundice	umbilical	upon
side	aortic	fresh	length	appears
observed	valve	buttock	female	congested
opened	aorta	surgery	placenta	autolytic
focal	pulmonary	exited	staining	moderately
soul	graft	quadrant	putrefactive	appear
uct	circumference	copper	congenital	smooth
bathabile	catheter	umbilicus	circumference	loot
intervention	sutured	saddle	congested	white

LDA Topics				
Topic 16	Topic1 17	Topic 18	Topic 19	Topic 20
autolysis	multiple	ligature	laceration	burn
decomposition	abrasion	congestion	contusion	heat
disease	laceration	vascular	multiple	pink
severe	intact	around	associated	cherry
poor	upon	cartilage	haematoma	coagulation
keeping	rib	thyroid	side	charred
focal	irregular	abrasion	opened	charring
chronic	region	petechial	lobe	soot
acute	extensive	dark	official	muscle
section	bilateral	brown	abrasion	degree

LDA Topics				
Topic 21	Topic 22	Topic 23	Topic 24	Topic 25
congested	remains	wound	wound	length
abrasion	natural	intact	gunshot	foetus
white	bone	region	defect	natural
friction	comment	upon	projectile	product
congestion	skeletal	situ	mentioned	conception
petechial	mm	tube	tract	officer
contusion	information	catheter	entrance	viable
ligature	molar	surgical	shaped	gestational
muscle	length	appears	exit	week
thyroid	skin	multiple	abrasion	non

LDA Topics				
Topic 26	Topic 27	Topic 28	Topic 29	Topic 30
wound	oedematous	ventricle	muscle	unremarkable
gunshot	urine	measure	heat	congested
tract	opened	aortic	sutured	skeletal
penetrating	white	valve	cherry	seen
described	visible	aorta	coagulation	white
defect	histology	pulmonary	charred	dissection
entrance	oedema	graft	charring	histology
irregular	official	circumference	soot	mabotja
exit	congestion	catheter	burn	matter
intact	congested	sutured	degree	routine

On the modelling of stock prices using Lévy processes

Nqaba Samkelo Duma 14378249

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr IJH Visagie

Department of Statistics, University of Pretoria

30 October 2017 (Final)

Abstract

In this study, we investigate the properties of Lévy processes and the ability of these processes to accurately model the behaviour of stock prices. A discussion of Lévy processes is presented and the method of maximum likelihood estimation is used to fit geometric Lévy process models to observed financial data. This is accompanied by an analysis of the characteristics of observed financial time series data. The aim of the research is to consider whether or not a geometric Lévy process model can be used to accurately model the behaviour of stock prices over time.

Declaration

I, *Nqaba Samkelo Duma,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Nqaba Samkelo Duma

_ _

Dr Jaco Visagie

_ _

Date

Acknowledgements

I acknowledge the financial support from the National Treasury, under the Ministry Of Finance. I would also like to thank my supervisor Dr Jaco Visagie, for his patient guidance, encouragement and the advice he has provided throughout my time as his student.

Contents

List of Figures

1 Introduction

Financial modelling dates back to the 1900s, when a French mathematician named Louis Bacheliers defended his thesis where he used a Brownian motion with a drift, to model stock prices, see [12]. The aim of financial modelling is to find a relatively simple model that captures the most important properties of observed financial data. Our task is to find a realistic model for the prices of stocks.

The Black-Scholes model was the first widely accepted financial model, it provided an explanation of the time evolution of a stock price under the assumption that log-returns follow a Brownian motion. In this research, we highlight that the Black-Scholes model fails to capture all the characteristics of observed log-returns, and we consider more flexible models based on Lévy processes. These processes are named after Paul Lévy, a French mathematician, who pioneered these processes, see [24].

The remainder of this report is structured as follows; Section 2 describes financial markets and empirical properties of log-returns, the so called stylized facts. We also define a stock price process in Section 2. Section 3 describes financial modelling. This section provides the formal definition of a Brownian motion. Then we discuss the Black-Scholes model and its imperfections, chief among these is the assumption that log-returns are normally distributed, see [24]. This assumption turns out to be unrealistic especially when we consider the skewness and heavy tails of observed log-returns, see [9]. In Section 4, Lévy processes are defined and a discussion of their most important properties is provided. We focus on three examples of Lévy process; a Brownian motion, the normal inverse Gaussian process and the Meixner process, see [24]. We then turn our attention to the estimation of parameters of the Lévy processes discussed. In Section 5, we discuss the method of maximum likelihood estimation, which is the method used to estimate the parameters of the various models considered. We fit the proposed models to observed financial data in Section 6. Then, Section 7 presents the conclusions of the study.

2 Financial markets and the empirical properties of log-returns

The term "financial market", is a broad term defining a market place where traders buy and sell financial assets. These assets include stocks, financial securities, bonds, and options to only mention a few. In this research, we focus mainly on the modeling of stock prices. The stock market, dates back to 1531, see [24]. Stocks, also known as shares, are financial instruments that provide partial ownership of a listed company to the holder. Upon listing on the securities exchange market, companies raise financial capital by selling stocks. A stock represents fractional ownership with limited liability in a company, and its value fluctuates on a day to day basis in response to market buy and sell dynamics, see [14]. The need to accurately model the prices of these stocks arise from the large amounts of money invested in these assets globally. In this section we discuss some statistical properties of log-returns.

The discussion contains examples from realized data from the Standard and Poor 500 index, abbreviated as S&P500. The data used was recorded on a daily basis and dates from 2010/09/06 to 2017/09/01. The S&P 500 is a stock market index that tracks 500 American companies that represent more than seventy percent of the total market capitalization. It is a capitalization weighted index that tracks the average movement of the stock market. The data consists of 365 daily log-returns downloaded from Yahoo Finance, which can be accessed from https://finance.yahoo.com and Figure 1 shows the evolution of the daily stock price of the S&P500 index. The graph shows that stock prices increase exponentially as time increases, hence the need of an exponential process for modeling them.

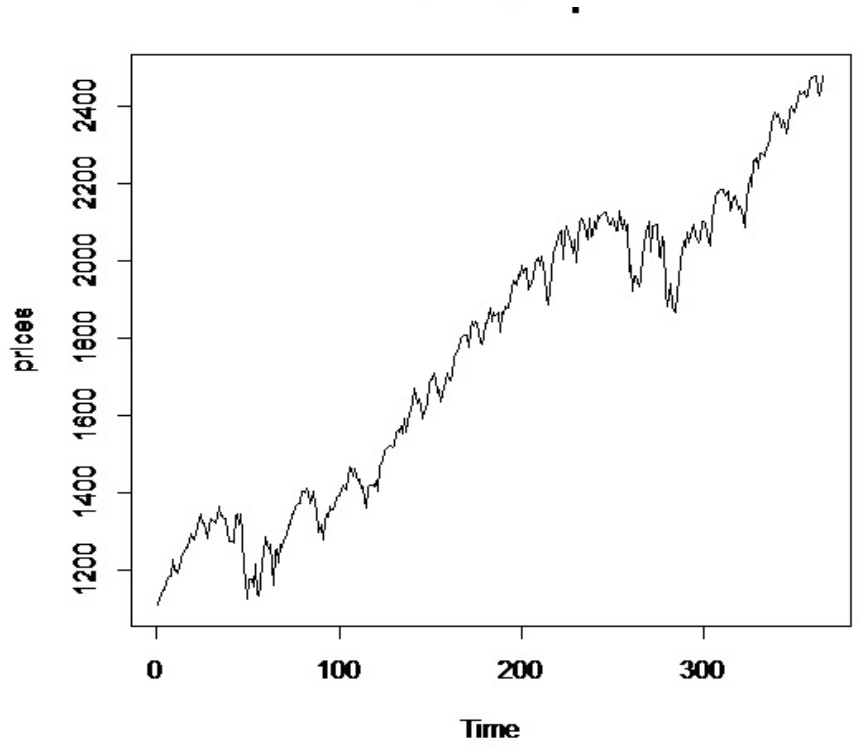


Figure 1: Time series of daily stock prices showing an irregular fluctuating pattern

When modeling financial markets one typically does not model the stock prices directly, one rather models the log-returns of the stock prices. That is, one typically models the stock price at time t as $S_t = S_0 \exp(X_t)$, where X_t is a stochastic process referred to as the log-return process of the stock. An analysis of log-returns is often are preferred to an analysis of actual prices because log-returns provide a scale free examination of the performance of the asset, see [20]. Figure 2 shows the time series analysis of daily log-returns of the S&P 500 index. It shows that log-returns are stationary and fluctuate around a constant average value.

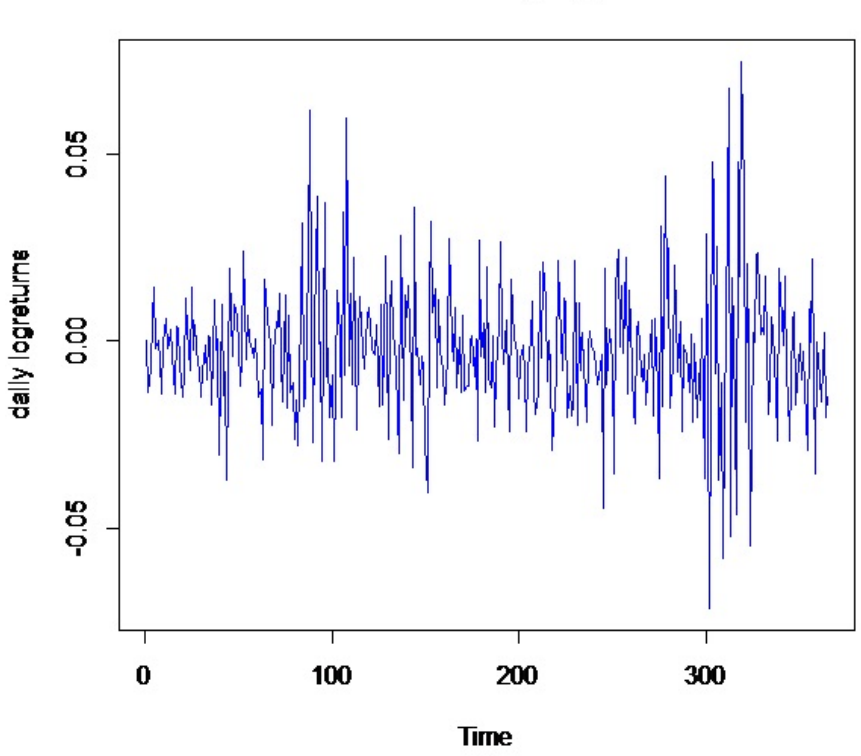


Figure 2: Log-returns of S&P500 index.

We denote the stock price process by $S = \{S_t; t \geq 0\}$, where S_t is the price of the stock at time $t \geq 0$. Our geometric model for the stock price is

$$S_t = S_{t-1} \exp(X_t),$$

where X_t is the log-return process. X_t can be written in terms of the stock price as follows:

$$X_t = \log\left(\frac{S_t}{S_{t-1}}\right)$$

We assume that X_t follows some Lévy process throughout. As a result, it is assumed that the increments of X_t are independent and identically distributed random variables.

In what follows, we present a detailed discussion of the empirical properties of log-returns of the S&P500 weekly prices. The properties discussed are referred to as stylized facts, see [9]. These properties are common across all financial markets, hence the need to take them into account when developing a model for financial data. The mentioned properties are discussed in turn below.

2.1 Absence of auto-correlation

Auto-correlations of financial asset returns are close to zero, except in cases where the process is observed for small time intervals, for instance a minute, see [9]. The time series analysis of log-returns

in Figure 2 often show no significant auto-correlations. As a result, log-returns are often assumed to be independent and identically distributed random variables and the proposed geometric Lévy process model does not violate this assumption, see [4]. This assumption makes our modeling task less complex. If auto-correlations were present in the market, then traders could take advantage of this and exploit this property in order to predict returns. As a result, investors could then take advantage of linear auto-correlations in the returns to construct strategies for making profits based on trends, see [18]. Figure 3 shows the auto-correlation plot of log-returns. From the plot, we see that the auto-correlation quickly drops to zero as the lag increases.

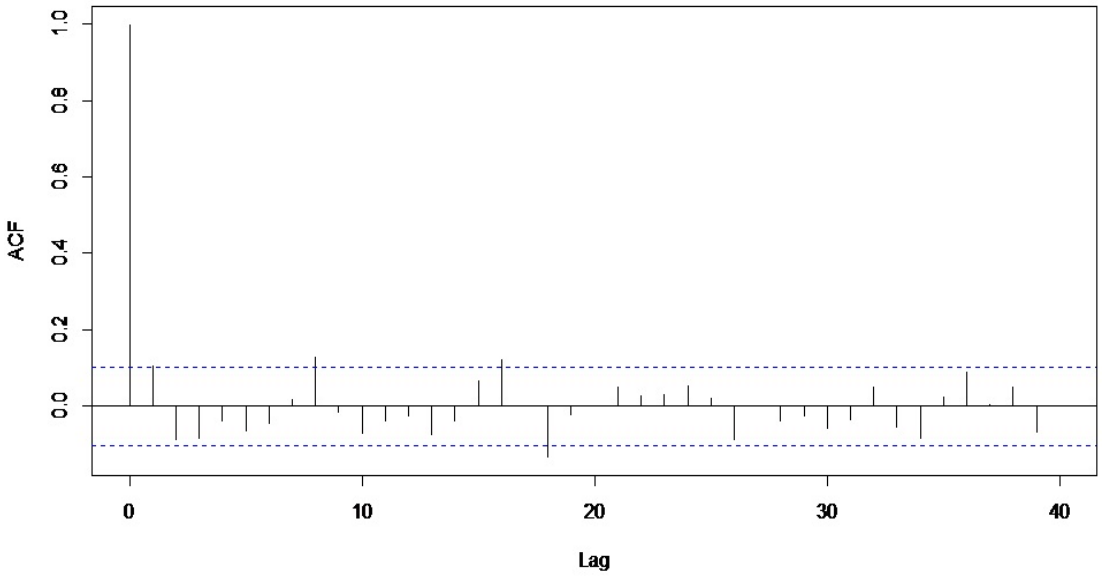


Figure 3: Auto-correlation plot of log-returns

2.2 Heavy-tailed distribution

The distribution of observed log-returns exhibit heavy tails, that is, there are more observed data points in the tails or the extremes of the distribution than one would expect under the assumption of normality, see [9]. These events have a high impact on the stock price, hence the need to capture the behaviour of extreme events. The heaviness of the tails of a distribution are measure by the kurtosis which is defined as the fourth standardized central moment:

$$Kur(X) = \frac{E\left[(X - E[X])^4\right]}{var\left[X\right]^2}.$$

$Kur(X) = 3$ for a normal distribution. A kurtosis of greater (smaller) than 3 indicates heavier (lighter) tails than those associated with the normal law.

Empirical results suggest that log-returns have a kurtosis greater than 3, which implies that log-returns have a heavy-tailed distribution. The calculated sample kurtosis of log-returns of S&P500 weekly prices is equal to 5.246 which is in line with the findings of [9] and [12].

2.3 Symmetry of the distribution

Empirical evidence suggests that the distribution of log-returns is skewed to the left. This suggest that positive log-returns are more common than negative log-returns. However, negative log-returns tend to be larger in magnitude than positive log-returns. The degree of asymmetry of a random variable is measured by the skewness of the distribution. Skewness is defined as the third standardized central moment:

$$Skew(X) = \frac{E\left[(X - E[X])^3\right]}{var\,[X]^{\frac{3}{2}}}.$$

For a symmetric distribution $Skew(X) = 0$. The normal distribution $N(\mu, \sigma^2)$, is symmetric.

An analysis of the observed S&P500 data, concludes that log-returns tend to be negatively skewed. The calculated sample skewness of log-returns of the S&P500 data is 0.395. As a result, the Brownian motion with its marginal normal distributions, is not ideally suited to model log-returns.

2.4 Aggregational Gaussianity

Aggregational Gaussianity means that long term aggregation of financial asset returns, that is considering the returns over longer periods will lead to approximately normally distributed log-returns while observed log-returns on a small time scale do not follow a normal distribution, see [5]. As the time scale over which log-returns are calculated increases, the distribution tends to be closer to normality. This implies that the shape of the distribution is not identical at different time scales, see [9]. Figure 4 shows a kernel density estimates of log-returns calculated on different time scales. The normal density is super imposed on the figures. The solid line represents the normal density function and the dashed line represents the kernel density estimator the of log-returns. Figure 4 shows a kernel density estimate of daily log-returns, a kernel density estimate of weekly log-returns and a kernel density estimate of monthly log-returns. It is seen in the change in the fit of the imposed normal density that as the time scale over which log-returns are calculated increases, the distribution tends to be closer to normality.

2.5 Volatility clustering

Volatility measures the spread of the distribution of log-returns. The variance of log-returns is often taken to be the volatility of the stock, and used as a proxy for the risk attached to investments. There

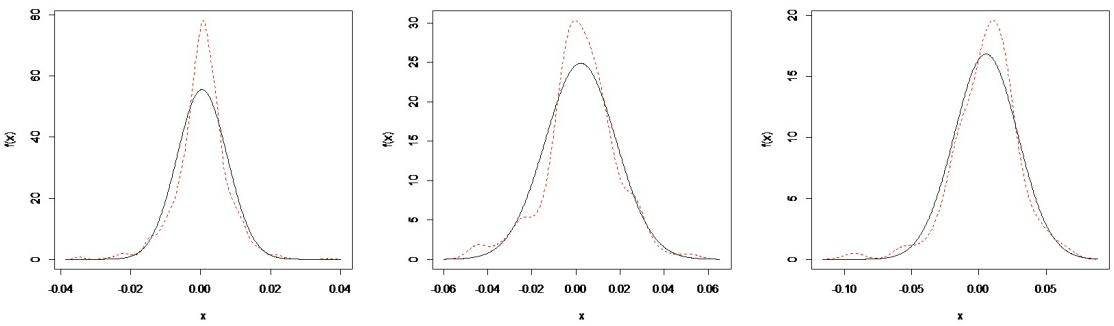


Figure 4: Density estimates on varying time scales.

are two types of volatility, namely historical volatility and implied volatility. Historical volatility is a measure of volatility that calculates price changes over a predefined period of time. Implied volatility is an expectation of volatility over a certain period implied by the prices of financial derivatives such as options. For the purposes of this study historical volatility plays an important role, while implied volatility is simply mentioned for the sake of completeness.

The volatility clustering property of asset returns implies that large fluctuations in prices are usually followed by large fluctuations, similarly with small fluctuations, see [10]. For instance, if an event that increases the price of a company's stock occurs at time t, say the company merged with a bigger company, then volatility clustering implies that there will likely be a large change in price at time $t + 1$. High volatility events tend to cluster together, see [9].

Most measures of volatility of stocks are negatively correlated with log-returns of that stock, leading to the concept of leverage effect. That is, volatility increases when the stock price decreases. The leverage effect refers to the relationship between stock returns and volatility, see [15]. Furthermore, the volume of trade in the market is also correlated with volatility. Empirical evidence of volatility clustering is shown in Figure 2 using S&P500 daily returns. Larger absolute log-returns tend to cluster together and smaller absolute log-returns likewise.

3 Financial modelling

The goal of financial modelling is to accurately model, forecast or predict future asset prices. An ideal financial model should capture all of the most prominent features of the relevant financial market without being over complicated. We assume that the log-return processes X_t follows three different Lévy processes, in what follows; a Brownian motion, a normal inverse Gaussian process and a Meixner process. We start by defining the Black-Scholes model and its imperfections and then turn our attention to Lévy processes.

3.1 The Black-Scholes model

The Black-Scholes model, proposed in, see [12], was the first widely accepted model for stock prices. Under the Black-Scholes model, the log-return process X_t is assumed to follow a Brownian motion.

Definition 1. The Black-Scholes model for stock prices

Consider the evolution of the stock price S in a small time interval $[t, t+\Delta t]$. Denote the change in the stock price $S_{t+\Delta t} - S_t$ by ΔS_t. Under the Black-Scholes model the dynamics of the price process are given by

$$\Delta S_t = \mu S_t \Delta t + \sigma S_t \Delta W_t \tag{1}$$

The stochastic differential in (1) contains two parts, a systematic and random part. The systematic part is given by $\mu S_t \Delta t$, where $\mu \Delta t$ represents the mean rate of return, see [24]. It is assumed that the expected rate of return is proportional to the length of the interval, Δt. The random part is captured by, $\sigma S_t \Delta W_t$ where $\sigma \geq 0$ is the volatility parameter and W_t is a standard Brownian motion.

The solution of (1) is given by

$$S_t = S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W_t\right) \tag{2}$$

The geometric Brownian motion stock price model in (2), is known as the Black-Scholes model.Under this model, log-returns are modeled by a Brownian motion with a drift of $\mu - \frac{1}{2}\sigma^2$. Under the Black-Scholes model, the log-return process is

$$X_t = \log\left(\frac{S_t}{S_0}\right) = \left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W_t \tag{3}$$

3.2 Imperfections of the Black-Scholes model

Below we provide an account of some of the imperfections of the Black-Scholes model. The arguments presented below are based on empirical properties of asset returns. We will focus mainly on the assumption that the distribution of log-returns is inconsistent with the normal law. Since under the Black-Scholes, log-returns are modeled using a Brownian motion with a drift, but empirical evidence has suggested otherwise. In [9], a detailed study of the properties of asset returns is provided.

[9] concludes that asset returns do not follow the normal law. Empirical evidence suggests that the distribution of the log-returns are heavy-tailed and negatively skewed. In this study, we model the stock price process of the standards and poor 500 (S&P 500) data using the three different Lévy process mentioned above. Our analysis of the S&P500 data in common with the findings of [9], suggest that the assumption of normality is unrealistic. To test for normality we use SAS. Figure 5 shows the histogram of log-returns with an imposed normal density. This suggests that the normal density density does not

fit well.

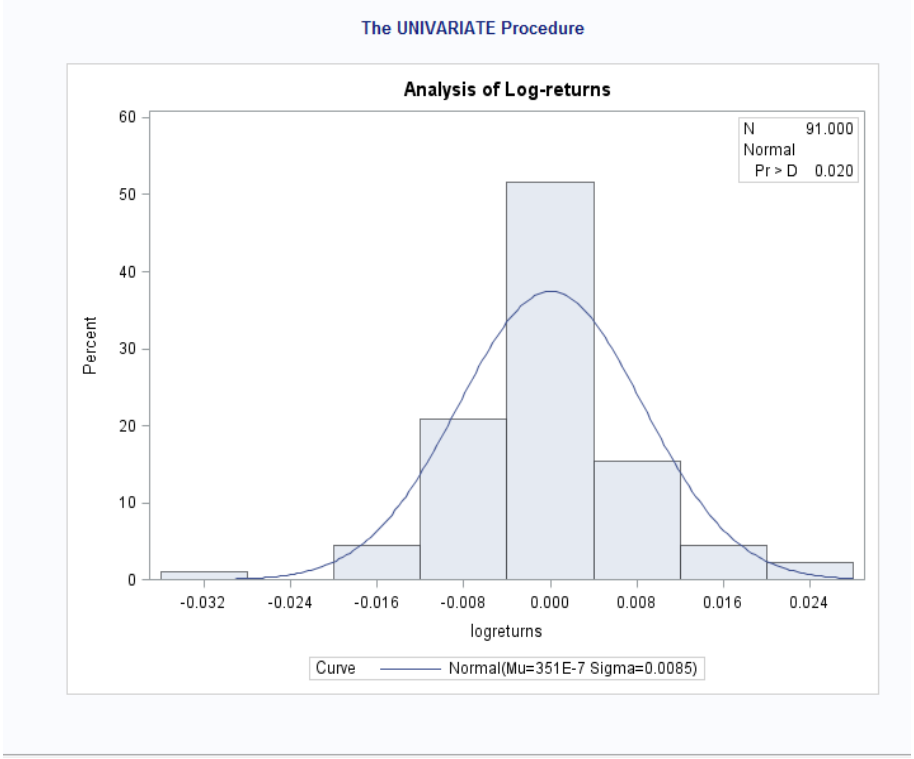


Figure 5: Normal density imposed on histogram of log-returns

Figure 6 shows output from SAS for testing for normality. The p-values for all the tests are all less than 0.05 which implies that we would reject the null hypothesis about the normality assumption at a 5% level of significance. The Kolmogorov-Smirnov test has a p-value of 0.02. The Cramer-von Mises test has a p-valu of 0.005 and the Anderson-Darling test has a p-value of 0.006. It is clear that log-returns are not normally distributed as we would also reject the null hypothesis even at a 10% significance level.

In the next section we consider the Brownian motion, normal inverse Gaussian and the Meixner distributions which have properties that are suitable to capture the properties displayed by the log-returns of stock prices.

4 Lévy processes

Lévy processes were first used in financial econometrics in [18], when Mandelbort proposed α-stable Lévy processes for modeling cotton prices. In this section, we start by defining two important concepts that form an integral part of Lévy processes, characteristic functions and infinitely divisible distributions.

Goodness-of-Fit Tests for Normal Distribution				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.4501658	Pr > D	<0.010
Cramer-von Mises	W-Sq	6.4590870	Pr > W-Sq	<0.005
Anderson-Darling	A-Sq	31.4080843	Pr > A-Sq	<0.005

Figure 6: SAS output

Then a formal definition of Lévy processes follows together with a discussion of the properties of these processes. As specific examples of Lévy process, we consider a Brownian motion, the normal inverse Gaussian process and the Meixner process.

Consider the definition of a characteristic function.

Definition 2. Characteristic function

A characteristic function, ψ of a random variable X, is the Fourier-Stieltjes transform of the cumulative distribution function $F(x) = P(X \leq x)$ such that

$$\psi_X(u) = E\left[\exp(iuX)\right] = \int_{-\infty}^{\infty} \exp(iux) dF(x)$$

Note that $\psi(0) = 1 \in |\psi(u)| \leq 1 \; for \; all \; u \in \mathbb{R}$. ψ uniquely determines the distribution function F. Furthermore, the characteristic function uniquely determines the distribution of a random variable and is always exists and continuous.

Proposition 3. ***The characteristic function of a Lévy process***

Suppose that $\{X_t; t \geq 0\}$ *is a Lévy process. Then there exists a continuous function* $\psi : \mathbb{R} \to \mathbb{R}$ *called the characteristic exponent of such that*

$$E\left[\exp\left(iz.X_t\right)\right] = \exp\left(t\psi\left(z\right)\right)$$

where $z \in \mathbb{R}$. *This defines the characteristic function of a Lévy process.*

Definition 4. An infinitely divisible distribution

A cumulative distribution function F, is said to be infinitely divisible if for any $n \in \mathbb{Z}, n \geq 2$ *there exists* n *i.i.d* random variables $Y_1, Y_2, ..., Y_n$ such that $Y_1 + ... + Y_n$ has the law$F$, see [12]. The marginal distribution of every Lévy process possesses the property of infinite divisibility.

Now that we have defined an infinitely divisible distribution, we can define a Lévy process. The definition below is taken from, see [12].

Definition 5. Lévy process

A stochastic process $X = \{X_t; t \geq 0\}$ defined on a filtered probability space $(\Omega, \mathcal{F}, \mathcal{F}_t, P)$ is called a Lévy process if it satisfies the following properties:

1. $X_0 = 0$.
2. X_t possesses independent increments; for every $0 \leq t_0 \leq t_1 \leq ..... \leq t_n$, the random variables $0 \leq X_{t_0}, X_{t_1} - X_{t_0}, ..., X_{t_n} - X_{t_{n-1}}$ are independent.
3. X_t has stationary increments; the distribution of $X_{t+k} - X_t$ only depends on the length of the interval, k.
4. X_t is stochastically continuous; $\lim_{k \to 0} P(|X_{t+k} - X_t| \geq \epsilon) = 0 \, \forall, \epsilon \geq 0$.

Remark 6. Note that properties (2) and (3) imply that a Lévy process is a Markov process, see [2]. They are the only Markov processes which are homogeneous in both space and time[12]. Lévy processes are homogeneous because the distribution of their increments only depend on the length of the interval as stated by property (2) .

Proposition 7. ***Infinite divisibility and Lévy processes***

If X is a Lévy process, then X_t is infinitely divisible for each $t \geq 0$, see [2].

Proof. For each $n \in \mathbb{N}$, we can write

$$X(t) = Y_1^n(t) + ... + Y_n^n(t)$$

where each

$$Y_k^n = X\left(\frac{kt}{n}\right) - X\left(\frac{(k-1)\,t}{n}\right)$$

The $Y_k^n(t)$ are independent and identically distributed, by (2) and (3) in definition 6. □

Definition 8. The Lévy measure

The Lévy measure is defined as follows, see [9]. Suppose that $\{X_t; t \geq 0\}$ is a Lévy process on $\mathbb{R}$. The measure ν on $\mathbb{R}$, defined by:

$$\nu(A) = E\left[number\, of\, jumps\, \{t \in [0,1] : \Delta X_t \neq 0, \, \Delta X_t \in A\}\right], \;\; A \in B(\mathbb{R})$$

is called a Lévy measure of X and $\nu(A)$ is the expected number of jumps per unit time, whose size belong to A. In Section 4.1 we define the Lévy-Khinchine formula with the Lévy triplet, which has the Lévy measure as one of its components.

4.1 The Lévy-Khintchine formula and the triplet of Lévy characteristics

The Lévy-Khintchine formula shows the relationship between infinite divisible distributions and stochastic processes with independent and stationary increments[6]. This formula provides a way of decomposing a Lévy process into three parts; a straight line component, a Brownian motion and a pure jump process. The characteristic exponent of all Lévy processes satisfies the Lévy-Khintchine formula:

$$\log\left(E^p\left[\exp\left(iuX_t\right)\right]\right) = t\left\{iu\gamma - \frac{\sigma^2u^2}{2} + \int_{-\infty}^{\infty}\left(e^{iux} - 1 - iuh(x)\right)\nu(dx)\right\},$$

where $\gamma\, in\, \mathbb{R}$, $\sigma^2 \geq 0$, $h(x)$ is some truncation function and ν is a measure on $\mathbb{R}\backslash\{0\}$ such that

$$\int_{-\infty}^{\infty}\{1, x\}\,\nu(dx) < \infty.$$

$\left(\gamma, \sigma^2, \nu(dx)\right)$ is called the triplet of Lévy characteristics. In the decomposition of a Lévy process, γ gives the slope of the straight line, σ^2 is the variance of the Brownian motion and $\nu(dx)$ governs the jumps made by the process as defined in the previous subsection.

Below we consider specific Lévy processes. We consider a Brownian motion, the normal inverse Gaussian process and the Meixner process.

4.2 Brownian motion

The Brownian motion is the only continuous example of a Lévy process with no jumps.The definition of a standard Brownian motion is given below.

Definition 9. A stochastic process $B = \{B_t; t \geq 0\}$ is referred to as a standard Brownian motion if

i) $B_0 = 0$ with probability 1

ii) B has independent increments

iii) B has stationary increments

iv) $B_{t+s} - B_t \sim N(0, s)$ where $N\left(\mu, \sigma^2\right)$ denotes a normal distribution with mean σ^2 and variance σ^2.

In the remainder of this report we require the following generalization of a Brownian motion, $W_t = \mu t + \sigma B_t$ which is called a Brownian motion with a drift of μ and volatility σ^2.

4.3 The Normal inverse Gaussian process(N∘IG)

The normal inverse Gaussian distribution was introduced in [3], it is a normal variance-mean mixture distribution where the mixing density is the inverse Gaussian distribution. The inverse Gaussian distribution will not be discussed in this paper, but the interested reader is referred to [24], for further reading.The marginal distribution of this process is a subclass of the hyperbolic distributions. The normal inverse Gaussian process is a four parameter Lévy process. This process has been used as a tool for modeling financial asset returns because of the fact that its properties that are similar to those of financial asset returns.

Definition 10. The normal inverse Gaussian distribution

A random variable X is said to follow a normal inverse Gaussian distribution if its probability density function given by;

$$f\left(x:\alpha,\beta,\mu,\delta\right)=\frac{\alpha\delta}{\pi}\exp\left(\delta\sqrt{\alpha^2-\beta^2}+\beta(x-\mu)^2\right)\frac{K_1\left(\alpha\sqrt{\delta^2-(x-\mu)^2}\right)}{\sqrt{\delta^2-(x-\mu)^2}},$$

where $\alpha>0,-\alpha<\beta<\alpha,\delta>0$ and $K_1\left(x\right)$ is a modified Bessel function of the third order with index 1, see [24]. If X^N has the density given above, then we use the notation $X^N\sim N\circ IG(\alpha,\beta,\mu,\delta)$ distribution. Each parameter has a different effect on the shape of the distribution; α captures the tail heaviness, β determines the level of asymmetry, δ is a scale parameter and μ is a location parameter.

The first four standardized central moments of the $N\circ IG(\alpha,\beta,\mu,\delta)$ distribution are as follows:

- Mean $=\frac{\alpha\beta}{\sqrt{\alpha^2-\beta^2}}+\mu$
- Variance $=\alpha^2\delta(\alpha^2-\beta^2)^{-\frac{3}{2}}$
- Skewness $=3\beta\alpha^{-1}\delta^{-1/2}(\alpha^2-\beta^2)^{-\frac{1}{4}}$
- Kurtosis $=3(1+\frac{\alpha^2+4\beta^2}{\delta\alpha^2\sqrt{\alpha^2-\beta^2}})$

From the above moments we can see that the N∘IG distribution has a kurtosis that is greater than that of a normal distribution. Note, that if $\beta=0$, then the $N\circ IG$ distribution is symmetric. In what follows we define the $N\circ IG$ process in terms of the summation of $N\circ IG$ distributed random variables.

The $N\circ IG$ process, $X=\{X_t,t\geq 0\}$, is a stochastic process with $X_0=0$, independent and identically distributed stationary increments which follow a $N\circ IG$ distribution. In particular $X_t\sim N\circ IG(\alpha,\beta,\mu t,\delta t)$ distribution, which is infinitely divisible.

4.4 The Meixner process

The Meixner distribution was named after German theoretical physicist Josef Meixner, who was know for his work in orthogonal polynomials, see [19]. The Meixner distribution is a special case of generalized

z-distributions, which were introduced in, see [16].

Definition 11. The Meixner distribution

A random variable has the Meixner distribution if its probability density function given by:

$$f(x;\alpha,\beta,\mu,\delta) = \frac{(2\cos(\beta/2))}{2\alpha\pi\Gamma(2d)}\exp\left(\frac{b(x-\mu)}{a}\right)\left|\Gamma\left(\delta+\frac{i(x-\mu)}{\alpha}\right)\right|^2,$$

where $\alpha > 0, -\pi < \beta < \pi, \delta > 0$. If X^M has the density given above, then $X^M \sim Meixner(\alpha,\beta,\mu,\delta)$. Each of the four parameters influence the shape of the distribution: α captures the tail heaviness, β determines the level of asymmetry, δ is a scale parameter and μ is a location parameter.

The first four standardized central moments of the distribution are given by:

- Mean $= \delta\beta\tan(\beta/2) + \mu$.
- Variance $= \frac{1}{2}\alpha^2\delta(\cos^{-2}(\beta/2))$.
- Skewness $= \sin(\beta/2)\sqrt{2/\delta}$.
- Kurtosis $= 3 + (2 - \cos(\beta))/\delta$.

As was the case with the $N \circ IG$ distribution, the Meixner distribution is symmetric if $\beta = 0$. It has a kurtosis greater than 3 which means that it has tails that are heavier than those of a normal distribution. The Meixner process, $X = \{X_t, t \geq 0\}$ is a stochastic process, with $X_0 = 0$, stationary and independent increments. The marginal distribution of X_t folllows a $Meixner(\alpha,\beta,\mu t,\delta)$ distribution, which is also infinitely divisible.

5 Maximum likelihood parameter estimation

The method of maximum likelihood estimation estimates parameters by choosing parameter estimates that make the data as likely as possible. The method of maximum likelihood estimation (mle) is used for estimating the parameters of the various models below. Let $f(x;\theta)$ denote the probability density function of a distribution, where θ is a set of unknown parameters. Our task is to estimate these unknown parameters.

It is assumed that we have n independent observations, $x_1, x_2, ...x_n$ of a random variable X. In our case these realizations of the random variable will be the observed log-returns of the stock prices. The mle is the parameter set that maximizes the likelihood function

$$L(\theta) = \prod_{i=1}^{n} f(x_i;\theta).$$

Note that maximizing the likelihood function is equivalent to maximizing the log-likelihood function.

$$\log L(\theta) = \sum_{i=1}^{n} \log f(x_i; \theta).$$

We use numerical methods to obtain the values of the parameters that maximize the log-likelihood function.

6 Fitting geometric Lévy process models to S&P 500 data

Below we fit the proposed models to the S&P500 stock prices by fitting the respective Lévy processes to the log-returns.

6.1 Geometric Lévy process

Consider the stock price process discussed earlier, $S = \{S_t; t \geq 0\}$. It is an exponential Lévy process which evolves in the form

$$S_t = S_0 \exp(X_t),$$

where $t \in [0, T]$. X_t is an exponential Lévy process. We shall write the stock price process in term of the natural logarithm as follows;

$$X_t = \log\left(\frac{S_t}{S_0}\right),$$

Since we work with the log-returns of the stock prices. Since we proposed three Lévy processes for our model, in the first instance X_t will be considered to be a Brownian motion, thereafter a normal inverse Gaussian process and lastly a Meixner process. In order to estimate the parameters of each of the models we use maximum likelihood estimation described in the previous section.

6.2 Procedure

To illustrate the application of the distributions discussed above, we used R for programming and fitting the distributions in to the S&P500 observed log-returns, see [21]. We start by calculating the log-returns from which we fit the normal distribution, where X_t is assumed to follow a Brownian motion. We observe that the normal distribution does not fit well in to the observed log-returns in Figure 6.

We then proceed to fitting the the normal inverse Gaussian distribution. We make use of the Bessel package installed in R to fit a kernel density estimator, see [17]. We use the method of likelihood estimation for estimating the four parameters of the model. This is done by first generating a large set of possible starting values from the uniform distribution from which we choose the parameter set that

has the largest likelihood. Another R package for fitting Hyperbolic distributions is installed to fit the normal inverse Gaussian density function, see [13].

A similar procedure was used for fitting the Meixner distribution and modeling the Meixner process. The only deviation in the procedure define above, are the packages necessary to program the Meixner process in R. We used the pracma package which is necessary for handling the complex gamma function found in the Meixner density function, see [8], and the optim function is used for the general-purpose optimization based on Nelder–Mead optimization algorithms.

6.3 Results

Figure 7 shows a kernel density estimate of the log-returns and the normal density super imposed in the figure. The red dashed line represents the normal density function and the black solid line represents the kernel density estyimator of log-returns

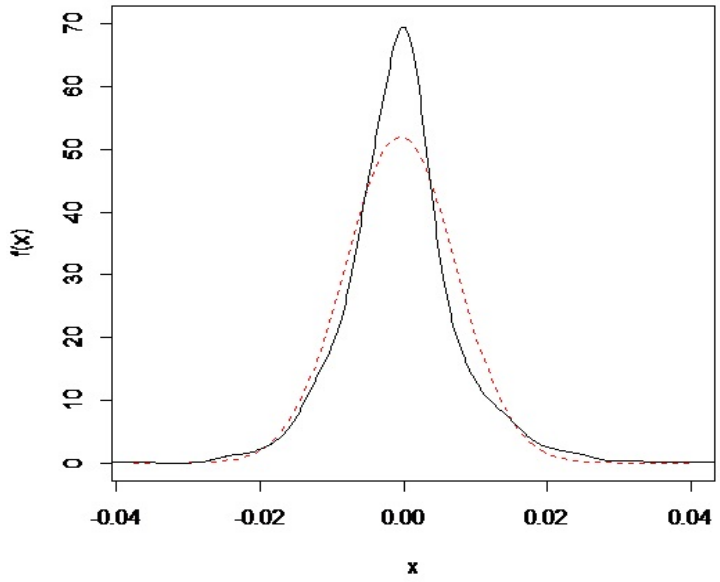


Figure 7: Normal distribution fitted to log-returns.

Figure 8 shows the a fit of the normal inverse Gaussian distribution into the log-returns of S&P 500 dataset. The red dashed line represents the normal inverse Gaussian density function and the black solid line represents the empirical distribution of log-returns. From the graph it is clear that the normal inverse Gaussian fits better than the normal distribution as shown earlier in Section 3 and Figure 7. This suggests that the normal inverse Gaussian distribution can be used to model log-returns of stock prices as suggested by [24] and [4]. Therefore the normal inverse Gaussian process captures most of the

characteristics of log-returns and thus improves the accuracy of the model compared to the traditional Black-Scholes model where the log-returns are driven by a Brownian motion.

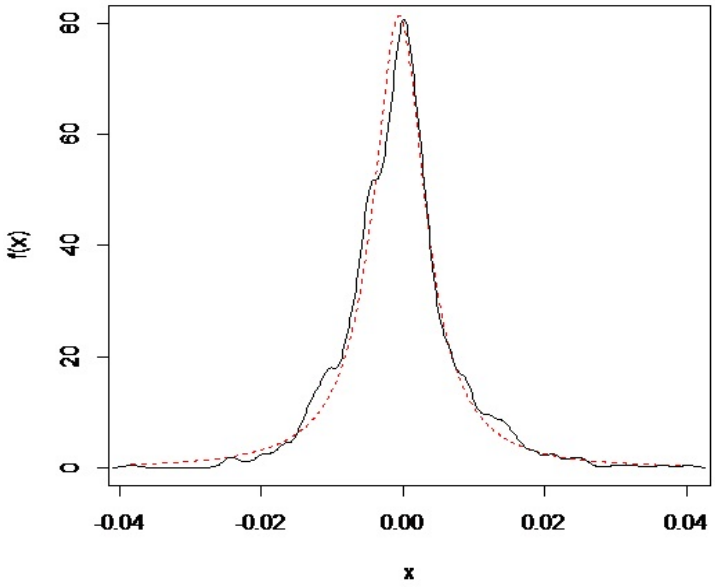


Figure 8: Normal inverse Gaussian distribution fitted to log-returns.

Figure 9 shows the fit of the Meixner distribution to the log-returns of the same S&P 500 data set as used previously for the normal inverse Gaussian distribution. From the graph it is clear that the Meixner distribution fits better than the normal distribution as shown earlier in Figure 6. Hence the Meixner process can also be used to model the log-returns of stock prices. This is in line with the findings of [23] and [24].

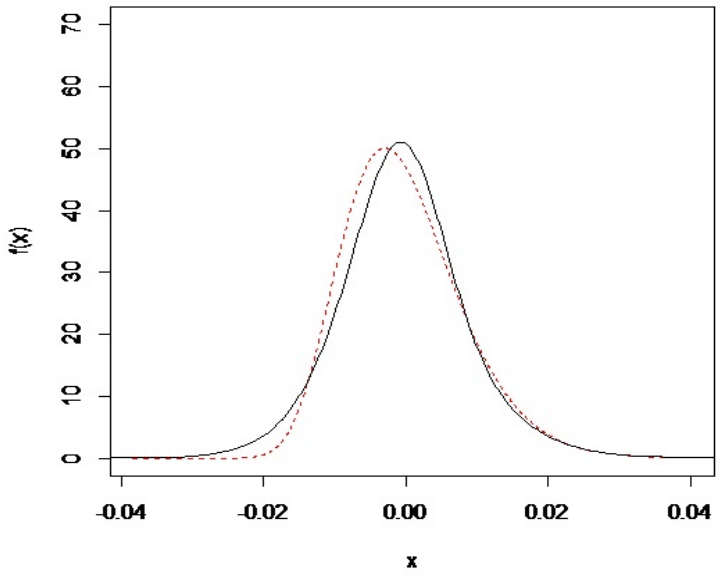


Figure 9: Meixner distribution fitted to log-returns.

7 Conclusion

The research presented has investigated the observed properties of the various distributions assumed to underlie the log-returns of financial time series data. In this research, we discussed properties of log-returns. The most important properties considered include the following: heavy-tailed distribution, negatively skewed distribution, aggregational Gaussianity and volatility clustering.

The properties mentioned above led to the conclusion that the normal distribution is not realistic for modeling log-returns. The need for a more flexible class of distributions arise from these observed properties. Lévy process are used to this end. Desirable properties of Lévy processes include infinite divisibility of the marginal distributions, independent increments and stationary increments. These processes are able to capture the observed skewness and excess kurtosis. Three models were proposed in this paper and the first model is the Black-Scholes model which uses a Brownian motion. Then we use the normal inverse Gaussian distribution to model log-returns of S&P 500 stock index data. The empirical results show that the normal inverse Gaussian distribution fit the log-returns data substantially better than the normal distribution. The third model uses the Meixner distribution to model the same data and the results are similar to those of the normal inverse Gaussian process.

References

[1] E. Akyildirim and H. Mete Soner. A brief history of mathematics in finance. *Borsa Istanbul Review*, 14(1):57 – 63, 2014.

[2] D. Applebaum. *Lévy processes and stochastic calculus*. Cambridge university press, 2009.

[3] O. Barndorff-Nielsen. Exponentially decreasing distributions for the logarithm of particle size. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 353, pages 401–419. The Royal Society, 1977.

[4] O. Barndorff-Nielsen. Normal inverse Gaussian distributions and stochastic volatility modelling. *Scandinavian Journal of Statistics*, 24(1):1–13, 1997.

[5] O. Barndorff-Nielsen. Processes of normal inverse Gaussian type. *Finance and Stochastics*, 2(1):41–68, 1997.

[6] J. Bertoin. *Lévy processes*, volume 121. Cambridge university press, 1998.

[7] T. Bollerslev, R.Y. Chou, and K.F. Kroner. Arch modeling in finance. *Journal of Econometrics*, 52(1):5 – 59, 1992.

[8] H.W. Borchers. *pracma: Practical Numerical Math Functions*, 2017. R package version 2.0.7.

[9] R. Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1:223–236, 2001.

[10] R. Cont. *Volatility Clustering in Financial Markets: Empirical Facts and Agent-Based Models*, pages 289–309. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[11] R. Cont. *Encyclopedia of quantitative finance*. Wiley, 2010.

[12] R. Cont and P. Tankov. Financial modelling with jump processes, 2004.

[13] S. David. *HyperbolicDist: The hyperbolic distribution*, 2009. R package version 0.6-2.

[14] M. Davison. *Quantitative finance: a simulation-based introduction using Excel*. CRC Press, 2014.

[15] S. Figlewski and X. Wang. Is the'leverage effect'a leverage effect? November 2000.

[16] B. Grigelionis. Processes of meixner type. *Lithuanian Mathematical Journal*, 39(1):33–41, 1999.

[17] M. Maechler. *Bessel: Bessel – Bessel Functions Computations and Approximations*, 2013. R package version 0.5-5.

[18] B. Mandelbrot. When can price be arbitraged efficiently? a limit to the validity of the random walk and martingale models. *The Review of Economics and Statistics*, 53(3):225–236, 1971.

[19] J. Meixner. Orthogonale polynomsysteme mit einer besonderen gestalt der erzeugenden funktion. *Journal of the London Mathematical Society*, 1(1):6–13, 1934.

[20] L. Quigley and D. Ramsey. Statistical analysis of the log returns of financial assets. *Financial mathematic, University of Limerick*, 2008.

[21] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[22] K. Sato. *Lévy processes and infinitely divisible distributions*. Cambridge university press, 1999.

[23] W. Schoutens. *The Meixner process: theory and applications in finance*. Eurandom Eindhoven, 2002.

[24] W. Schoutens. *Lévy processes in finance: pricing financial derivatives*. John Wiley & Sons Inc., 2003.

[25] W. Schoutens and J.L. Teugels. Lévy processes, polynomials and martingales. *Stochastic Models*, 14(1-2):335–349, 1998.

Appendix A

This section gives a description of the code we used for modelling S&P500 data and fitting the normal inverse Gaussian distribution on the log-returns in R.

```
##########################################################################
#Entering the data
data = read.csv("table.csv")
prices = data[,4]
prices = prices[length(prices):1]
plot(prices,type = "l")
##########################################################################
#Calculating log-returns
n = length(prices)-1
logrets = 1:n*0
for (j in 1:n){ logrets[j] = log(prices[j+1]/prices[j]) }
plot(logrets,type = "l") plot(hist(logrets)) lines(density(logrets))
##########################################################################
#Fitting a normal distribution
muHat = mean(logrets)
sigmaHat = sd(logrets)*sqrt((n-1)/n)
x_min = min(logrets)
x_max = max(logrets)
l_vec = 100
x = (1:l_vec)/l_vec *(x_max-x_min) +x_min
plot(x,dnorm(x,muHat,sigmaHat),col = "red",ylim = c(-0.1,85),type = "l")
lines(density(logrets))
##########################################################################
#NoIG density
alpha = 2
beta = 1
mu = 1
delta = 1
#install.packages("Bessel") library(Bessel)
f_NoIG <- function(x,alpha,beta,mu,delta)
```

```
{ f = alpha*delta/pi*exp(delta*sqrt(alpha^2-beta^2)+beta*(x-mu))*BesselK(alpha*sqrt(delta^2+(x-
mu)^2),1)/sqrt(delta^2+(x-mu)^2); return(f) }
x_min = -1
x_max = 5
l_vec = 100
x = (1:l_vec)/l_vec *(x_max-x_min) +x_min
y = f_NoIG(x,alpha,beta,mu,delta)
plot(x,y,type="l")
##################################################################
# NoIG likelihood
minLL_NoIG <- function(parms)
{ alpha = parms[1]
beta = parms[2]
mu = parms[3]
delta = parms[4]
if (alpha>0 & alpha>abs(beta) & delta>0)
{
LLvec = f_NoIG(logrets,alpha,beta,mu,delta)
LLvec = log(LLvec)
LL = sum(LLvec) }
else { LL = Inf } minLL = -LL return(minLL) }
parms = c(alpha,beta,mu,delta)
mLL = minLL_NoIG(parms)
##################################################################
# Starting values
n_startvals = 1000
alpha_s <- runif(n_startvals,1,100)
beta_s <- runif(n_startvals,-80,80)
mu_s <- runif(n_startvals,-10,10)
delta_s <- runif(n_startvals,1,100)
startevals <- rep(0,n_startvals)
for (k in 1:n_startvals)
{ sparms <- c(alpha_s[k],beta_s[k],mu_s[k],delta_s[k])
startevals[k] <- minLL_NoIG(sparms)
```

```
}
minval <- min(startevals[is.finite(startevals)])
indx = which.min(abs(startevals-minval))
#startevals[indx]
startvals <- c(alpha_s[indx],beta_s[indx],mu_s[indx],delta_s[indx])
optm <- optim(startvals,minLL_NoIG)
alpha = optm$par[1]
beta = optm$par[2]
mu = optm$par[3]
delta = optm$par[4]
x_min = min(logrets)
x_max = max(logrets)
l_vec = 100
x = (1:l_vec)/l_vec *(x_max-x_min) +x_min
y = f_NoIG(x,alpha,beta,mu,delta)
plot(x,y,col = "red",ylim = c(-0.1,100),type = "l") lines(density(logrets,adjust = 0.7))
##########################################################################
```

Appendix B

This section gives a description of the code we used for modelling S&P500 data and fitting the Meixner distribution on the log-returns in R.

```
##########################################################################
#Entering the data
data = read.csv("C:/Users/Nqaba/Downloads/^GSPC (1).csv")
prices = data[,5]
prices = prices[length(prices):1]
plot(prices,type = "l")
##########################################################################
#Calculating log-returns
n = length(prices)-1
logrets = rep(0,n)
for (j in 1:n)
{ logrets[j] = log(prices[j+1]/prices[j]) }
plot(logrets,type = "l")
```

```
plot(hist(logrets))
lines(density(logrets))
##########################################################################
#Meixner density
install.packages("pracma")
library(pracma)
f_Meixner <- function(x,alpha,beta,mu,delta)
{
T1 = ((2*cos(beta/2))^(2*delta))
T2 = (2*alpha*pi*gamma(2*delta))
T3 = (beta*(x-mu)/alpha)
T4 = abs(gammaz(delta+1i*((x-mu)/alpha)))
f_Meixner = T1/T2*exp(T3)*T4^2
;
return(f_Meixner) }
alpha = 2
beta = 1
mu = 0
delta = 1
x_min = -3
x_max = 8
x = seq(x_min,x_max,(x_max-x_min)/999)
y = f_Meixner(x,alpha,beta,mu,delta)
plot(x,y,type="l")
##########################################################################
# Meixner likelihood
LLvec = f_Meixner(logrets,alpha,beta,mu,delta)
LLvec = log(LLvec)
LL = sum(LLvec)
if (alpha>0 & abs(beta)<pi & delta>0)
{
LLvec = f_Meixner(logrets,alpha,beta,mu,delta)
LLvec = log(LLvec)
LL = sum(LLvec) } e
```

```
lse { LL = -Inf }
parms = c(alpha,beta,mu,delta)
minLL_Meixner <- function(parms)
{
alpha = parms[1]
beta = parms[2]
mu = parms[3]
delta = parms[4]
if (alpha>0 & abs(beta)<pi & delta>0)
{
LLvec = f_Meixner(logrets,alpha,beta,mu,delta)
LLvec = log(LLvec)
LL = sum(LLvec)
}
else { LL = -Inf }
minLL = -LL
return(minLL) }
parms = c(alpha,beta,mu,delta)
mLL = minLL_Meixner(parms)
##################################################################
# Starting values
n_startvals = 10000
alpha_s <- runif(n_startvals,0.01,100)
beta_s <- runif(n_startvals,-pi,pi)
mu_s <- runif(n_startvals,-10,10)
delta_s <- runif(n_startvals,0.01,100)
startvals = rep(0,4)
startevals = rep(0,n_startvals)
besteval = Inf for (k in 1:n_startvals)
{
sparms <- c(alpha_s[k],beta_s[k],mu_s[k],delta_s[k])
startevals[k] <- minLL_Meixner(sparms)
if (is.finite(startevals[k]) & startevals[k]<besteval)
{
```

```
startvals = sparms
besteval = startevals[k]
}
}
besteval
startvals
minLL_Meixner(startvals)
optm <- optim(startvals,minLL_Meixner)
alphaHat = optm$par[1]
betaHat = optm$par[2]
muHat = optm$par[3]
deltaHat = optm$par[4]
x_min = min(logrets)
x_max = max(logrets)
x = seq(x_min,x_max,(x_max-x_min)/999)
y = f_Meixner(x,alphaHat,betaHat,muHat,deltaHat)
plot(x,y,col="red",ylab = "f(x)",main = "Meixner distribution fitted on logreturns",type="l",ylim=c(0,30))
lines(density(logrets,adjust=2.5))
##########################################################
```

Appendix C: SAS Code

SAS code used for normality test:

```
data a;
input logreturns @@;
cards;
7.302E-13
0.3984877
0.3989423
1.449E-12
2.845E-12
5.532E-12
1.065E-11
2.03E-11
3.83E-11
7.156E-11
1.324E-10
2.424E-10
4.395E-10
7.888E-10
1.4019E-9
2.4665E-9
4.2965E-9
7.4098E-9
1.2652E-8
2.1387E-8
3.5795E-8
5.9312E-8
9.73E-8
1.5803E-7
2.5412E-7
4.0456E-7
6.3766E-7
9.9506E-7
1.5373E-6
```

2.3515E-6

3.561E-6

5.3391E-6

7.9252E-6

0.0000116

0.0000169

0.0000244

0.0000348

0.0000492

0.0000687

0.0000951

0.0001303

0.0001768

0.0002375

0.0003158

0.0004157

0.0005419

0.0006993

0.0008934

0.0011301

0.0014152

0.0017547

0.0021539

0.0026177

0.0031497

0.003752

0.0044251

0.005167

0.0059733

0.0068366

0.0077469

0.008691

0.0096532

0.0106153

0.011557

0.0124571

0.0132937

0.0140454

0.0146919

0.0152152

0.0156003

0.0158361

0.0159155

0.0158361

0.0156003

0.0152152

0.0146919

0.0140454

0.0132937

0.0124571

0.011557

0.0106153

0.0096532

0.008691

0.0077469

0.0068366

0.0059733

0.005167

0.0044251

0.003752

0.0031497

0.0026177

0.0021539

0.0017547

0.0014152

0.0011301

0.0008934

0.0006993

0.0005419

0.0004157

0.0003158

0.0002375

0.0001768

;

data a;

input logreturns;

; title 'Analysis of Log-returns';

ods select Histogram ParameterEstimates GoodnessOfFit FitQuantiles Bins;

proc univariate data=a;

histogram / normal(percents=20 40 60 80 midpercents)

odstitle = title;

inset n normal(ksdpval) / pos = ne format = 6.3; run;

var logreturns;

run;

Determinants of judicial decision-making

Nicola Gawler 12184587

STK795 Research Report

Submitted in partial fulfillment of the degree BCom(Hons) Statistics

Supervisor: Dr J van Niekerk

Department of Statistics, University of Pretoria

30 October 2017

Abstract

In this study classification methods will be used to predict the judicial decisions that are made in the U.S. Supreme Court by judges of the Roberts Court. The decision made by the Supreme Court either affirms or reverses the lower court's ruling. Machine learning techniques will be used to model the binary categorical variable as affirmed or reversed and to ascertain influential factors and predictors.

Declaration

I, *Nicola Gawler,* declare that this essay, submitted in partial fulfillment of the degree *BCom(Hons) Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____

Nicola Gawler

_____

Janet van Niekerk

30/10/2017

_____

Date

Acknowledgements

The author would like to acknowledge Statomet and Center for AI Research, Meraka Institute, CSIR for the financial support. The opinions and views expressed herein are that of the author and not Statomet or CAIR.

Contents

List of Figures

List of Tables

1 Introduction

Judicial decision-making should ideally be independent of a judge's personal ideology, political affiliation, and other external factors. This is unfortunately not always true [7]. The Supreme Court of Appeal is the highest federal court in the United States with eight associate judges and one Chief Justice. In each of the 88 court cases that are heard in a year, each judge gets one vote in every case and a decision is based on the majority vote. Records of all decisions made by the Supreme Court since 1946 are available to the public. The decisions are contained in a database that is coded into 60 variables per case which are composed of a further 2633 elements [15]. This database will be used in order to develop the classification models focusing on cases heard since the beginning of the Roberts Court, which is the court since John G. Roberts became Chief Justice in 2005.

The objective of this study is to build a statistical model using these specific Supreme Court cases and to establish which variables may be determinants in the judicial decision-making process. The outcome variable that will be modelled is a binary categorical variable, namely whether the Supreme Court affirms or reverses the lower court's ruling. A classification tree model has been used to forecast this outcome variable in the past [14]. Logistic regression is a conventional model for a binary response, as in this case, but it suffers from parametric assumptions which might be misspecified. For this reason a random forest, which is a nonparametric model, will be developed to classify the decisions into either affirmed or reversed. Since the data is labelled, an estimated error rate can be obtained to evaluate the prediction accuracy of the model. Missing data is often a concern in this type of statistical analysis. In this study missing data is dealt with using an extra logical category.

From a statistical point of view, this study is not necessarily unique in its classification approach to modelling this dataset, but rather in its focus of what the classifier can reveal. A recent paper made use of a random forest classifier to predict the outcome of Supreme Court cases over an extended period of time. The study focused on building a predictor that could be applied to a myriad of cases heard by the Supreme Court, past and future, without paying attention to the interpretability [8]. This study, however, focuses on understanding the factors that influence a judicial decision and whether those decisions are based on the judge's personal ideology.

From a legal perspective, this study will make a contribution to the discourse around judicial decision-making and the determinants (legal and political factors) that shape the decision-making process. Although judges write opinions and reasoning for most of their judgments, this does not necessarily capture all factors that may influence the decision-making process. Political factors that are statistically significant determinants of judicial decisions are of particular importance because of the need for judicial impartiality. The personal political ideology of a judge should not influence a judge's decision. A judge's political ideology is represented in the Supreme Court Database by the variable *justiceName* and *scIdeol-*

ogy, a judge's Segal-Cover score. Due to the far-reaching consequences of the Supreme Court's decisions, prediction of the outcome of a case is valuable information for court watchers and could be used as guidelines for the legislative branch of government in policy making.

In Sections 2, 3 and 4 theory will be discussed. Section 2 deals with decision trees, Section 3 with bagging and then Section 4 with random forests. This concludes the theoretical section of this study. Section 5 deals with application on the Supreme Court dataset.

2 Decision trees

2.1 Introduction

There are two broad categories of decision trees: regression and classification trees. The objective of a classification tree is to predict a categorical response (outcome) variable given certain predictor (input) variables, as opposed to regression trees where the outcome variable is numerical. The use of decision trees is a relatively old technique, but it has recently gained popularity since it has been discovered that higher accuracy can be achieved by an ensemble of different trees when generalising [4].

Prior to the development of this technique, linear discriminant analysis (1930), logistic regression (1944) and nearest neighbours classifiers (1951) were used to solve classification problems. Decision trees were developed in 1963 by Morgan and Sonquist, and this technique was fine-tuned by other notable pioneers such as Breiman, Friedman, Olshen, Stone and Quinlan [16]. One of the motives for the development of this technique is that is it a nonparametric model with no formal distributional assumptions. Consequently, it is capable of handling non-linear interactions and classification decision boundaries. This technique also has several advantages such as being able to deal with mixed data (discrete and continuous) and missing values, variable selection is automatic and it can be interpreted with ease [12].

2.2 Definition and structure

A decision tree is a collection of nodes or questions organised in a hierarchical manner [4]. The base of the tree is a called a root node where the full dataset is inputted and a split function is applied, splitting the data into daughter nodes. This split can be binary (two nodes) or non-binary (more than two nodes) [12], however, this study will focus on binary nodes. The subset of incoming data to each daughter node then gets split recursively at each internal node until the data point reaches a terminal or leaf node that contains a classifier (predictor) which associates an output (class label) with the input [4]. It is also to be noted that an estimated probability of membership of a particular class can also be obtained.

There are two conceptual phases in the development of the classification tree: the growth of the classification tree and pruning of the classification tree [13].

2.2.1 Growing the tree

The parameters (the test at each node and the leaf predictors) can be selected by hand if the data is very simple but the tree structure and its parameters are more generally learnt automatically from training data, as an algorithm. The test function associated with each split node depends on the subset of incoming training data at that node. The parameters that best split the training data is learned by maximising an objective function (splitting criteria) at each node [4].

There are many different splitting criteria, however, most are univariate impurity based criteria favouring the purer split (majority vote). Univariate splitting criteria only consider a single attribute per node for the split [13].

The incoming training data is best split into the two child nodes, which is formulated as a maximization of an objective function, I_j, at node j.

$$\theta_j^* = \arg\max_{\theta_j \epsilon \mathcal{T}} I_j$$

where

- $I_j = I(S_j, S_j^L, S_j^R, \theta_j)$ is the objective function (defined abstractly)
- S_j is the subset of incoming training data at node j
- $S_j^L = \{(v,y)\epsilon S_j \mid h(v,\theta_j) = 0\}$ and $S_j^R = \{(v,y)\epsilon S_j \mid h(v,\theta_j) = 1\}$ are the subsets of the training data at node j that go to the left and right child node, respectively
- (v,y) is a training point where v is the vector of input features and y is a known label
- θ_j^* are the parameters of the test function at node j, obtained from optimising the objective function I_j in the training phase
- $h(v,\theta_j)$ is the test function at node j with a binary outcome, 0 or 1

I_j, the objective function, was defined abstractly as there are many possible objective functions. The splitting options available in R (rpart), "information" and "gini", make use of two different objective functions. These two objective functions are based on measures of node purity/homogeneity, the first being based on the Shannon entropy and the second on the Gini index.

Definition 1. Shannon entropy is defined as

$$H(S) = -\sum_{c \epsilon \mathcal{C}} p(c)\log(p(c))$$

where

- *p(c)* is the probability that a random entry in the leaf belongs to class c
- c is the class label and $\mathcal{C}$ is the set of all classes

Entropy is a measure of purity/homogeneity in the split and the leaves of the splits. It is the discrete empirical distribution obtained from the training points within the set S. The "best split" is a purer split that will therefore have lower entropy or uncertainty of prediction.

Although entropy is not specified as an objective function by R, it used indirectly through information gained. Maximising the information gained is equivalent to minimising the entropy [12].Information gained is a measure of the effectiveness of a feature in classifying the training data. It is the difference between the entropy of all the incoming data (the training data) and the expected entropy (the weighted sum of the child entropies) after the data is split using a certain attribute [11]. If the children distributions are purer (a lower entropy) and the information content has increased, this is the desired improvement from a split. The "best split" is chosen on the basis of the attribute that gives the **maximum information gained** [4].

Definition 2. Information gained when using a certain attribute/split, is defined as

$$\begin{aligned} I_{info} &= H(S) - \sum_{i \epsilon \{L,R\}} \frac{\mid S^i \mid}{\mid S \mid} H(S^i) \\ I_{info} &= H(S) - \frac{\mid S^L \mid}{\mid S \mid} H(S^L) - \frac{\mid S^R \mid}{\mid S \mid} H(S^R) \end{aligned}$$

The Gini index is the alternative measure of node impurity used by R. The "best split" is chosen in the same manner as above, where the feature or split with the **maximum impurity reduction** is preferred.

Definition 3. Gini index is defined as

$$G(S) = \sum_{c \epsilon C} p(c)(1 - p(c))$$

Definition 4. Gini measure, the impurity reduction when using a certain attribute/split, is defined as

$$\begin{aligned} I_{gini} &= G(S) - \sum_{i \epsilon \{L,R\}} \frac{\mid S^i \mid}{\mid S \mid} G(S^i) \\ I_{gini} &= G(S) - \frac{\mid S^L \mid}{\mid S \mid} G(S^L) - \frac{\mid S^R \mid}{\mid S \mid} G(S^R) \end{aligned}$$

The termination of this phase is dependent on the chosen stopping criteria. Common criteria involve a certain threshold tree depth, whether all attributes in a training subset belong to a single class or are sufficiently homogenous (the best splitting criteria is not greater than a certain threshold, this is a trade-off between cost and gain) or whether a node contains too few training points [4, 12].

2.2.2 Pruning the tree

The tree structure (the depth of the tree) is determined by the stopping criteria and the pruning of a tree which together optimise the ability of a tree to generalise. An increase in the depth of the tree initially increases the generalisation accuracy of the model, however, this accuracy starts to decrease after a certain optimal tree depth. Beyond this optimal depth, the model tends to overfit the training dataset, resulting in a poor ability to generalise independent test data [11].

If the stopping criteria is too loose then the model tends to overfit training dataset. The problem of overfitting could be solved with stricter stopping criteria, however, this method is not favoured as it is considered myopic and it can result in under-fitting the model. The preferred solution is first growing the "full tree" with looser stopping criteria and then pruning it, in other words, rid the model of unnecessary branches that do not contribute sufficiently to the generalisation accuracy [12]. There are various pruning methods, but there is no single method that is said to be the most suitable for every classification tree. This is known as the no free lunch theorem. Most criteria that are used to decide which branches should be pruned use a cross-validation error as an indication of generalisation accuracy. The trade-off between accuracy and simplicity is also taken into account in many methods, such as the cost-complexity method [13].

2.3 Titanic example

Rpart is the package used by R to implement recursive partitioning for the development of a classification tree model as described above, in other words, it grows a classification tree.[1] The dataset used in the execution of this example is data provided by Kaggle for the competition "Titanic: Machine Learning from Disaster" [9]. This dataset is commonly used in classification analysis and for illustrative purposes. This dataset will be used throughout the study.

The outcome variable modelled or predicted was whether or not a passenger survived or not given 6 input variables. These 6 variables are: the ticket class (*Pclass*), the sex of the passenger (*Sex*), the age of the passenger (*Age*), the number of siblings or spouses on board (*SibSp*), the number of parents or children on board (*Parch*), the passenger's fare (*Fare*) and the port of embarkation for passengers (*Embarked*). The following R code was used to in building a classification tree to model the outcome

[1] This analysis was performed using R software, Version 3.3.1 for Windows. Copyright © 2016 The R Foundation for Statistical Computing Platform, Vienna, Austria.

variable, namely whether a passenger survived or not (*Survived*).

```
1  library(rpart)
2  library(rpart.plot)
3  library(rattle)
4  fit1 <- rpart(Survived ~ Pclass + Sex + Age + SibSp + Fare + Embarked + Parch,
       data = 'Titanic train', method = "class", parms = list(split = 'gini'))
5  fit2 <- rpart(Survived ~ Pclass + Sex + Age + SibSp + Fare + Embarked + Parch,
       data = 'Titanic train', method = "class", parms = list(split = 'information'))
6  fancyRpartPlot(fit1,cex=0.58, suffix="\n\n",under.cex=0.58)
7  fancyRpartPlot(fit2,cex=0.53, suffix="\n\n",under.cex=0.53)
8  plotcp(fit1)
9  pfitt<-prune(fit1,cp=fit1$cptable[which.min(fit1$cptable[,"xerror"]),"CP"])
10 fancyRpartPlot(pfitt, cex=0.6, suffix="\n\n",under.cex=0.6)
```

Line 4 and 5 of the programme make use of the rpart package to fit a decision tree using the Gini Index and Information Gained, respectively, as the splitting criteria. Effectively growing the classification tree.

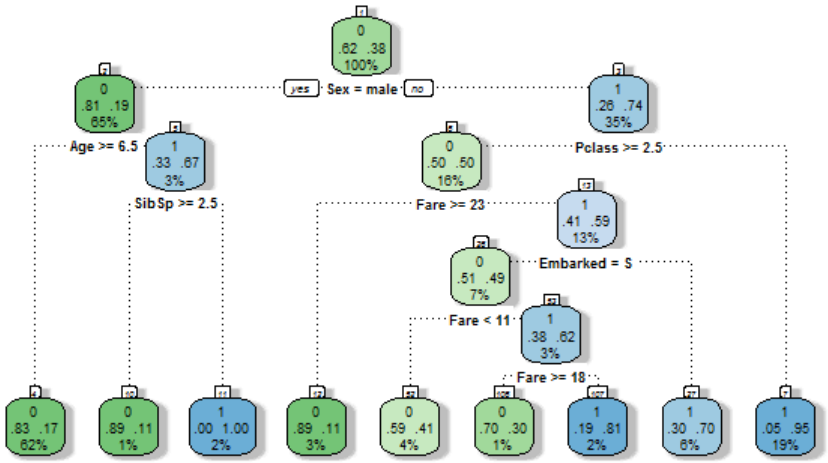


Figure 1: Classification tree generated using the Gini index as the splitting criteria

Line 8 creates Figure 3, which is a visual representation of the cross validation error of the fitted model against the size of the tree. Rpart uses a built in cross validation method (k-fold cross validation) to determine the accuracy of the model, where it splits the data into a training and validation dataset. The training set is used to build the model and the validation set is used to determine how accurately the model is able to generalise on the unseen (validation) data. This is repeated k-fold times and then averaged to give a final measure of performance.

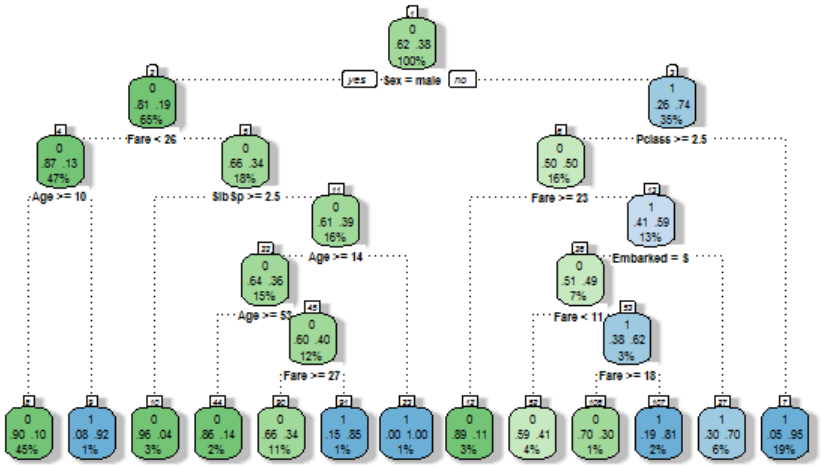


Figure 2: Classification tree generated using information gained as the splitting criteria

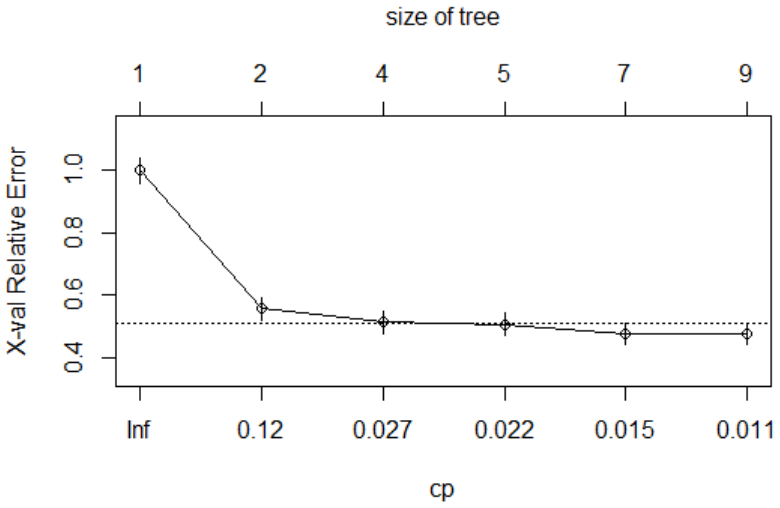


Figure 3: Cross validation error against the complexity parameter and tree size

The next phase is the pruning of the tree, which is executed in line 9 and 10. Here the tree size or complexity parameter (cp) is chosen when the cross-validation error is at its minimum point. This is the optimal cp value or size to which the tree will be pruned. R recommends that the optimal cp is "often the leftmost value for which the mean lies below the horizontal line". In the Titanic example, this is when the size of the tree is 3. Figure 4 below shows this optimal number of nodes (3) which is a result of pruning the original tree, Figure 1.

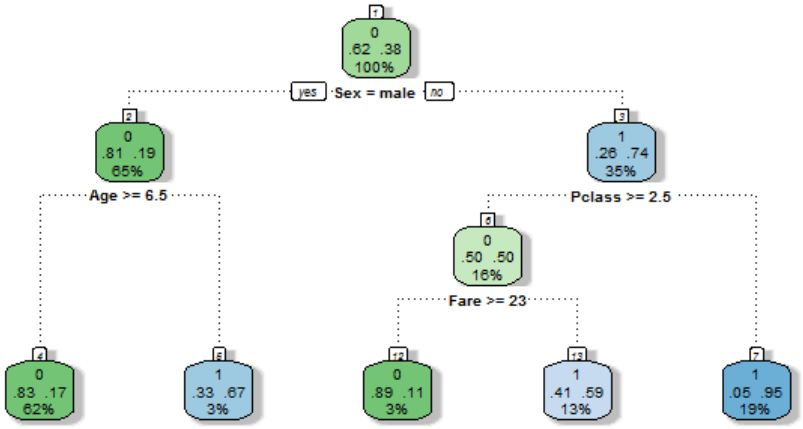


Figure 4: Pruned classification tree

3 Bagging

Bagging, which was proposed by Breiman in 1996 [1], is a method used to inject randomness into a system in an attempt to improve the accuracy of machine learning predictors and avoid overfitting [4]. The word bagging was coined through a combination of "bootstrapping" and "aggregating". This is where numerous versions of a predictor are created by sampling different subsets of the training data, after which these predictors are aggregated to form one predictor. Bootstrapping, which is a simple random sampling technique with replacement, is used to create the different subsets of training data. These classification predictors are aggregated by either averaging posterior probabilities or by voting, where each predictor makes its classification and then the majority decision is used as a final classification [1].

4 Random Forests

4.1 Introduction

Attribute bagging is the main modification that distinguishes a random forest from the original bagging algorithm used in building and aggregating multiple decision trees [1]. This is a process whereby a multitude of decision trees are grown using independent bootstrap samples to create different subsets of the training data [3]. The differentiating factor of random forests is that at each node of the individual trees grown, all available input features are not used. Instead, m input features are independently and randomly selected out of all M possible features [6]. The splitting criteria is then applied to the m input features to find the best split at that particular node. This process continues until the decision trees are fully grown to their maximum depth. The decision trees are unpruned as pruning is no longer necessary because overfitting is addressed by the two injections of randomness [2].

4.2 Structure and methodology

The two injections of randomness in the progression from a single decision tree to a random forest are:

- bootstrap sampling of the training data;
- attribute bagging of the features at each node [4].

As in bagging, these trees are then aggregated in order to develop one final classification by either averaging posterior probabilities of the outcomes produced by each tree or by majority voting of the class outcome produced by each tree [4]. The rationale behind this development is to reduce the correlation between trees, thereby improving the accuracy and stability of the ensemble of classifiers (decision trees) [2, 4]. This correlation between trees is due to strong predictors being selected as the features in many of the trees which are created by different ordinary bootstrap samples [3].

Due to the nature of random sampling, about a third of the cases are excluded from the bootstrap sample used to construct an individual tree. These cases that are left out are called "out of bag" (OOB) data. This data is valuable as random forests no longer require cross-validation or a separate test set to obtain an unbiased estimate of the classification error as this is estimated internally with the OOB data [2]. It is also valuable as it addresses the major disadvantage of a random forest: the loss of visual representation provided by an individual tree that provides insights into the variable importance and the point at which they become important. This loss is due to the aggregation process in the formation of a random forest classifier that only generates a final classification [12]. This issue is addressed by two measures of variable importance computed using the OOB data. These two measures are the permutation importance measure and the Gini importance measure. Permutation importance is obtained by calculating the classification error rate by making use of the OOB data, before and after a single predictor variable is permuted [10]. The difference between these two values is an indication of variable importance because if the permutation of a certain variable had little impact on the error rate this would imply that the particular variable is unimportant [6]. This measure is obtained for every variable, across all decision trees. The final measure of variable importance is obtained by averaging and standardising these differences, divided by the standard deviation of the differences. The Gini importance measure is based on the Gini impurity measure, which is the reduction in node impurity achieved when splitting on a certain variable. The greater the decrease in impurity, the more important a variable is. The final measure is obtained by averaging the Gini impurity measure for each variable across all decision trees generated by the random forest [10].

4.3 Titanic example continued

The randomForest package in R is used to implement Leo Breiman's random forest algorithm for classification problems [10]. One problem faced when using random forests is how to deal with missing data, which is not an issue when using a decision tree. There are various ways in which this can be dealt with and appropriate techniques depend on the specific dataset. The summary function in R is very useful in identifying which variables are problematic in this regard.

```
1  library(rpart)
2  library(randomForest)
3  library(party)
4  summary(`Titanic train`)
5  Agefit <- rpart(Age ~ Pclass + Sex + SibSp + Parch + Fare + Embarked, data=`
       Titanic train`[!is.na(`Titanic train`$Age),], method="anova")
6  `Titanic train`$Age[is.na(`Titanic train`$Age)] <- predict(Agefit, `Titanic train
       `[is.na(`Titanic train`$Age),]) summary(`Titanic train`$Embarked)
7  which(`Titanic train`$Embarked == '')
8  `Titanic train`$Embarked[c(62,830)] = "S"
9  `Titanic train`$Embarked <- factor(`Titanic train`$Embarked)
10 set.seed(12184587)
11 RF <- randomForest(as.factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare
       + Embarked, data=`Titanic train`, importance=TRUE, ntree=2000, replace=TRUE)
12 varImpPlot(RF,main="Investigation of variable importance")
13 importance(RF)
14 print(RF)
```

For the Titanic dataset, the *Age* variable is particularly problematic as it has 177 missing values. The programming in lines 5 and 6 uses a decision tree, built using all the data without missing values, to impute what the age of a passenger would be where there is missing information. The second problematic variable is *Embarked*, where two of the passengers' embarkment details are missing. Due to the fact that there are only two missing values and that *Embarked* is a categorical variable, the most appropriate method is arguably to replace the missing values with the mode of the missing variable - which for *Embarked* is *S* for Southampton.

Once the problem of missing data has been dealt with, a random forest can be constructed. This is executed by line 11. The programme uses the OOB data from the construction of the random forest in order to obtain a confusion matrix and an unbiased estimate of the classification error. This OOB error rate estimate is 16.05% and the confusion matrix is given in Table 1.

	0	1	classification error
0	506	43	0.07832423
1	100	242	0.29239766

Table 1: Confusion matrix when predicting *Survived* and imputing the missing values using the mode (for *Embarked*) and a decision tree (for *Age*)

As mentioned earlier, there are various ways to deal with missing data before the random forest classifier can be constructed. The randomForest package has a 'na.action' option, which makes it possible to deal with NA values automatically in many different ways, two examples being 'na.omit' or 'na.roughfix'. The first options omits observations with missing values and the second option imputes the missing values by making use of a median or mode value for that variable. There is a third option specific to the randomForest package called rfImpute which imputes missing values by making use of a proximity measure.

Line 3 in the programme below constructs a random forest and makes use of the 'na.action' option to deal with missing values.

```
1 TitanicNA<-read.csv("C:/Users/Nicola/Documents/2017/STK795/JUNE/trainNA.csv")
2 set.seed(12184587)
3 RanF<-randomForest(as.factor(Survived)~Pclass + Sex + Age + SibSp + Parch + Fare +
      Embarked, data='TitanicNA', importance=TRUE, ntree=2000, replace=TRUE, na.
    action=na.roughfix)
4 varImpPlot(RF)
5 importance(RanF)
6 print(RanF)
```

The OOB error rate estimate using this method is 16.39%. The confusion matrix constructed using OOB data can be seen in Table 2.

	0	1	classification error
0	504	45	0.08196721
1	101	241	0.29532164

Table 2: Confusion matrix when predicting *Survived* and imputing the missing values using 'na.roughfix'

It is interesting to note that the OOB estimate of error rate is lower for the first method used to deal with missing data, using a decision tree to impute the missing *Age* variable and the mode for the missing *Embarked* variable, as opposed to the second 'na.roughfix' method, which just made use of a median or mode to impute the missing variables.

Figure 5 was constructed using line 12 in the first programme and line 4 in the second programme. The randomForest package makes use of two measures of variable importance. As explained earlier, the

mean decrease in accuracy and the mean decrease in node impurity. Although the visual representation of a single decision tree is lost, Figure 5 is a good substitute for this visual representation as it makes variable importance quite comprehensible.

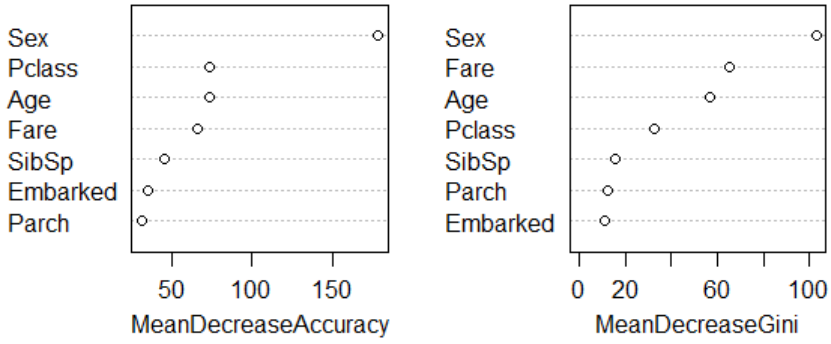


Figure 5: Variable importance in predicting *Survived*

The MeanDecreaseAccuracy is the permutation based measure of variable importance, where the bigger the difference in the classification error rate due to permutation of the variable is an indication of higher variable importance in the classification problem [6]. The interpretation of Figure 5 is therefore that the rank of variable importance when classifying whether a passenger survived or not from most important to least is: *Sex, Pclass, Age, Fare, SibSp, Embarked* and *Parch*. For instance, if the variable *Sex* is permuted it would lead to a large decrease in accuracy, which indicates the importance of the variable. Contrary to this, if the variable *Parch* is permuted, it is clear from the graph that it would lead to a small decrease in accuracy, therefore, indicating the unimportance of the variable.

The graph showing MeanDecreaseGini against the variables, on the right in Figure 5, comes to a slightly different finding with the rank of variable importance, namely: *Sex, Fare, Age, Pclass, SibSp, Parch* and *Embarked*. The logic behind this interpretation is that the higher the MeanDecreaseGini measure is, the greater the decrease in node impurity is when splitting on a certain variable, implying a greater important of a variable in the classification problem at hand [10].

5 Application

5.1 Introduction

The objective of this study is to establish which variables may be determinants in the judicial decision-making process as judicial decisions made should ideally be independent from a judge's personal ideology, political affiliation, and other external factors. In effect, many of the variables should ideally be insignificant, for example *justiceName*, in predicting a judicial decision or the outcome of a case. In this study, the Supreme Court dataset is used to investigate the impartiality of the judges in the decision-making

process.

5.2 Variable overview

Although there are 60 variables per case in the database, not all of these variables are logical or relevant to use for the objective of this study. The variables are broadly categorised by the database into six categories: identification variables, background variables, chronological variables, substantive variables, outcome variables and voting and opinion variables.

Identification variables

- The identification variables are different for every case and were not used because despite the supposed variable importance being potentially high, it is not meaningful or logical to conclude that these have any impact on the outcome of a case. For example the Supreme Court Citation variable (*sctCite*) cannot be used to model the outcome of a case in a meaningful way.

Background variables

- The background variables should not, in theory, be determinants in judicial decision-making so it is important to include these variables.
- Some of these variables have been excluded, however, as the difference between the *caseOrigin* and the *caseOriginState* is already captured in the variable *jurisdiction*.
- The other variables excluded were done so on the basis of being poorly coded variables. To illustrate, the variable *respondent* has 310 distinct values with many of these distinctions being superficial and overlapping, for example, value 154 is a female employee or job applicant and value 155 is a female. The three-judge District Court variable (*ThreeJudgeFdc*) was excluded on the basis that it is almost 'non-existent' in the Roberts Court.

Chronological variables

- Out of the chronological variables, the only 2 variables kept were *term* and the *naturalCourt*.
- The *term* is the year in which the decision was passed down, which is the most telling date, as deliberation can occur up until this date.
- The *naturalCourt* is also an interesting variable as there are natural courts that are considered to be stronger than others. A natural court is a court where no personnel change occurs, in other words, the same 9 judges are in office for a period of time.

Substantive variables

- All but 3 of the substantive variables were included, namely the legal provisions considered by the court (*lawType*), the supplementary legal provision considered (*lawSupp*) as well as the minor legal provisions considered (*lawMinor*). This is because the broader variable of issue area (*issueArea*) is more valuable and to a large extent incorporates these variables.

Outcome variables

- Only the *partyWinning* variable is modelled as a univariate outcome.

Voting and opinion variables

- The majority of these variables are irrelevant for the purposes of this study and only provide details about the manner in which the judgment was given, for example, the majority opinion writer variable (*majOpinWriter*) specifies which judge was the author of the court's opinion or judgment.
- The only meaningful variable that could potentially be a determinant in the decision-making process is the majority variable (*majority*) which captures whether a judge voted with the majority or not. This may provide insight into how often certain judges vote with the majority or minority.

5.3 Missing data

There are many different approaches that can be used to handle missing data, the most common of which is to impute these missing values in some way. In some modelling techniques, missing values are not problematic, such as with a decision tree. In other instances, however, such as in a random forest, missing values are problematic [6].

Table 1 summarises the percentage of missing values per variable in the Supreme Court dataset used in this study. Only the missing values of variables that will be used, as discussed in the variable overview, are summarised in Table 1. Missing values in this context are of a different nature to the missing values discussed in the Titanic example. Missing values for many of the variables in this dataset are absent because of legal procedure. The jurisdiction of the court is of particular importance in this regard because if the Supreme Court is the court of first instance, for example, there will be no Lower Court Disposition (*lcDisposition*). This is not true for all variables *caseOriginState* and *caseSourceState* is particularly problematic with 87% of the missing values. These 2 variables will therefore be omitted as they do not contain any information (if an independent variable has constant x values, here NAs, it cannot be used to explain the dependent variable).

Variable	Frequency missing	Percentage missing
caseOrigin	200	2.05
caseOriginState	8521	87.16
caseSource	115	1.18
caseSourceState	8529	87.24
lcDisagreement	17	0.17
lcDisposition	463	4.74
lcDispositionDirection	88	0.90
issueArea	17	0.17
authorityDecision1	80	0.82
authorityDecision2	7710	78.87
lawType	243	2.49
majority	207	2.12

Table 3: Missing values per variable

The programme below codes the missing values as an extra category with a logical reasoning for why the data is missing. This coding may be quite effective as something like an uncommon *issueArea* may, in fact, be a decisive factor in a case.

```
1  two2$caseOrigin[is.na(two2$caseOrigin)] <- "SC original jurisdiction"
2  two2$caseSource[is.na(two2$caseSource)] <- "SC original jurisdiction"
3  two2$authorityDecision1[is.na(two2$authorityDecision1)] <- "no reason"
4  two2$authorityDecision2[is.na(two2$authorityDecision2)] <- "no second reason"
5  two2$lcDisagreement[is.na(two2$lcDisagreement)] <- "jurisdiction affects"
6  two2$lcDisposition[is.na(two2$lcDisposition)] <- "jurisdiction affects"
7  two2$lcDispositionDirection[is.na(two2$lcDispositionDirection)] <- "not specified/
       jurisdiction affects"
8  two2$issueArea[is.na(two2$issueArea)] <- "uncommon/not categorised"
9  two2$lawType[is.na(two2$lawType)] <- "uncommon/not categorised"
10 two2$majority[is.na(two2$majority)] <- "not given"
```

5.4 Prescreening of variables

In this section, prescreening of the data was completed in order to determine which variables are essential in building the final model. The large number of predictor variables poses a challenge for the interpretability in the modelling of the decision, since all predictor variables are categorical in nature. Logistic regression and a single decision tree were used to screen the variables for initial importance. This is used to reduce the dimensionality of the dataset. The initial investigation was also done to establish the extent to which the missing values will influence the random forest.

5.4.1 Decision tree

The coding below was used to construct a decision tree to determine the outcome of a case, using the Supreme Court Database.

```
1 fittree <- rpart(partyWinning ~ term + naturalCourt + jurisdiction +
      lcDisagreement + lcDisposition + lcDispositionDirection + issueArea +
      authorityDecision1 + authorityDecision2 + lawType + justiceName + majority +
      caseOrigin + caseSource + scQual + scIdeology, data=two2, method = "class",
      parms = list(split = 'gini'))
2 fancyRpartPlot(fittree, cex=0.53, suffix="\n\n", under.cex=0.53)
3 summary(fittree, file)
```

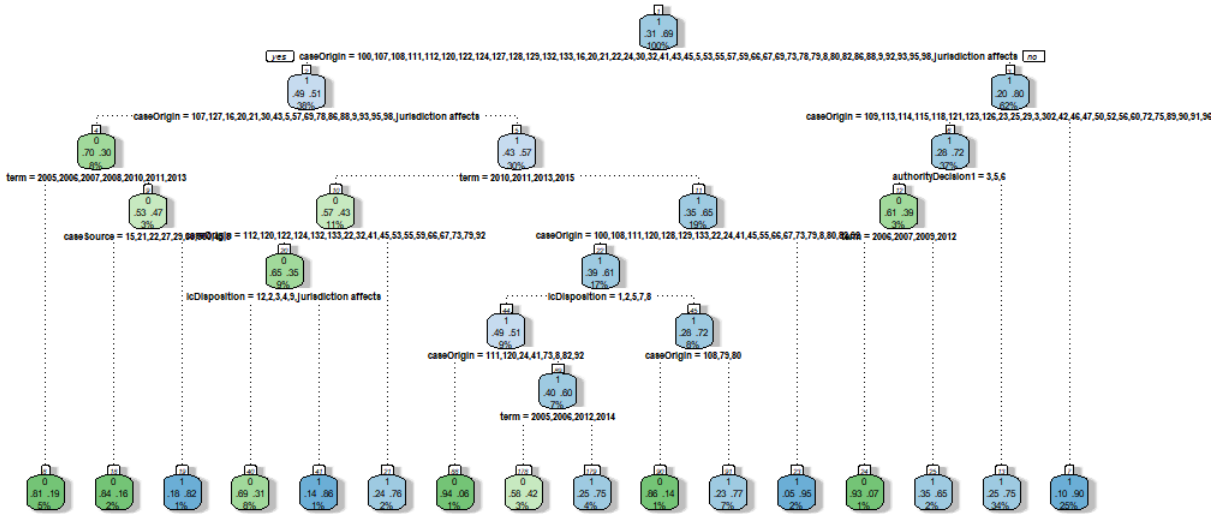


Figure 6: Classification tree generated using the Gini index as the splitting criteria

The advantage of a decision tree is the ease of interpretation. The interpretation is of which variables are important in the decision-making process and when these variables become important. From Figure 6 it can be seen that the significant variables are: *caseOrigin, term, caseSource, lcDisposition* and *authorityDecision1*. This result is promising in showing the impartiality of judges as it is clear that the *justiceName, scIdeology* and *scQual* are not significant variables. The fact that *authorityDecision1* is an important variable is significant as the judgment should indeed be based on what a judge claims the reasoning for a decision was, as captured in this variable. It is logical that the term is a significant splitting variable as this could be indicative of the development of the law; development of law is necessary in order to meet the ever changing needs and ideals of society, which a country's law should be representative of.

5.4.2 Logistic regression

Below is the code used in SAS to fit a logistic regression model to the data.[2]

```
1 proc logistic data=use;
2 class term naturalCourt justiceName;
3 model partyWinning = caseOrigin caseSource term naturalCourt jurisdiction
      lcDisagreement lcDisposition lcDispositionDirection issueArea
      authorityDecision1 authorityDecision2 lawType justiceName majority scQual
      scIdeology;
4 run;
```

The logistic regression model is given in the equation below.

$$\hat{f} = g(X_{caseOrigin}, X_{caseSource}, X_{term}, X_{naturalCourt}, X_{jurisdiction}, X_{lcDisagreement}, X_{lcDisposition}, X_{lcDispositionDirection}, X_{issueArea}, X_{authorityDecision1}, X_{authorityDecision2}, X_{lawType}, X_{justiceName}, X_{majority})$$

From the results the following variables were found to be significant (at a 5% level of significance): *caseOrigin, caseSource, term, lcDisposition, issueArea, authorityDecision2* and *lawType*. It is interesting to note that the conventional approach to the categorisation of a binary outcome variable, logistic regression, gives an error rate of 23.43%. The variables *scIdeology* and *scQual* were set to 0 as they are a linear combination of an intercept term and various *justiceNames*. *ScQual* and *scIdeology* were therefore excluded from the final model due to the multicollinearity between these variables and *justiceName*. The *naturalCourt* variable was not excluded from the final model as only one category was found to be a linear combination of other variables. From the 60 possible predictors the random forest is then developed using this subset of the 12 remaining predictors.

5.5 Final model for judicial decision-making

From the results of the prescreening, the following variables were included in the final model: *term, naturalCourt, jurisdiction, lcDisagreement, lcDisposition, lcDispositionDirection, issueArea, authorityDecision1, authorityDecision2, lawType, justiceName* and *majority*. During this process it was found that the ideology scores (Segal-Cover scores) and qualifications of the judges are a linear combination of the *justiceName* variable. It is because of this result that these two variables, *scIdeology* and *scQual*, were excluded from the model.

The following R code was used to construct a random forest classifier to model the outcome variable of whether a judicial decision is favourable or not.

[2]This analysis was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2017 SAS Institute Inc., Cary, NC, USA.

```
forest <- randomForest(partyWinning ~ term + naturalCourt + jurisdiction +
    lcDisagreement + lcDisposition + lcDispositionDirection + issueArea +
    authorityDecision1 + authorityDecision2 + lawType + justiceName + majority,
    data = two2, importance=TRUE, ntree=2000, replace=TRUE)
varImpPlot(forest)
importance(forest)
print(forest)
```

Hereafter the final model, a random forest classifier, was constructed which yielded the following results:

	0	1	classification error
0	2929	120	0.03935717
1	114	6613	0.01694663

Table 4: Confusion matrix when predicting *partyWinning*

The estimated OOB error rate for the dataset using the developed random forest classifier is 2.39%. This error rate is very small indicating exceptional performance in predicting the outcome of a U.S. Supreme Court case. The aim of this study is not only to predict the outcome accurately but to investigate which factors are most important in this decision. For this purpose, a visual representation of the variable importance is given in Figure 7. The figure displays the mean decrease in accuracy based on a specific variable on the left, and the mean decrease in the Gini criterion on the right. It is noted that the higher a variable is on these plots, the more important it is in predicting the response. It can be seen that the two most important variables are identical and the top seven differ only in ordering but comprise of the same subset of variables. From Figure 7 it is evident that the most important predictors are: *term, issueArea, authorityDecision1, lcDisposition, caseOrigin, caseSource* and *lawType*. While the three least important predictors are: *justiceName, jurisdiction* and *majority*.

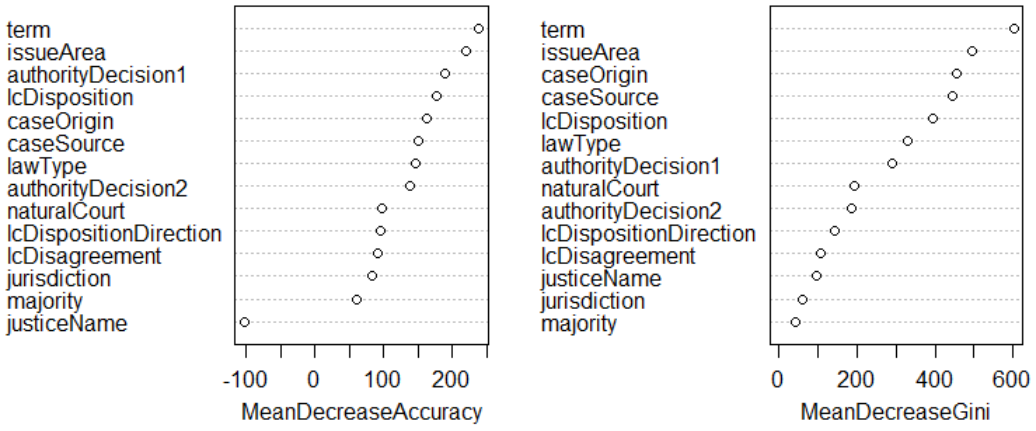


Figure 7: Variable importance in predicting *partyWinning*

This proves the objective of the study, although not definitively, that judges are impartial in the judicial decision-making process. This is evident from the variable *justiceName* being one of the least important variables. The variable *majority* is often analysed as it could be indicative of certain judges always voting with the majority, it is also positive to see that this does not appear to be the case.

It is interesting to note that the variable *term* is the most important variable as this could be indicative of the legal development over time. The fact that the *issueArea* is a very important variable is also a good indication of judicial impartiality, as the type of legal issue dealt with should be a determinant in the judicial decision-making process. The *authorityDecision* variables are the reason(s) given for the decision made. It is therefore expected that this variable should be important, as it would indicate a judge's accuracy in capturing the reasoning for the judicial decision made. The fact that the *lcDisposition* is in the top 5 most important variables is interesting. It is interesting as this could be indicative of the Supreme Court's lack of willingness to disagree with lower courts, which could be worrisome. This, however, could also be interpreted as an indication of the quality of the lower courts. This is perhaps a more realistic interpretation as the lack of willingness to disagree with lower courts does not take into account the fact that the Supreme Court may not be willing to change the lower courts disposition because the lower court made a sound conclusion; which is not problematic. The *caseSource* and *caseOrigin* variables may also be indicative of the quality of courts within certain jurisdiction, as well as the differences between ideals in the different circuits. This is expected as the United States that has both Federal and State laws which enable certain areas to differ in their legal systems. This is in line with certain jurisprudential theories that the law should be reflective of societies ideals which can vary greatly based on geographical location.

5.6 Other issues encountered

An issue encountered with the data was that the variables *caseOrigin* and *caseSource* have 211 distinct values which the randomForest package was unable to deal with. Consequently the variables were grouped into new logical categories, according to the various court groupings made in the United States. The 211 distinct values were coded into 11 distinct circuits, the District of Columbia, the Federal Circuit and 5 other courts; this coding can be seen in the Appendix. The missing values, as mentioned earlier, were replaced by logical reasoning for their absence. For the variables *caseOrigin* and *caseSource* the reason for missing data is that the Supreme Court has original jurisdiction. This makes up the last category of the variables as seen in Figure 13 and Figure 14 in the Appendix.

There was a problem with the case of Ivan Eberhart vs United States. This case was coded as a number that did not exist according to the codebook provided by the Supreme Court Database. After further investigation of the Supreme Court Database website, it was found that this case was heard in

the Illinois Middle U.S. District Court which forms part of the Eleventh Circuit, so it was coded as such.

Another issue encountered in the application process was that R could not handle the NA values in the object (the edited Supreme Court dataset), even though these variables were not being made use of in the final model. These missing values were coded (see the Appendix) as an extra category which solved this issue and randomForest package was able to work.

The coding according to codebook, while useful, was not easily interpreted. The coded variables were therefore formatted back to worded categories that can be easily interpreted, see the Appendix for this coding.

6 Conclusion

In an ideal legal system judicial decisions made by judges are independent of judge's personal ideology, political affiliation, and other external factors; judges should be impartial. This independence is often called into question to ensure the principle of separation of powers is adhered to. The purpose of this study was to investigate this independence using cases from the United States Supreme Court, as captured by the Supreme Court Database. To this end, a random forest classification model was used to predict the outcome of a court case, and to investigate the most influential predictors of the outcome of a case.

The justice's political ideology was represented by the variable *justiceName* which could also be interchanged with the Segal-Cover Scores of the judges, denoted in the Supreme Court Database by the variable *scIdeology*. The impartiality of judges could be indicated by this variable as it is one of the least influential predictors of the outcome in the judicial decision-making process.

From the developed model it is found that the most important predictors of the outcome are: *term, issueArea, authorityDecision1, lcDisposition, caseOrigin, caseSource* and *lawType*; while the three least important are: *justiceName*, *jurisdiction* and *majority*. The impartiality of judges is evident from the variable *justiceName* being one of the least important variables. The fact the *issueArea* is an important variable is also a good indication of judicial impartiality, as the type of legal issue dealt with should be a determinant in the judicial decision-making process. The *authorityDecision* variables are the reason given for the decision made. It is therefore expected that this variable should be important, as it would indicate a judge's accuracy in capturing the reasoning for the judicial decision made. It is encouraging that this was found to be the case.

It can therefore be concluded that the determinants of the judicial decision-making process are independent of the judge's personal ideology, political affiliation, and other external factors. This study has found the judges in the Supreme Court of the United States to be impartial in the judicial decision-making process.

7 Recommendations

The potential for development of this study is vast. A few ideas for future development are mentioned below.

Theoretical:

- From a theoretical point of view, a recommendation would be to make use of new developments in random forests to construct the classifier.
- There have been several extensions of random forests over the years. One such technique uses the McNemar nonparametric significance test in an attempt to decrease the number of trees that contribute to the majority vote, without a decrease in the accuracy of the model. Another development is weighted random sampling, in place of simple random sampling, during the selection of features at each node. Two more recent developments are: the genetic algorithm-based random forest (GARF) and the hybrid weighted random forest algorithm [5].

Application:

- Future research could be done by testing the random forest built in this study on the cases heard in 2017. This data will, however, only be published in July 2018.
- Alternatively, the data used to develop the random forest could have excluded the cases heard in 2016. The 2016 cases could have then been used to develop a "toy example", as an illustration of the functioning of the model developed on cases from 2005 until the end of 2015. This seems redundant due to the out of bag data that serves a similar purpose, however, it may be valuable for illustrative purposes.

Alternative programmes/software packages:

- The debate around which programming language or software is the "best" analytical tool is multi-faceted. It would be interesting to consider various other programming languages in the construction and application of random forests. Alternative programming languages and software that could be used in future studies to develop random forests, are listed below.
  - SAS® Enterprise Miner has a procedure called PROC HPFOREST that can be used to construct a random forest
  - The scikit-learn library in Python has a RandomForestClassifier function
  - Random Forests™ is a software trade marked by Adele Cutler and Leo Breiman [6]

References

[1] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[2] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] Robert Bryll, Ricardo Gutierrez-Osuna, and Francis Quek. Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition*, 36(6):1291–1302, 2003.

[4] Antonio Criminisi, Jamie Shotton, and Ender Konukoglu. Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, 7(2–3):81–227, 2012.

[5] Khaled Fawagreh, Mohamed M Gaber, and Eyad Elyan. Random forests: from early developments to recent advancements. *Systems Science & Control Engineering: An Open Access Journal*, 2(1):602–609, 2014.

[6] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer Series in Statistics, 2001.

[7] James L Gibson. Challenges to the impartiality of state supreme courts: legitimacy theory and 'new-style' judicial campaigns. *American Political Science Review*, 102(1):59–75, 2008.

[8] Daniel M Katz, Michael J Bommarito II, and Josh Blackman. A general approach for predicting the behavior of the Supreme Court of the United States. *Plos One*, 12(4):e0174698, 2017.

[9] Raja B Koushik and Sharan K Ravindran. *R Data Science Essentials*. Packt Publishing Ltd, 2016.

[10] Andy Liaw and Matthew Wiener. Classification and regression by randomForest. *R News*, 2(3):18–22, 2002.

[11] Tom M Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[12] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[13] Lior Rokach and Oded Maimon. Decision trees. *Data Mining and Knowledge Discovery Handbook*, 20:165–192, 2005.

[14] Theodore W Ruger, Pauline T Kim, Andrew D Martin, and Kevin M Quinn. The Supreme Court forecasting project: legal and political science approaches to predicting Supreme Court decisionmaking. *Columbia Law Review*, 104(4):1150–1210, 2004.

[15] Harold J Spaeth, Lee Epstein, Andrew D Martin, Jeffrey A Segal, Theodore J Ruger, and Sara C Benesh. Supreme Court Database Version 2016 Release 01, 2016.

[16] Xiaogang Su, Andres Azuero, June Cho, Elizabeth Kvale, Karen M Meneses, and Patrick McNees. An introduction to tree-structured modeling with application to quality of life (qol) data. *Nursing Research*, 60(4):247, 2011.

Appendix

7.1 Descriptive statistics

Descriptive statistics were done on the raw dataset, however, only a few will be explained in this section. For a record of the descriptive statistics not included in this section, see Section 7.2 and Section 7.3. Please refer to the following link to access the Supreme Court database codebook to see details about the coding of every variable: `http://scdb.wustl.edu/_brickFiles/2016_01/SCDB_2016_01_codebook.pdf`.

7.1.1 Winning party

The *partyWinning* variable captures whether a person appealing a lower court's judgment was successful in their appeal or not. From the output below, it can be seen that there are more than double the number of favourable outcomes as opposed to outcomes that are not favourable.

The FREQ Procedure

partyWinning				
partyWinning	Frequency	Percent	Cumulative Frequency	Cumulative Percent
no favorable disposition for petitioning party apparent	3049	31.19	3049	31.19
petitioning party received a favorable disposition	6727	68.81	9776	100.00

Chi-Square Test for Equal Proportions	
Chi-Square	1383.7647
DF	1
Pr > ChiSq	<.0001

Sample Size = 9776

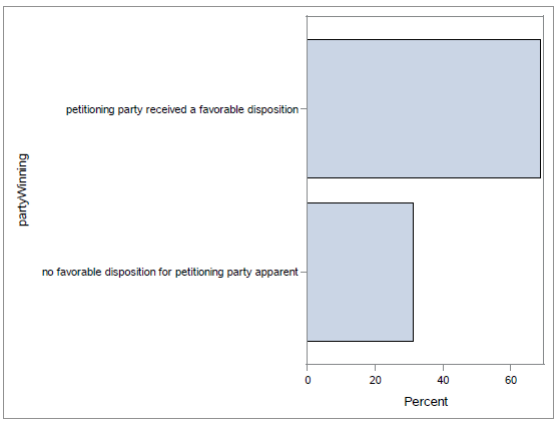


Figure 8: Percent of each category present in the variable winning party

7.1.2 Law type

This variable captures the legislation, constitutional provisions or court rules that the court takes into consideration in a case.

The FREQ Procedure

lawType				
lawType	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Constitution	632	6.63	632	6.63
Constitutional Amendment	2165	22.71	2797	29.34
Federal Statute	3145	32.99	5942	62.33
Court Rules	321	3.37	6263	65.70
Other	171	1.79	6434	67.49
Infrequently litigated statutes	2655	27.85	9089	95.34
State or local law or regulation	117	1.23	9206	96.57
No Legal Provision	327	3.43	9533	100.00
Frequency Missing = 243				

Chi-Square Test for Equal Proportions	
Chi-Square	9163.8207
DF	7
Pr > ChiSq	<.0001

Effective Sample Size = 9533
Frequency Missing = 243

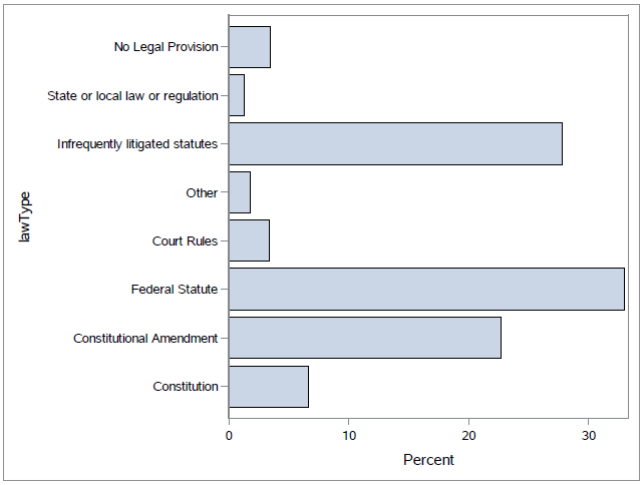


Figure 9: Percent of each category present in the variable law type

7.1.3 Issue area

This variable identifies which area of law is being considered in a particular case.

The FREQ Procedure

issueArea				
issueArea	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Criminal Procedure	2568	26.31	2568	26.31
Civil Rights	1574	16.13	4142	42.44
First Amendment	657	6.73	4799	49.18
Due Process	233	2.39	5032	51.56
Privacy	206	2.11	5238	53.67
Attorneys	179	1.83	5417	55.51
Unions	222	2.27	5639	57.78
Economic Activity	2035	20.85	7674	78.64
Judicial Power	1307	13.39	8981	92.03
Federalism	527	5.40	9508	97.43
Interstate Relations	62	0.64	9570	98.06
Federal Taxation	108	1.11	9678	99.17
Miscellaneous	63	0.65	9741	99.82
Private Action	18	0.18	9759	100.00
Frequency Missing = 17				

Chi-Square Test for Equal Proportions	
Chi-Square	12948.5768
DF	13
Pr > ChiSq	<.0001

Effective Sample Size = 9759
Frequency Missing = 17

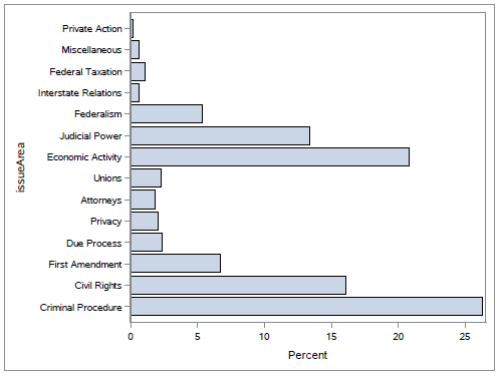


Figure 10: Percent of each category present in the variable issue area

7.1.4 Lower court disposition

This variable specifies what the lower court's outcome is which is being reviewed in the Supreme Court.

The FREQ Procedure

lcDisposition				
lcDisposition	Frequency	Percent	Cumulative Frequency	Cumulative Percent
stay, petition, or motion granted	135	1.45	135	1.45
affirmed	4947	53.12	5082	54.57
reversed	1728	18.55	6810	73.12
reversed and remanded	762	8.18	7572	81.31
vacated and remanded	444	4.77	8016	86.07
affirmed and reversed (or vacated) in part	198	2.13	8214	88.20
affirmed and reversed (or vacated) in part and remanded	224	2.41	8438	90.60
vacated	204	2.19	8642	92.80
petition denied or appeal dismissed	591	6.35	9233	99.14
remand	54	0.58	9287	99.72
unusual disposition	26	0.28	9313	100.00
Frequency Missing = 463				

Chi-Square Test for Equal Proportions	
Chi-Square	24631.5224
DF	10
Pr > ChiSq	<.0001

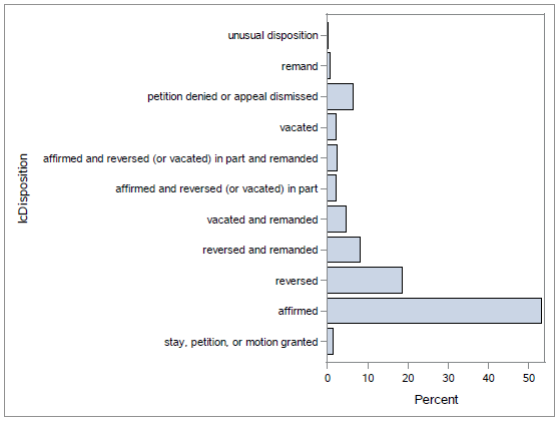


Figure 11: Percent of each category present in the variable lower court disposition

7.1.5 Authority for decisions

This variable identifies the legal basis on which a reason was taken. Some decisions have more than one legal basis and this is captured by the second authority decision variable.

The FREQ Procedure

authorityDecision1				
authorityDecision1	Frequency	Percent	Cumulative Frequency	Cumulative Percent
judicial review (national level)	1030	10.62	1030	10.62
judicial review (state level)	1835	18.93	2865	29.55
SC supervision of lower federal/state courts/original jurisdiction	550	5.67	3415	35.22
statutory construction	5224	53.88	8639	89.10
interpretation of administrative regulation/rule/executive order	314	3.24	8953	92.34
diversity jurisdiction	8	0.08	8961	92.42
federal common law	735	7.58	9696	100.00
Frequency Missing = 80				

Chi-Square Test for Equal Proportions	
Chi-Square	13882.5687
DF	6
Pr > ChiSq	<.0001

Effective Sample Size = 9696
Frequency Missing = 80

authorityDecision2				
authorityDecision2	Frequency	Percent	Cumulative Frequency	Cumulative Percent
judicial review (national level)	145	7.02	145	7.02
judicial review (state level)	90	4.36	235	11.37
SC supervision of lower federal/state courts/original jurisdiction	78	3.78	313	15.15
statutory construction	784	37.95	1097	53.10
interpretation of administrative regulation/rule/executive order	205	9.92	1302	63.02
federal common law	764	36.98	2066	100.00
Frequency Missing = 7710				

Chi-Square Test for Equal Proportions	
Chi-Square	1638.5092
DF	5
Pr > ChiSq	<.0001

Effective Sample Size = 2066
Frequency Missing = 7710

WARNING: 79% of the data are missing.

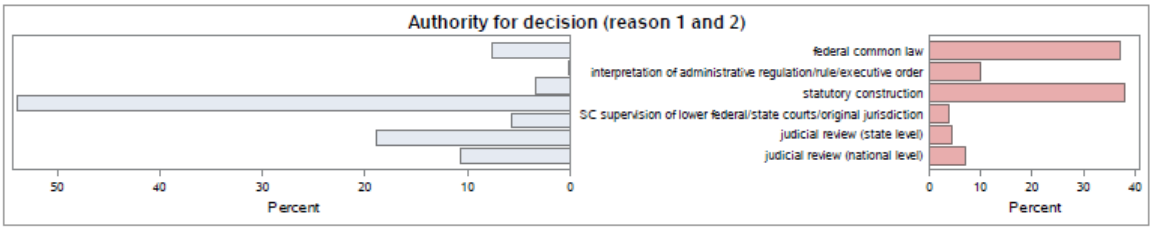


Figure 12: Percent of each category present in the authority decision variable

The graph on the left of Figure12 summarises the first reason given and the graph on the right summaries the second reason, if given.

7.1.6 Case origin

This variable indicates which lower court the case originated from.

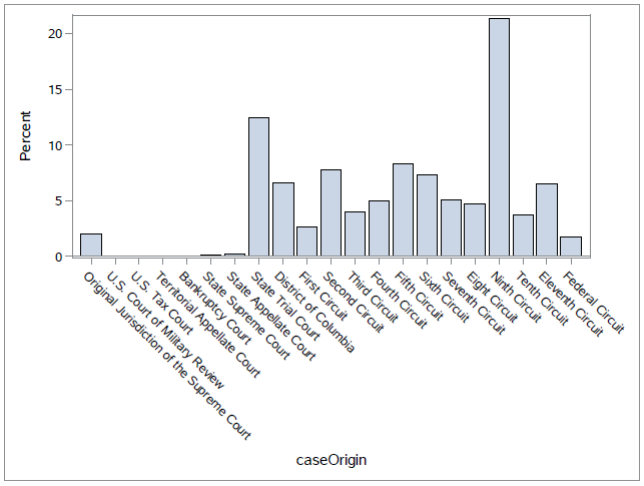


Figure 13: Percent of each category present in the case origin variable

7.1.7 Case source

This variable indicates which lower court's decision the Supreme Court is reviewing.

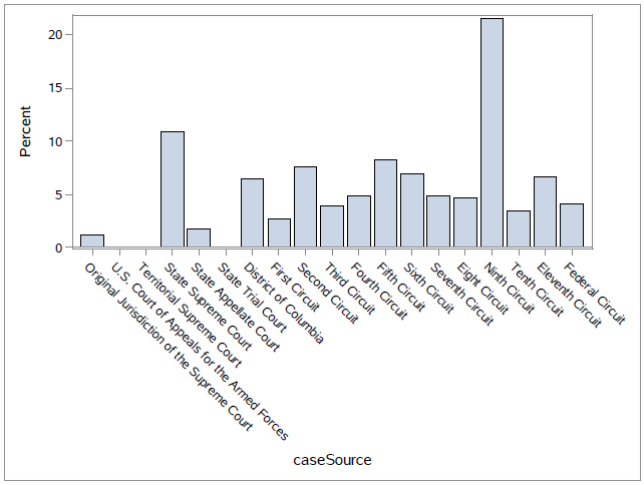


Figure 14: Percent of each category present in the case source variable

7.2 Descriptive statistics using R

The summary function is used to get a general overview of the data, but more specifically to see where missing values are problematic. Note that there were two unlabelled cases that were removed before this

investigation. Kansas vs Carr originally had two different dockets at a lower court level but they were consolidated at the Supreme Court level. The second unlabelled case was U.S. Airways vs McCutchen that was an "unclear" outcome that could not be coded into a favorable or unfavorable disposition.

```
two<-subset('2.00', chief=='Roberts')
two2<-subset(two, docket!='14-450'& docket!='11-1285')
summary(two2)
```

```
     caseId           docketId           caseIssuesId                    voteId        dateDecision   decisionType
2011-077:  81   2009-077-01:  36   2005-001-01-01:   9   2005-001-01-01-01-01:   1   2010-06-21: 108   Min.   :1.000
2005-084:  72   2009-077-02:  36   2005-002-01-01:   9   2005-001-01-01-01-02:   1   2006-06-28:  99   1st Qu.:1.000
2009-077:  72   2005-014-02:  27   2005-003-01-01:   9   2005-001-01-01-01-03:   1   2007-06-25:  99   Median :1.000
2015-063:  56   2005-039-01:  27   2005-004-01-01:   9   2005-001-01-01-01-04:   1   2012-06-28:  99   Mean   :1.474
2013-063:  54   2008-058-01:  27   2005-004-01-02:   9   2005-001-01-01-01-05:   1   2016-05-16:  96   3rd Qu.:1.000
2005-014:  36   2009-088-01:  27   2005-005-01-01:   9   2005-001-01-01-01-06:   1   2008-06-12:  81   Max.   :7.000
(Other) :9432   (Other)    :9623   (Other)       :9749   (Other)              :9797   (Other)   :9221
     term       naturalCourt        chief           docket
Min.   :2005   Min.   :1701   Burger   :   0   08-1498 :  36
1st Qu.:2007   1st Qu.:1702   Rehnquist:   0   09-89   :  36
Median :2010   Median :1704   Roberts  :9803   04-1152 :  27
Mean   :2010   Mean   :1703   Vinson   :   0   04-1236 :  27
3rd Qu.:2012   3rd Qu.:1704   Warren   :   0   07-10374:  27
Max.   :2015   Max.   :1705                    08-861  :  27
                                               (Other) :9623
                                                                                                                    caseName
ERIC H. HOLDER, JR., ATTORNEY GENERAL, et al. v. HUMANITARIAN LAW PROJECT et al.                                          :  36
HUMANITARIAN LAW PROJECT, et al. v. ERIC H. HOLDER, JR., ATTORNEY GENERAL, et al.                                         :  36
AMERICAN ELECTRIC POWER COMPANY, INC., et al., PETITIONERS v. CONNECTICUT ET AL                                           :  27
DEPARTMENT OF HEALTH AND HUMAN SERVICES, et al., PETITIONERS v. FLORIDA et al.                                            :  27
DONALD H. RUMSFELD, SECRETARY OF DEFENSE, et al. v. FORUM FOR ACADEMIC AND INSTITUTIONAL RIGHTS, INC., et al.:  27
FLORIDA, et al., PETITIONERS v. DEPARTMENT OF HEALTH AND HUMAN SERVICES, et al.                                           :  27
(Other)                                                                                                                   :9623
    dateArgument       dateRearg      petitioner    petitionerState   respondent    respondentState  jurisdiction
           : 853              :9740   Min.   :  1.0   Min.   : 1.00   Min.   :  1.0   Min.   : 1.00   Min.   : 1.000
2006-03-01:  99   2009-09-09:  18   1st Qu.: 28.0   1st Qu.: 6.00   1st Qu.: 27.0   1st Qu.:10.00   1st Qu.: 1.000
2010-02-23:  81   2006-03-21:   9   Median :136.0   Median :25.00   Median :130.0   Median :23.00   Median : 1.000
2012-03-26:  81   2006-04-25:   9   Mean   :131.4   Mean   :25.93   Mean   :127.8   Mean   :27.01   Mean   : 1.178
2016-03-23:  56   2006-05-18:   9   3rd Qu.:195.0   3rd Qu.:41.00   3rd Qu.:188.0   3rd Qu.:45.00   3rd Qu.: 1.000
2007-04-17:  54   2012-10-01:   9   Max.   :600.0   Max.   :60.00   Max.   :600.0   Max.   :60.00   Max.   :12.000
(Other)   :8579   (Other)    :   9   NA's   :26      NA's   :7552    NA's   :9       NA's   :7528


 adminAction     adminActionState threeJudgeFdc       caseOrigin    caseOriginState   caseSource     caseSourceState
Min.   :  4.00   Min.   : 4.00    Min.   :0.00000   Min.   :  3.0   Min.   : 1.00   Min.   :  4.00   Min.   : 1.00
1st Qu.: 28.00   1st Qu.: 8.00    1st Qu.:0.00000   1st Qu.: 51.0   1st Qu.:12.00   1st Qu.: 25.00   1st Qu.:12.00
Median : 40.00   Median :25.00    Median :0.00000   Median : 81.0   Median :25.00   Median : 29.00   Median :23.00
Mean   : 54.12   Mean   :24.49    Mean   :0.03024   Mean   :104.6   Mean   :27.24   Mean   : 62.91   Mean   :26.78
3rd Qu.: 76.00   3rd Qu.:38.25    3rd Qu.:0.00000   3rd Qu.:119.0   3rd Qu.:43.00   3rd Qu.: 31.00   3rd Qu.:41.00
Max.   :117.00   Max.   :55.00    Max.   :1.00000   Max.   :302.0   Max.   :57.00   Max.   :302.00   Max.   :57.00
NA's   :7844     NA's   :9519     NA's   :81        NA's   :200     NA's   :8539    NA's   :115      NA's   :8547
lcDisagreement     certReason     lcDisposition    lcDispositionDirection declarationUncon caseDisposition
Min.   :0.0000   Min.   : 1.000   Min.   : 1.000   Min.   :1.000          Min.   :1.000    Min.   :1.000
1st Qu.:0.0000   1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.:1.000          1st Qu.:1.000    1st Qu.:2.000
Median :0.0000   Median :11.000   Median : 2.000   Median :2.000          Median :1.000    Median :4.000
Mean   :0.2629   Mean   : 8.342   Mean   : 3.346   Mean   :1.533          Mean   :1.125    Mean   :3.789
3rd Qu.:1.0000   3rd Qu.:12.000   3rd Qu.: 4.000   3rd Qu.:2.000          3rd Qu.:1.000    3rd Qu.:4.000
Max.   :1.0000   Max.   :13.000   Max.   :12.000   Max.   :3.000          Max.   :4.000    Max.   :9.000
NA's   :17       NA's   :99       NA's   :463      NA's   :88                              NA's   :54
caseDispositionUnusual  partyWinning    precedentAlteration  voteUnclear     issue          issueArea
Min.   :0.0000000      Min.   :0.0000   Min.   :0.00000     Min.   :0    Min.   : 10010   Min.   : 1.000
1st Qu.:0.0000000      1st Qu.:0.0000   1st Qu.:0.00000     1st Qu.:0    1st Qu.: 10570   1st Qu.: 1.000
Median :0.0000000      Median :1.0000   Median :0.00000     Median :0    Median : 40010   Median : 4.000
Mean   :0.0009181      Mean   :0.6905   Mean   :0.02479     Mean   :0    Mean   : 50024   Mean   : 4.986
3rd Qu.:0.0000000      3rd Qu.:1.0000   3rd Qu.:0.00000     3rd Qu.:0    3rd Qu.: 80180   3rd Qu.: 8.000
Max.   :1.0000000      Max.   :2.0000   Max.   :1.00000     Max.   :0    Max.   :140070   Max.   :14.000
                       NA's   :9                                         NA's   :17       NA's   :17
decisionDirection decisionDirectionDissent authorityDecision1 authorityDecision2    lawType         lawSupp
Min.   :1.000     Min.   :0.00000          Min.   :1.00       Min.   :1.000      Min.   :1.000   Min.   :104
1st Qu.:1.000     1st Qu.:0.00000          1st Qu.:2.00       1st Qu.:4.000      1st Qu.:2.000   1st Qu.:230
Median :1.000     Median :0.00000          Median :4.00       Median :4.000      Median :3.000   Median :341
Mean   :1.494     Mean   :0.00271          Mean   :3.51       Mean   :4.873      Mean   :3.809   Mean   :399
3rd Qu.:2.000     3rd Qu.:0.00000          3rd Qu.:4.00       3rd Qu.:7.000      3rd Qu.:6.000   3rd Qu.:600
Max.   :3.000     Max.   :1.00000          Max.   :7.00       Max.   :7.000      Max.   :9.000   Max.   :900
NA's   :17        NA's   :193              NA's   :80         NA's   :7737       NA's   :243     NA's   :243
```

```
          lawMinor    majOpinWriter   majOpinAssigner   splitVote    majVotes        minVotes        justice
              :1922   Min.   :103.0   Min.   :103.0   Min.   :1   Min.   :4.000   Min.   :0.000   Min.   :103
18-924        :  63   1st Qu.:106.0   1st Qu.:111.0   1st Qu.:1   1st Qu.:5.000   1st Qu.:0.000   1st Qu.:106
9-1           :  63   Median :109.0   Median :111.0   Median :1   Median :7.000   Median :2.000   Median :109
18 U.S.C. § 924:  45   Mean   :108.8   Mean   :110.2   Mean   :1   Mean   :7.129   Mean   :1.683   Mean   :109
18-922        :  36   3rd Qu.:111.0   3rd Qu.:111.0   3rd Qu.:1   3rd Qu.:9.000   3rd Qu.:3.000   3rd Qu.:111
(Other)       :2394   Max.   :114.0   Max.   :111.0   Max.   :1   Max.   :9.000   Max.   :4.000   Max.   :114
NA's          :5280   NA's   :1293    NA's   :417
    justiceName        vote          opinion         direction       majority     firstAgreement   secondAgreement
AMKennedy :1098   Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :  0.00   Min.   :  0.00
CThomas   :1098   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.00   1st Qu.:2.000   1st Qu.:103.00   1st Qu.:  0.00
RBGinsburg:1098   Median :1.000   Median :1.000   Median :1.00   Median :2.000   Median :108.00   Median :  0.00
JGRoberts :1097   Mean   :1.511   Mean   :1.259   Mean   :1.49   Mean   :1.812   Mean   : 86.44   Mean   : 19.96
SGBreyer  :1097   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.00   3rd Qu.:2.000   3rd Qu.:110.00   3rd Qu.:  0.00
SAAlito   :1066   Max.   :8.000   Max.   :3.000   Max.   :2.00   Max.   :2.000   Max.   :114.00   Max.   :114.00
(Other)   :3249   NA's   :131     NA's   :135     NA's   :321    NA's   :207     NA's   :8013     NA's   :9323
       scQual          scIdeology      X               X.1             X.2             X.3
1         :2152   0,159999996:1098   Mode:logical   Mode:logical   Mode:logical   Mode:logical
0,810000002:1762   0,36500001 :1098   NA's:9803      NA's:9803      NA's:9803      NA's:9803
0,414999992:1098   0,680000007:1098
0,889999986:1098   0,119999997:1097
0,545000017:1097   0,474999994:1097
0,970000029:1097   0,100000001:1066
(Other)    :1499   (Other)    :3249
```

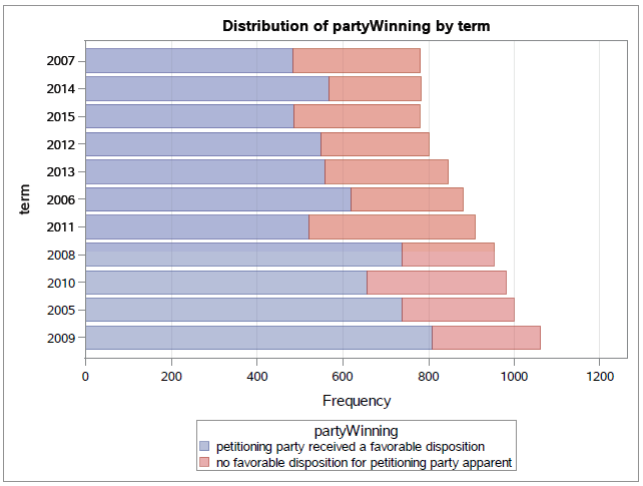
7.3 Descriptive statistics using SAS

The above dataset modifications made in R are also made in SAS before an overview is conducted. Two-way frequency tables of the most important variables (according to the final model) are shown here.

```
data Rsame;
set sasuser.scdb;
if chief ^= "Roberts" then delete;
if partyWinning = 2 then delete;
if docket = 11-1285 then delete;
if docket = 14-450 then delete;
run;
data use;
set two;                          *see Section 7.5 used to create this data;
format partyWinning partyWinningfmt.  authorityDecision1 authorityDecision1fmt.
    authorityDecision2 authorityDecision2fmt. issueArea issueAreafmt. jurisdiction
    jurisdictionfmt. lcDisagreement lcDisagreementfmt. lcDisposition lcDispositionfmt.
    lcDispositionDirection lcDispositionDirectionfmt. majority majorityfmt. caseOrigin
    caseOriginfmt.  caseSource caseOriginfmt.;
run;
proc freq data=use order=freq;
tables partyWinning*term partyWinning*caseOrigin partyWinning*caseSource partyWinning*
    issueArea partyWinning*lcDisposition partyWinning*lawType partyWinning*
    authorityDecision1 partyWinning*authorityDecision2  /chisq plots=freqplot(twoway=
    stacked orient=horizontal);
run;
```

Winning party by term

Frequency Percent Row Pct Col Pct	Table of partyWinning by term											
	term(term)											
partyWinning(partyWinning)	2009	2005	2010	2008	2011	2006	2013	2012	2015	2014	2007	Total
petitioning party received a favorable disposition	810 8.29 12.04 76.27	738 7.55 10.97 73.87	657 6.72 9.77 66.97	738 7.55 10.97 77.36	522 5.34 7.76 57.43	619 6.33 9.20 70.34	558 5.71 8.29 65.96	549 5.62 8.16 68.54	485 4.96 7.21 62.18	567 5.80 8.43 72.41	484 4.95 7.19 61.97	6727 68.81
no favorable disposition for petitioning party apparent	252 2.58 8.27 23.73	261 2.67 8.56 26.13	324 3.31 10.63 33.03	216 2.21 7.08 22.64	387 3.96 12.69 42.57	261 2.67 8.56 29.66	288 2.95 9.45 34.04	252 2.58 8.27 31.46	295 3.02 9.68 37.82	216 2.21 7.08 27.59	297 3.04 9.74 38.03	3049 31.19
Total	1062 10.86	999 10.22	981 10.03	954 9.76	909 9.30	880 9.00	846 8.65	801 8.19	780 7.98	783 8.01	781 7.99	9776 100.00
Frequency Missing = 9												



Statistics for Table of partyWinning by term

Statistic	DF	Value	Prob
Chi-Square	10	170.3338	<.0001
Likelihood Ratio Chi-Square	10	170.1898	<.0001
Mantel-Haenszel Chi-Square	1	25.2205	<.0001
Phi Coefficient		0.1320	
Contingency Coefficient		0.1309	
Cramer's V		0.1320	

Effective Sample Size = 9776
Frequency Missing = 9

Winning party by case origin

Frequency
Percent
Row Pct
Col Pct

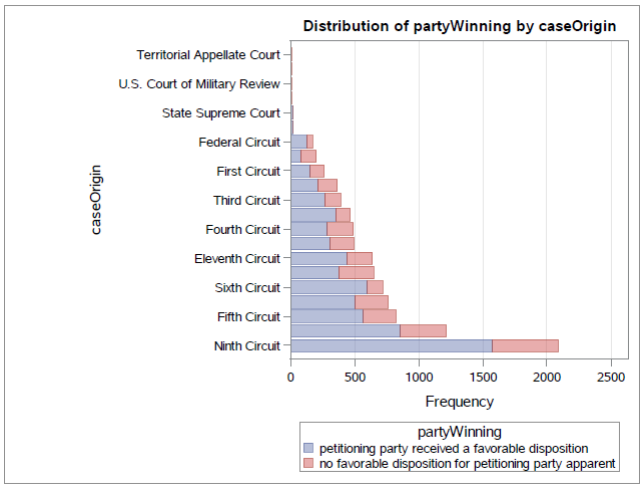
Table of partyWinning by caseOrigin									
	caseOrigin(caseOrigin)								
partyWinning(partyWinning)	Ninth Circuit	State Trial Court	Fifth Circuit	Second Circuit	Sixth Circuit	District of Columbia	Eleventh Circuit	Seventh Circuit	Fourth Circuit
petitioning party received a favorable disposition	1576 16.12 23.43 75.55	857 8.77 12.74 70.59	570 5.83 8.47 69.51	502 5.14 7.46 66.05	600 6.14 8.92 83.80	376 3.85 5.59 58.20	439 4.49 6.53 68.92	306 3.13 4.55 61.82	286 2.93 4.25 58.61
no favorable disposition for petitioning party apparent	510 5.22 16.73 24.45	357 3.65 11.71 29.41	250 2.56 8.20 30.49	258 2.64 8.46 33.95	116 1.19 3.80 16.20	270 2.76 8.86 41.80	198 2.03 6.49 31.08	189 1.93 6.20 38.18	202 2.07 6.63 41.39
Total	2086 21.34	1214 12.42	820 8.39	760 7.77	716 7.32	646 6.61	637 6.52	495 5.06	488 4.99
Frequency Missing = 9									

Frequency
Percent
Row Pct
Col Pct

Table of partyWinning by caseOrigin								
	caseOrigin(caseOrigin)							
partyWinning(partyWinning)	Eight Circuit	Third Circuit	Tenth Circuit	First Circuit	Original Jurisdiction of the Supreme Court	Federal Circuit	State Appellate Court	State Supreme Court
petitioning party received a favorable disposition	349 3.57 5.19 75.54	267 2.73 3.97 68.11	212 2.17 3.15 58.24	151 1.54 2.24 58.53	78 0.80 1.16 39.00	125 1.28 1.86 73.53	17 0.17 0.25 100.00	16 0.16 0.24 100.00
no favorable disposition for petitioning party apparent	113 1.16 3.71 24.46	125 1.28 4.10 31.89	152 1.55 4.99 41.76	107 1.09 3.51 41.47	122 1.25 4.00 61.00	45 0.46 1.48 26.47	0 0.00 0.00 0.00	0 0.00 0.00 0.00
Total	462 4.73	392 4.01	364 3.72	258 2.64	200 2.05	170 1.74	17 0.17	16 0.16
Frequency Missing = 9								

Frequency
Percent
Row Pct
Col Pct

Table of partyWinning by caseOrigin					
	caseOrigin(caseOrigin)				
partyWinning(partyWinning)	Bankruptcy Court	U.S. Court of Military Review	U.S. Tax Court	Territorial Appellate Court	Total
petitioning party received a favorable disposition	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	6727 68.81
no favorable disposition for petitioning party apparent	9 0.09 0.30 100.00	9 0.09 0.30 100.00	9 0.09 0.30 100.00	8 0.08 0.26 100.00	3049 31.19
Total	9 0.09	9 0.09	9 0.09	8 0.08	9776 100.00
Frequency Missing = 9					



Statistics for Table of partyWinning by caseOrigin

Statistic	DF	Value	Prob
Chi-Square	20	410.8640	<.0001
Likelihood Ratio Chi-Square	20	424.5111	<.0001
Mantel-Haenszel Chi-Square	1	22.7614	<.0001
Phi Coefficient		0.2050	
Contingency Coefficient		0.2008	
Cramer's V		0.2050	

Effective Sample Size = 9776
Frequency Missing = 9

Winning party by case source

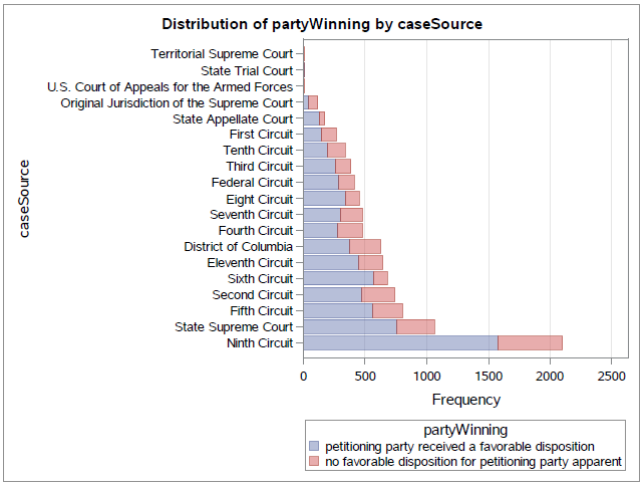
Frequency
Percent
Row Pct
Col Pct

Table of partyWinning by caseSource

partyWinning(partyWinning)	caseSource(caseSource)									
	Ninth Circuit	State Supreme Court	Fifth Circuit	Second Circuit	Sixth Circuit	Eleventh Circuit	District of Columbia	Fourth Circuit	Seventh Circuit	Eight Circuit
petitioning party received a favorable disposition	1576 16.12 23.43 74.94	756 7.73 11.24 71.32	561 5.74 8.34 69.95	475 4.86 7.06 64.10	565 5.78 8.40 82.97	448 4.58 6.66 69.35	376 3.85 5.59 59.87	277 2.83 4.12 57.83	297 3.04 4.42 62.26	340 3.48 5.05 75.06
no favorable disposition for petitioning party apparent	527 5.39 17.28 25.06	304 3.11 9.97 28.68	241 2.47 7.90 30.05	266 2.72 8.72 35.90	116 1.19 3.80 17.03	198 2.03 6.49 30.65	252 2.58 8.27 40.13	202 2.07 6.63 42.17	180 1.84 5.90 37.74	113 1.16 3.71 24.94
Total	2103 21.51	1060 10.84	802 8.20	741 7.58	681 6.97	646 6.61	628 6.42	479 4.90	477 4.88	453 4.63

Frequency Missing = 9

Table of partyWinning by caseSource										
	caseSource(caseSource)									
partyWinning(partyWinning)	Federal Circuit	Third Circuit	Tenth Circuit	First Circuit	State Appellate Court	Original Jurisdiction of the Supreme Court	U.S. Court of Appeals for the Armed Forces	State Trial Court	Territorial Supreme Court	Total
petitioning party received a favorable disposition	285 2.92 4.24 69.68	258 2.64 3.84 67.36	194 1.98 2.88 57.57	142 1.45 2.11 53.38	126 1.29 1.87 73.68	43 0.44 0.64 37.39	0 0.00 0.00 0.00	8 0.08 0.12 100.00	0 0.00 0.00 0.00	6727 68.81
no favorable disposition for petitioning party apparent	124 1.27 4.07 30.32	125 1.28 4.10 32.64	143 1.46 4.69 42.43	124 1.27 4.07 46.62	45 0.46 1.48 26.32	72 0.74 2.36 62.61	9 0.09 0.30 100.00	0 0.00 0.00 0.00	8 0.08 0.26 100.00	3049 31.19
Total	409 4.18	383 3.92	337 3.45	266 2.72	171 1.75	115 1.18	9 0.09	8 0.08	8 0.08	9776 100.00
Frequency Missing = 9										



Statistics for Table of partyWinning by caseSource

Statistic	DF	Value	Prob
Chi-Square	18	325.5843	<.0001
Likelihood Ratio Chi-Square	18	328.6455	<.0001
Mantel-Haenszel Chi-Square	1	3.1054	0.0780
Phi Coefficient		0.1825	
Contingency Coefficient		0.1795	
Cramer's V		0.1825	

Effective Sample Size = 9776
Frequency Missing = 9

Winning party by issue area

Frequency
Percent
Row Pct
Col Pct

Table of partyWinning by issueArea

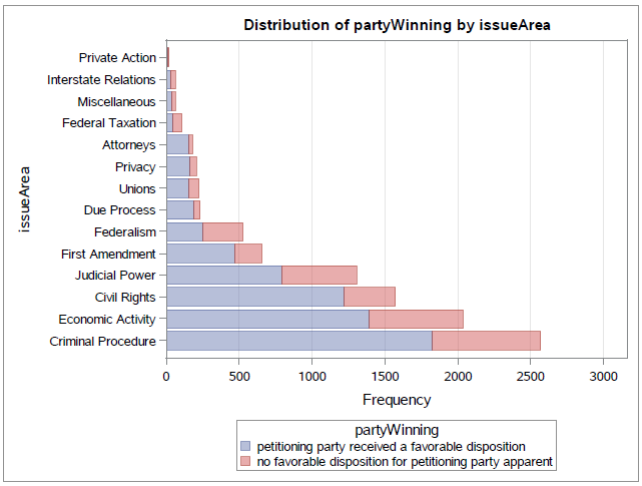
partyWinning(partyWinning)	issueArea(issueArea) Criminal Procedure	Economic Activity	Civil Rights	Judicial Power	First Amendment	Federalism	Due Process
petitioning party received a favorable disposition	1827 18.72 27.20 71.14	1393 14.27 20.74 68.45	1218 12.48 18.13 77.38	790 8.10 11.76 60.44	469 4.81 6.98 71.39	252 2.58 3.75 47.82	188 1.93 2.80 80.69
no favorable disposition for petitioning party apparent	741 7.59 24.37 28.86	642 6.58 21.11 31.55	356 3.65 11.71 22.62	517 5.30 17.00 39.56	188 1.93 6.18 28.61	275 2.82 9.04 52.18	45 0.46 1.48 19.31
Total	2568 26.31	2035 20.85	1574 16.13	1307 13.39	657 6.73	527 5.40	233 2.39

Frequency Missing = 26

Table of partyWinning by issueArea

partyWinning(partyWinning)	issueArea(issueArea) Unions	Privacy	Attorneys	Federal Taxation	Miscellaneous	Interstate Relations	Private Action	Total
petitioning party received a favorable disposition	152 1.56 2.26 68.47	161 1.65 2.40 78.16	152 1.56 2.26 84.92	45 0.46 0.67 41.67	36 0.37 0.54 57.14	26 0.27 0.39 41.94	9 0.09 0.13 50.00	6718 68.84
no favorable disposition for petitioning party apparent	70 0.72 2.30 31.53	45 0.46 1.48 21.84	27 0.28 0.89 15.08	63 0.65 2.07 58.33	27 0.28 0.89 42.86	36 0.37 1.18 58.06	9 0.09 0.30 50.00	3041 31.16
Total	222 2.27	206 2.11	179 1.83	108 1.11	63 0.65	62 0.64	18 0.18	9759 100.00

Frequency Missing = 26



Statistics for Table of partyWinning by issueArea

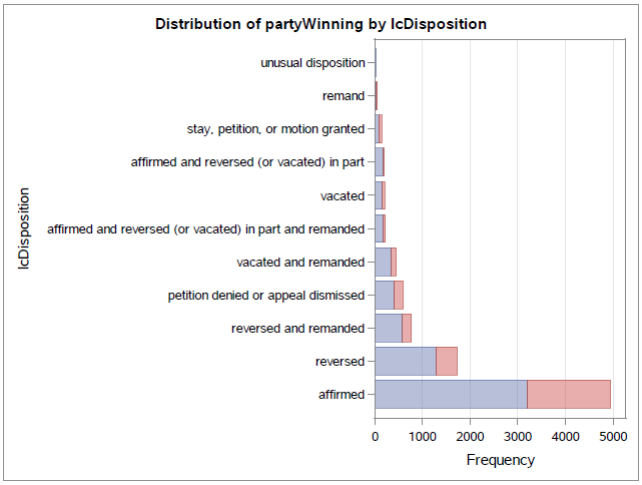
Statistic	DF	Value	Prob
Chi-Square	13	323.8108	<.0001
Likelihood Ratio Chi-Square	13	316.1602	<.0001
Mantel-Haenszel Chi-Square	1	162.7324	<.0001
Phi Coefficient		0.1822	
Contingency Coefficient		0.1792	
Cramer's V		0.1822	

Effective Sample Size = 9759
Frequency Missing = 26

Winning party by lower court disposition

Frequency Percent Row Pct Col Pct	Table of partyWinning by lcDisposition					
	lcDisposition(lcDisposition)					
partyWinning(partyWinning)	affirmed	reversed	reversed and remanded	petition denied or appeal dismissed	vacated and remanded	affirmed and reversed (or vacated) in part and remanded
petitioning party received a favorable disposition	3201 34.37 49.64 64.71	1271 13.65 19.71 73.55	584 6.27 9.06 76.64	394 4.23 6.11 66.67	338 3.63 5.24 76.13	188 2.02 2.92 83.93
no favorable disposition for petitioning party apparent	1746 18.75 60.96 35.29	457 4.91 15.96 26.45	178 1.91 6.22 23.36	197 2.12 6.88 33.33	106 1.14 3.70 23.87	36 0.39 1.26 16.07
Total	4947 53.12	1728 18.55	762 8.18	591 6.35	444 4.77	224 2.41
Frequency Missing = 472						

Table of partyWinning by lcDisposition						
	lcDisposition(lcDisposition)					
partyWinning(partyWinning)	vacated	affirmed and reversed (or vacated) in part	stay, petition, or motion granted	remand	unusual disposition	Total
petitioning party received a favorable disposition	150 1.61 2.33 73.53	180 1.93 2.79 90.91	90 0.97 1.40 66.67	36 0.39 0.56 66.67	17 0.18 0.26 65.38	6449 69.25
no favorable disposition for petitioning party apparent	54 0.58 1.89 26.47	18 0.19 0.63 9.09	45 0.48 1.57 33.33	18 0.19 0.63 33.33	9 0.10 0.31 34.62	2864 30.75
Total	204 2.19	198 2.13	135 1.45	54 0.58	26 0.28	9313 100.00
Frequency Missing = 472						



Statistics for Table of partyWinning by lcDisposition

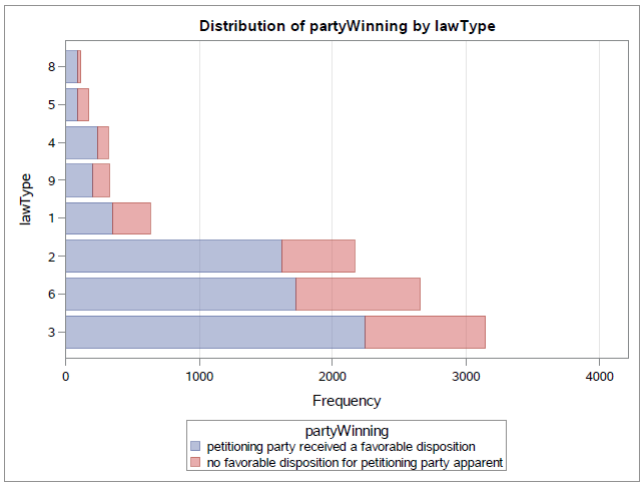
Statistic	DF	Value	Prob
Chi-Square	10	163.0583	<.0001
Likelihood Ratio Chi-Square	10	177.2087	<.0001
Mantel-Haenszel Chi-Square	1	31.0776	<.0001
Phi Coefficient		0.1323	
Contingency Coefficient		0.1312	
Cramer's V		0.1323	

Effective Sample Size = 9313
Frequency Missing = 472

Winning party by law type

Frequency
Percent
Row Pct
Col Pct

Table of partyWinning by lawType									
	lawType(lawType)								
partyWinning(partyWinning)	3	6	2	1	9	4	5	8	Total
petitioning party received a favorable disposition	2245 23.55 34.24 71.38	1724 18.08 26.30 64.93	1614 16.93 24.62 74.55	348 3.65 5.31 55.06	204 2.14 3.11 62.39	241 2.53 3.68 75.08	90 0.94 1.37 52.63	90 0.94 1.37 76.92	6556 68.77
no favorable disposition for petitioning party apparent	900 9.44 30.23 28.62	931 9.77 31.27 35.07	551 5.78 18.51 25.45	284 2.98 9.54 44.94	123 1.29 4.13 37.61	80 0.84 2.69 24.92	81 0.85 2.72 47.37	27 0.28 0.91 23.08	2977 31.23
Total	3145 32.99	2655 27.85	2165 22.71	632 6.63	327 3.43	321 3.37	171 1.79	117 1.23	9533 100.00
Frequency Missing = 252									



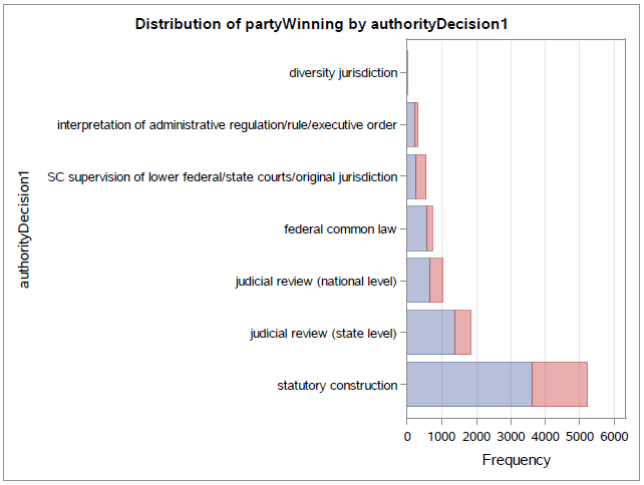
Statistics for Table of partyWinning by lawType

Statistic	DF	Value	Prob
Chi-Square	7	153.6651	<.0001
Likelihood Ratio Chi-Square	7	150.2952	<.0001
Mantel-Haenszel Chi-Square	1	16.7796	<.0001
Phi Coefficient		0.1270	
Contingency Coefficient		0.1260	
Cramer's V		0.1270	

Effective Sample Size = 9533
Frequency Missing = 252

Winning party by authority for decisions

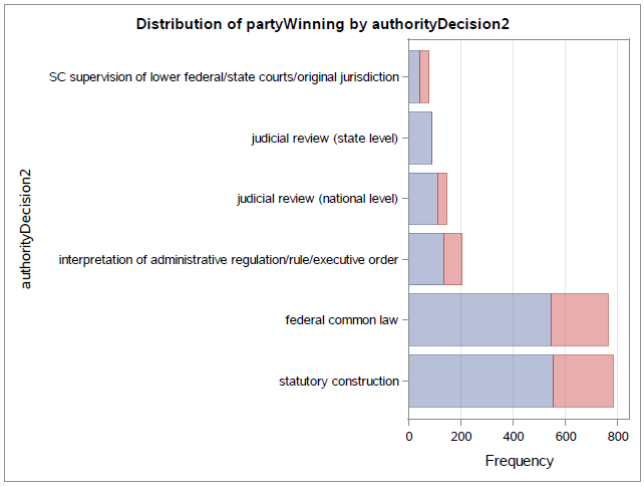
Table of partyWinning by authorityDecision1			
	authorityDecision1(authorityDecision1)		
partyWinning(partyWinning)	interpretation of administrative regulation/rule/executive order	diversity jurisdiction	Total
petitioning party received a favorable disposition	216 2.23 3.24 68.79	0 0.00 0.00 0.00	6664 68.73
no favorable disposition for petitioning party apparent	98 1.01 3.23 31.21	8 0.08 0.26 100.00	3032 31.27
Total	314 3.24	8 0.08	9696 100.00
Frequency Missing = 89			



Statistics for Table of partyWinning by authorityDecision1

Statistic	DF	Value	Prob
Chi-Square	6	273.4716	<.0001
Likelihood Ratio Chi-Square	6	261.0623	<.0001
Mantel-Haenszel Chi-Square	1	14.0965	0.0002
Phi Coefficient		0.1679	
Contingency Coefficient		0.1656	
Cramer's V		0.1679	

Effective Sample Size = 9696
Frequency Missing = 89



Statistics for Table of partyWinning by authorityDecision2

Statistic	DF	Value	Prob
Chi-Square	5	51.4492	<.0001
Likelihood Ratio Chi-Square	5	75.1264	<.0001
Mantel-Haenszel Chi-Square	1	3.7925	0.0515
Phi Coefficient		0.1578	
Contingency Coefficient		0.1559	
Cramer's V		0.1578	

Effective Sample Size = 2066
Frequency Missing = 7719

WARNING: 79% of the data are missing.

7.4 Conversion of variables

The 'as.factor' and 'as.numeric' functions in R were used to convert variables to character or numerical values where necessary. Note that even variables that were not used in the final model, were converted to the correct variable type so that the summary function could be an accurate representation of the data.

```
two2$majority <-as.factor(two2$majority)
two2$scQual <-as.numeric(two2$scQual)
```

7.4.1 Descriptive statistics post dealing with missing values and conversions

After missing values and conversion of variables were dealt with, a summary of the data was done using R.

```
summary(two2)
```

```
     caseId           docketId           caseIssuesId
2011-077:  81   2009-077-01:  36   2005-001-01-01:   9
2005-084:  72   2009-077-02:  36   2005-002-01-01:   9
2009-077:  72   2005-014-02:  27   2005-003-01-01:   9
2015-063:  56   2005-039-01:  27   2005-004-01-01:   9
2013-063:  54   2008-058-01:  27   2005-004-01-02:   9
2005-014:  36   2009-088-01:  27   2005-005-01-01:   9
(Other) :9405   (Other)    :9596   (Other)       :9722
                 voteId         dateDecision  decisionType
2005-001-01-01-01-01:   1   2010-06-21: 108   1:8232
2005-001-01-01-01-02:   1   2006-06-28:  99   2: 819
2005-001-01-01-01-03:   1   2007-06-25:  99   4:  26
2005-001-01-01-01-04:   1   2012-06-28:  99   5:  77
2005-001-01-01-01-05:   1   2016-05-16:  96   6: 290
2005-001-01-01-01-06:   1   2008-06-12:  81   7: 332
(Other)             :9770   (Other)   :9194
     term      naturalCourt       chief           docket
2009   :1062   1701: 288    Burger   :   0   08-1498 :  36
2005   : 999   1702:3326    Rehnquist:   0   09-89   :  36
2010   : 981   1703:1044    Roberts  :9776   04-1152 :  27
2008   : 954   1704:4518    Vinson   :   0   04-1236 :  27
2011   : 909   1705: 600    Warren   :   0   07-10374:  27
2006   : 880                                 08-861  :  27
(Other):3991                                 (Other) :9596

                  caseName
ERIC H. HOLDER, JR., ATTORNEY GENERAL, et al. v. HUMANITARIAN LAW PROJECT et al.
                        :  36
HUMANITARIAN LAW PROJECT, et al. v. ERIC H. HOLDER, JR., ATTORNEY GENERAL, et al.
                        :  36
AMERICAN ELECTRIC POWER COMPANY, INC., et al., PETITIONERS v. CONNECTICUT ET AL
                        :  27
DEPARTMENT OF HEALTH AND HUMAN SERVICES, et al., PETITIONERS v. FLORIDA et al.
                        :  27
DONALD H. RUMSFELD, SECRETARY OF DEFENSE, et al. v. FORUM FOR ACADEMIC AND INSTITUTION
AL RIGHTS, INC., et al.:  27
```


```
     caseId           docketId           caseIssuesId
2011-077:  81   2009-077-01:  36   2005-001-01-01:   9
2005-084:  72   2009-077-02:  36   2005-002-01-01:   9
2009-077:  72   2005-014-02:  27   2005-003-01-01:   9
2015-063:  56   2005-039-01:  27   2005-004-01-01:   9
2013-063:  54   2008-058-01:  27   2005-004-01-02:   9
2005-014:  36   2009-088-01:  27   2005-005-01-01:   9
(Other) :9405   (Other)    :9596   (Other)       :9722
                 voteId         dateDecision  decisionType
2005-001-01-01-01-01:   1   2010-06-21: 108   1:8232
2005-001-01-01-01-02:   1   2006-06-28:  99   2: 819
2005-001-01-01-01-03:   1   2007-06-25:  99   4:  26
2005-001-01-01-01-04:   1   2012-06-28:  99   5:  77
2005-001-01-01-01-05:   1   2016-05-16:  96   6: 290
2005-001-01-01-01-06:   1   2008-06-12:  81   7: 332
(Other)             :9770   (Other)   :9194
     term      naturalCourt       chief           docket
2009   :1062   1701: 288    Burger   :   0   08-1498 :  36
2005   : 999   1702:3326    Rehnquist:   0   09-89   :  36
2010   : 981   1703:1044    Roberts  :9776   04-1152 :  27
2008   : 954   1704:4518    Vinson   :   0   04-1236 :  27
2011   : 909   1705: 600    Warren   :   0   07-10374:  27
2006   : 880                                 08-861  :  27
(Other):3991                                 (Other) :9596

                  caseName
ERIC H. HOLDER, JR., ATTORNEY GENERAL, et al. v. HUMANITARIAN LAW PROJECT et al.
                        :  36
HUMANITARIAN LAW PROJECT, et al. v. ERIC H. HOLDER, JR., ATTORNEY GENERAL, et al.
                        :  36
AMERICAN ELECTRIC POWER COMPANY, INC., et al., PETITIONERS v. CONNECTICUT ET AL
                        :  27
DEPARTMENT OF HEALTH AND HUMAN SERVICES, et al., PETITIONERS v. FLORIDA et al.
                        :  27
DONALD H. RUMSFELD, SECRETARY OF DEFENSE, et al. v. FORUM FOR ACADEMIC AND INSTITUTION
AL RIGHTS, INC., et al.:  27
```

```
             lcDisagreement   certReason                lcDisposition
0                   :7186   12     :4231   2                   :4947
1                   :2573   2      :2567   3                   :1728
jurisdiction affects:  17   11     :1764   4                   : 762
                            1      : 446   9                   : 591
                            10     : 197   jurisdiction affects: 463
                            (Other): 472   5                   : 444
                            NA's   :  99   (Other)             : 841
                       lcDispositionDirection declarationUncon
1                                  :4735       1:9039
2                                  :4755       2: 324
3                                  : 198       3: 341
not specified/jurisdiction affects:  88       4:  72


caseDisposition caseDispositionUnusual partyWinning precedentAlteration
4      :3622    0:9767                 0:3049       0:9533
2      :2533    1:   9                 1:6727       1: 243
5      :1604
3      :1154
7      : 315
(Other): 494
NA's   :  54
voteUnclear     issue        issueArea    decisionDirection
0:9776      10020  : 430   1      :2568   1   :5041
            80130  : 313   8      :2035   2   :4611
            100030 : 268   2      :1574   3   : 107
            10050  : 256   9      :1307   NA's:  17
            10560  : 251   3      : 657
            (Other):8241   10     : 527
            NA's   :  17   (Other):1108


 decisionDirectionDissent authorityDecision1        authorityDecision2
 0   :9557                4      :5224       1               : 145
 1   :  26                2      :1835       2               :  90
 NA's: 193                1      :1030       3               :  78
                          7      : 735       4               : 784
                          3      : 550       5               : 205
                          5      : 314       7               : 764
                          (Other):  88       no second reason:7710
    lawType        lawSupp                lawMinor    majOpinWriter
 3      :3145   600    :2655                  :1922   105    :1060
 6      :2655   341    : 432   18-924         :  63   111    :1056
 2      :2165   200    : 377   9-1            :  63   106    :1054
 1      : 632   205    : 355   18 U.S.C. § 924:  45   109    : 920
 9      : 327   900    : 327   18-922         :  36   108    : 903
 4      : 321   (Other):5387   (Other)        :2394   (Other):3490
 (Other): 531   NA's   : 243   NA's           :5253   NA's   :1293
 majOpinAssigner splitVote majVotes minVotes    justice
 103 : 504       1:9776    4:  93   0:4175   106    :1095
 105 : 196                 5:2572   1: 703   108    :1095
 106 : 462                 6:1324   2:1311   109    :1095
 108 :  36                 7:1117   3:1251   110    :1094
 111 :8161                 8:1304   4:2336   111    :1094
 NA's: 417                 9:3366            112    :1063
                                             (Other):3240
     justiceName        vote      opinion     direction
 AMKennedy :1095   1      :6723   1   :7187   1   :4818
 CThomas   :1095   2      :1799   2   :2410   2   :4637
 RBGinsburg:1095   3      : 573   3   :  44   NA's: 321
 JGRoberts :1094   4      : 432   NA's: 135
 SGBreyer  :1094   8      :  72
 SAAlito   :1063   (Other):  46
 (Other)   :3240   NA's   : 131


      majority    firstAgreement secondAgreement     scQual
 1        :1799   0      : 358   0      : 383    Min.   : 2.000
 2        :7770   110    : 239   105    :  17    1st Qu.: 4.000
 not given: 207   109    : 173   108    :  14    Median : 6.000
                  105    : 172   111    :  13    Mean   : 6.873
                  111    : 151   112    :  12    3rd Qu.:10.000
                  (Other): 682   (Other):  32    Max.   :11.000
                  NA's   :8001   NA's   :9305
   scIdeology        X             X.1            X.2
 Min.   : 2.000   Mode:logical   Mode:logical   Mode:logical
 1st Qu.: 4.000   NA's:9776      NA's:9776      NA's:9776
 Median : 8.000
 Mean   : 7.617
 3rd Qu.:11.000
 Max.   :14.000
```

7.5 Fixing other data issues encountered

The code below was used to group together categories within the *caseSource* and *caseOrigin* variables to form 21 new categories. This was done to form every new category, however, only the coding to create the category *First Circuit* is shown below. The code below was also used to convert missing values in the data to a new category called *Original Jurisdiction*.

```
data two;
set Rsame;
*CaseOrigin;
*First Circuit;
if caseOrigin=78 then caseOrigin=1111; *Maine;
if caseOrigin=416 then caseOrigin=1111;
if caseOrigin=80 then caseOrigin=1111; *Massachusetts;
if caseOrigin=418 then caseOrigin=1111;
if caseOrigin=91 then caseOrigin=1111; *New Hampshire;
if caseOrigin=424 then caseOrigin=1111;
if caseOrigin=112 then caseOrigin=1111; *Puerto Rico;
if caseOrigin=113 then caseOrigin=1111; *Rhode Island;
if caseOrigin=431 then caseOrigin=1111;
if caseOrigin=21 then caseOrigin=1111; *First Circuit;
*CaseSource;
*First Circuit;
if caseSource=78 then caseSource=1111; *Maine;
if caseSource=416 then caseSource=1111;
if caseSource=80 then caseSource=1111; *Massachusetts;
if caseSource=418 then caseSource=1111;
if caseSource=91 then caseSource=1111; *New Hampshire;
if caseSource=424 then caseSource=1111;
if caseSource=112 then caseSource=1111; *Puerto Rico;
if caseSource=113 then caseSource=1111; *Rhode Island;
if caseSource=431 then caseSource=1111;
if caseSource=21 then caseSource=1111; *First Circuit;
*Missing Values;
if caseOrigin=. then caseOrigin=00000; *Original Jurisdiction;
if caseSource=. then caseSource=00000; *Original Jurisdiction;
run;
```

The following code was used to format the coded variables back to their categorical names, as given by the Supreme Court Database codebook. This is only shown below for the variable *partyWinning*.

```
proc format;
value partyWinningfmt
```

```
        0="unfavorable disposition for petitioner"
        1="favorable disposition for petitioner";
run;
data use;
set two;
format partyWinning partyWinningfmt.;
run;
```

Minimum information for training a classifier

Catherine Amber Halsey 14348587

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr F Kanfer

Department of Statistics, University of Pretoria

October 29, 2017

Abstract

Classifier accuracy is extremely important and can be improved by increasing the size of the sample data on which the classifier is based. However, in experimental/laboratory cases it is not always possible to obtain enough of the required data to train the classifier, as even very large data sets may not contain enough information and access to the information may become extremely computer intensive.

For this reason a sequential method of training classifiers can be of use. This method, which is based on certain stopping criteria and evaluates the classification rule at each step is able to ensure with a certain level of confidence that the probability of the classifier making an error is within a pre-specified level of the absolute minimum feasible error, the Bayes error, whilst only requiring the smallest possible number of observations.

Keywords: Bayes error, fixed-width confidence interval, classifier training

Declaration

I, *Catherine Amber Halsey,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Catherine Amber Halsey

_ _

Dr F Kanfer

_ _

30 October 2017

Acknowledgments

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

Contents

List of Figures

List of Tables

1 Introduction

Classifiers are used to assign various observations to certain classes based on the characteristics or attributes of the particular observation or to predict the classes of new data based on what is currently available. This is typically done through a training process whereby a certain classification rule is applied to historical data to train the classifier. Once this has been done, the classifier can be used for the clustering of future observations.

Depending on what the classifier is used for, it can become and in most cases is extremely important for the decision made by the classifier to be correct. In finance for example, erroneous classification can lead to defaulting parties which in turn can lead to the loss of incredibly large amounts of money, or in the medical field, incorrect diagnosis of certain diseases/disorders can prove fatal.

The accuracy can be improved by increasing the sample size of the data used to train the classifier. In practice however, this can prove difficult as factors such as time, affordability, information availability and level of computer intensity can come into play. Therefore it is ideal to as accurately as possible train a classifier using the least amount of observations.

This study will explore the use of a sequential training procedure in which an algorithm, based on a derived stopping criteria, is used to update the classification rule following each sequential step until it can be guaranteed with a prescribed level of confidence that the probability of an incorrect classification is within a certain pre-specified level of the absolute minimum feasible error [5].

Fu *et al.* [2] previously proposed a sequential training approach in which a stopping rule is applied once an observation has been classified after each sequential step, this process trains a classifier that ensures with a predetermined $(1-\alpha)\%$ level of confidence that future observations will have a misclassification probability that is less than a chosen upper limit $\varepsilon > 0$. Through a simulation study the general behavior of the sequential procedure is depicted in Fu *et al.* [2] and it is observed that the resulting misclassification rates of the sequentially trained classifiers are similar to that of the Bayes error. A problem arises however when the pre-specified misclassification rate, ε, is smaller than the Bayes error since this is the minimum feasible misclassification rate. This can often occur as the Bayes error is frequently unknown, when this happens the sequential procedure suggested continues indefinitely [5].

In order to train a classifier to estimate the proportion of misclassified observations it is first necessary to determine a fixed-width confidence interval. Frey [1] suggests a sequential method of obtaining such intervals for a proportion by proposing four different stopping criteria as well as providing the critical values necessary in their application. These methods lead to confidence intervals which ensure a coverage probability of $(1-\alpha)$ or more while requiring the least number of observations. By applying these methods in simulation studies the stopping rule based on an adapted version of the Wald confidence interval for a proportion is found to yield the best results [2].

To address the problem of specified misclassification rates being unfeasibly low this idea is used by Potgieter [5] to propose a sequential method of training a classifier to estimate the Bayes error and ensure that the rate of misclassification remains within a pre-specified level of this error. This is done by applying the Wald confidence interval [1] and repeatedly sampling, training and classifying observations until the misclassification rate lies within a fixed range of the optimum feasible error rate i.e the Bayes error [5].

In this essay various simulation studies are conducted, first to demonstrate the ideas developed in Frey [1] and then to imitate the simulation study conducted in Potgieter [5] which compares the observed rate of misclassification to the Bayes error and shows that the Bayes error is never larger than the misclassification rate, as such the sequential procedure suggested never continues indefinitely, it can also be noted that the classifier is never over trained and only the smallest possible number of observations is required to train it [5].

This method of classifier training is ideal as it gives the researcher more control over the process by specifying when the sequential procedure should be stopped, it is also not restricted to any single method of classification due to the constant updating of classification rules at each step, as such this method can be applied to LDA, QDA and KNN classification [5].

2 A general sequential approach of training a classifier

2.1 Classification methods

2.1.1 Linear and Quadratic discriminant analyses

To obtain ideal classification results it is necessary to have knowledge of the class posteriors , i.e. to know $P(Class\ A|X = \underline{x})$ [4].

Let the class-conditional density of X in class $A = k$ be denoted by $f_k(\underline{x})$ and the prior probability of class k be denoted by π_k where $\sum_{k=1}^{K} \pi_k = 1$, applying the Bayes Theorem gives

$$P(A = k|X = \underline{x}) = \frac{f_k(\underline{x})\pi_k}{\sum_{j=1}^{K} f_j(\underline{x})\pi_j} \tag{1}$$

Modeling each class density as a multivariate normal distribution gives

$$f_k(\underline{x}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\mathbf{\Sigma}_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu}_k)'\mathbf{\Sigma}_k(\underline{x}-\underline{\mu}_k)} \tag{2}$$

where $\underline{\mu}_k : p \times 1$ and $\mathbf{\Sigma}_k : p \times p$ are the population mean and covariance of class k respectively [4].

Linear Discriminant Analysis (LDA) considers the case where $\mathbf{\Sigma}_k = \mathbf{\Sigma} \forall k$. When comparing any two classes, say class k to class l Hastie *et al.* [4] concluded that it is satisfactory to use the log ratio of their posteriors

$$\log \frac{P(A=k|X=\underline{x})}{P(A=l|X=\underline{x})} = \log \left\{ \frac{\frac{f_k(\underline{x})\pi_k}{\sum_{j=1}^{K} f_j(\underline{x})\pi_j}}{\frac{f_l(\underline{x})\pi_l}{\sum_{j=1}^{K} f_j(\underline{x})\pi_j}} \right\} = \log \frac{f_k(\underline{x})\pi_k}{f_l(\underline{x})\pi_l} = \log \frac{\pi_k}{\pi_l} - \frac{1}{2}(\underline{\mu}_k + \underline{\mu}_l)'\mathbf{\Sigma}^{-1}(\underline{\mu}_k - \underline{\mu}_l) + \underline{x}'\mathbf{\Sigma}^{-1}(\underline{\mu}_k - \underline{\mu}_l) \tag{3}$$

which follows from equations 1 and 2. The linear discriminant functions can therefore be written as $\delta_k(\underline{x}) = \log \pi_k - \frac{1}{2}\underline{\mu}_k'\mathbf{\Sigma}^{-1}\underline{\mu}_k + \underline{x}'\mathbf{\Sigma}^{-1}\underline{\mu}_k$ which is linear in $\underline{x}$ [4].

If however the assumption that $\mathbf{\Sigma}_k = \mathbf{\Sigma} \forall k$ is not made the linear equation as given in equation 3 can not be obtained, thus resulting in Quadratic Discriminant Analysis (QDA) with discriminant functions defined as $\delta_k(\underline{x}) = \log \pi_k - \frac{1}{2}\log|\mathbf{\Sigma}_k| - \frac{1}{2}(\underline{x} - \underline{\mu}_k)'\mathbf{\Sigma}_k^{-1}(\underline{x} - \underline{\mu}_k)$.

In both cases an observation will be classified as a member of class l if $\delta_l(\underline{x}) > \delta_k(\underline{x})$ [4].

2.1.2 K-nearest neighbours

K-nearest neighbours (K-NN) uses a measure of distance such as Euclidean distance (shown in equation 4) to determine the k "nearest" observations to that of the observation under consideration.

$$d(\underline{a}, \underline{b}) = \sqrt{(a_1 - b_1)^2 + \ldots + (a_p - b_p)^2} = \|\underline{a} - \underline{b}\| \text{ where } \underline{a} \text{ and } \underline{b} \text{ are vectors} \tag{4}$$

If Ψ is defined as the set of K observations that are nearest to the input observation then the K-NN classifier is given by $\hat{Y} = \frac{1}{K}\sum_{y_i \epsilon \Psi} y_i$ [5].

If $y_i \epsilon \{0, 1\}$, implying that $0 \leq \hat{Y} \leq 1$, the input observation will be classified as a 1 if $\hat{Y} \geq 0.5$ and as a 0 if $\hat{Y} < 0.5$ [5].

2.2 A general sequential approach of training a classifier

To address the problem of having a limited number of observations available to train a classifier due to various factors such as availability and cost, Fu *et al.* [2] suggests a sequential training approach, derived using the Martingale Central Limit theorem, in which a stopping rule is applied once a randomly sampled observation has been classified after each sequential step.

This process trains a classifier that ensures with a predetermined $(1-\alpha)\%$ level of confidence that the probability of the classifier making an error when clustering new data is less than a chosen upper limit $\varepsilon > 0$.

In order to derive the stopping rule let $\Upsilon_i = \{Y_1, Y_2, ..., Y_i\}$, $i = 1, 2, ..$ be a set of independent uncorrelated observations, $Q_i = \begin{cases} 1 & \text{if } Y_i \text{ was misclassified} \\ 0 & \text{otherwise} \end{cases}$ and $\pi_i = P(Q_i = 1|\Upsilon_{i-1})$ be the conditional probability of Y_i being misclassified given that the previous $i-1$ results are known [2].

Therefore, the goal is to find the number of observations, N, needed in order to be able to say with a certain level of confidence that the next observation will have a misclassification probability of $\pi_N \leq \varepsilon$, $0 < \varepsilon < 1$.

Two assumptions are then made, namely:

1. π_n is weakly monotonically decreasing and
2. $\pi_\infty > 0$.

The implication of the first assumption is that the more observations that are used in the training of the classifier, the lower the conditional probability of Y_i being misclassified becomes, i.e π_n will converge weakly to $\pi_\infty \geq 0$.

The second assumption implies that there exists a positive probability of misclassification no matter which method of classification is used.

If there exists a non-zero Bayes error, π_{Bayes}, then $\pi_\infty \geq \pi_{Bayes} > 0$ since the Bayes error by definition is the minimum feasible error rate and no classifier should be able to obtain an error rate lower than this [5].

The stopping rule on which this sequential procedure is based follows from the above mentioned assumptions as well as from a theorem based on the Martingale Central Limit theorem.

As stated and proved in Fu *et al.* [2], Theorem 1 suggests that as N approaches infinity there will be a probability tending to $1-\alpha$ that the probability of the trained classifier incorrectly classifying the next randomly sampled observation is below a constant value ε.

Let $\hat{\kappa}_N = N^{-1}\sum_{i=1}^{N}\hat{\pi}_i(1-\hat{\pi_i})$ and $\hat{\pi}_i = \frac{1}{i}\sum_{j=1}^{i}Q_j$ [2]. The minimum sample size required for $P(\pi_N < \varepsilon) \geq 1-\alpha$ can then be determined as follows

Set

$$\varepsilon = \frac{1}{N}\sum_{i=1}^{N}Q_i + z_{1-\alpha}\hat{\kappa}_N/N^{1/2}$$

$$\Rightarrow \varepsilon - \frac{1}{N}\sum_{i=1}^{N}Q_i = z_{1-\alpha}\hat{\kappa}_N/N^{1/2}$$

$$\Rightarrow N^{1/2} = \frac{z_{1-\alpha}\hat{\kappa}_N}{\varepsilon - \frac{1}{N}\sum_{i=1}^{N}Q_i}$$

$$\Rightarrow N = \left(\frac{z_{1-\alpha}\hat{\kappa}_N}{\varepsilon - \frac{1}{N}\sum_{i=1}^{N}Q_i}\right)^2$$

where $\varepsilon > \frac{1}{N}\sum_{i=1}^{N}Q_i$, $\hat{\kappa}_N > 0$ and $z_{1-\alpha}$ denotes the $(1-\alpha)$ quantile of the standard normal distribution.

In order to save resources and calculation time, Fu *et al.* [2] suggests a second rule whereby the se-

Algorithm 1 General sequential procedure

1. Obtain an initial sample of size S_0 and set $N_0 = 0$.
2. At the i^{th} iteration, train the classifier using all observations obtained thus far.
3. Sample an additional random observation and classify it using the classifier trained in step 2.
4. Test whether or not the new observation has been correctly classified. If a correct classification is made set $Q_i = 0$ and $N_0 = N_0 + 1$. If the observation is misclassified set $Q_i = 1$ and $N_0 = 0$.
5. Using equation 5 evaluate the stopping rule. If either the stopping rule is met or $N_0 = M$ stop the sequential training procedure, otherwise repeat steps 2 to 5.

quential training procedure is stopped if and when a sufficiently large number, given by N_0, of consecutive correct classifications occurs where $N_0 = \frac{\log(\alpha)}{\log(1-\varepsilon)}$ [2].

Therefore the number of sequential steps necessary before the sequential procedure is stopped is given by

$$N \geq \min\left\{\left(\frac{z_{1-\alpha}\hat{\kappa}_N}{\varepsilon - \frac{1}{N}\sum_{i=1}^{N} Q_i}\right)^2, N_0\right\} \tag{5}$$

where $0 < \frac{1}{N}\sum_{i=1}^{N} Q_i < \varepsilon < 1$ [2].

Assume that the maximum sample size is M, the sequential training procedure suggested by Fu *et al.* [2] is summarised by algorithm 1.

In order to demonstrate the sequential training method a series of simulations is run in Fu *et al.* [2] and the observed misclassification rate is compared to that of the theoretical or Bayes error, the results of which can be found in tables 1 and 2 of Fu *et al.* [2].

Since the Bayes error is the minimum feasible error rate of any classifier, an optimal classifier should have an error rate which tends towards the Bayes error.

2.3 Bayes error

Consider a data set consisting of two random samples, one generated from a $N(\mu_1, \sigma_1^2)$ and one from a $N(\mu_2, \sigma_2^2)$ with probabilities p and $1 - p$ respectively. The Bayes error in this case is the probability of misclassifying an observation as being an element from a $N(\mu_1, \sigma_1^2)$ distribution when it is actually from a $N(\mu_2, \sigma_2^2)$ or vice versa.

If the classifier is denoted by λ, then

$$
\begin{aligned}
\text{Bayes Error} &= p\left\{P\left[X > \lambda | X \sim N(\mu_1, \sigma_1^2)\right]\right\} + (1-p)\left\{P\left[[X \leq \lambda | X \sim N(\mu_2, \sigma_2^2)]\right]\right\} \\
&= p\left\{P\left[Z > \frac{\lambda - \mu_1}{\sigma_1}\right]\right\} + (1-p)\left\{P\left[Z \leq \frac{\lambda - \mu_2}{\sigma_2}\right]\right\} \\
&= p\left\{1 - P\left[Z \leq \frac{\lambda - \mu_1}{\sigma_1}\right]\right\} + (1-p)\left\{P\left[Z \leq \frac{\lambda - \mu_2}{\sigma_2}\right]\right\} \\
&= p\left\{1 - \Phi\left(\frac{\lambda - \mu_1}{\sigma_1}\right)\right\} + (1-p)\Phi\left(\frac{\lambda - \mu_2}{\sigma_2}\right) \qquad (6)
\end{aligned}
$$

Consider the case where there is equal probability of the observation being generated from each of the two distributions, i.e. where $p = \frac{1}{2}$ and $\lambda = \frac{\triangle}{2}$ then

$$
\begin{aligned}
\text{Bayes Error} &= \frac{1}{2}\left\{1 - \Phi\left(\frac{\lambda - \mu_1}{\sigma_1}\right)\right\} + (1 - \frac{1}{2})\Phi\left(\frac{\lambda - \mu_2}{\sigma_2}\right) \\
&= \frac{1}{2} - \frac{1}{2}\Phi\left(\frac{\lambda - \mu_1}{\sigma_1}\right) + \frac{1}{2}\Phi\left(\frac{\lambda - \mu_2}{\sigma_2}\right) \\
&= \frac{1}{2}\left[1 - \Phi\left(\frac{\lambda - \mu_1}{\sigma_1}\right) + \Phi\left(\frac{\lambda - \mu_2}{\sigma_2}\right)\right] \qquad (7)
\end{aligned}
$$

In figure 1, the density of two normal distributions are plotted, one a N(0,1) distribution the other a N(2,1) distribution.

In this case, an observation from a N(2,1) will misclassified as being from a N(0,1) distribution if the observation is smaller than λ and an observation from a N(0,1) will be misclassified as being from a N(2,1) distribution if the observation is larger than λ. The Bayes error is represented by the yellow shaded area of figure 1.

2.4 Performance of suggested sequential approach

In an effort to test the sequential training method previously discussed Potgieter [5] imitated the simulation study conducted in Fu *et al.* [2] changing only the method with which the resulting classifier is tested.

Initially 5 observations are randomly sampled from a $N(0,1)$ distribution and 5 from a $N(\triangle,1)$ distribution with probabilities $\hat{p}_1 = \frac{n_1}{n_1+n_2} = \frac{5}{10} = 0.5$ and $\hat{p}_2 = 1 - \hat{p}_1 = 0.5$ respectively, where $\triangle \epsilon \{1, 1.3, 1.5, 2, 2.3, 2.5, 3.4\}$. Using these observations a classifier is trained using the method outlined in algorithm 1 with $M \epsilon \{40, 90\}$ and $\varepsilon \epsilon \{0.05, 0.1, 0.15, 0.2\}$.

Once either $N_0 = M$ or the stopping rule is met the sequential procedure is stopped and the classifier is then used to classify an additional 10000 observations, 5000 each from a $N(0,1)$ and $N(\triangle,1)$ distribution respectively.

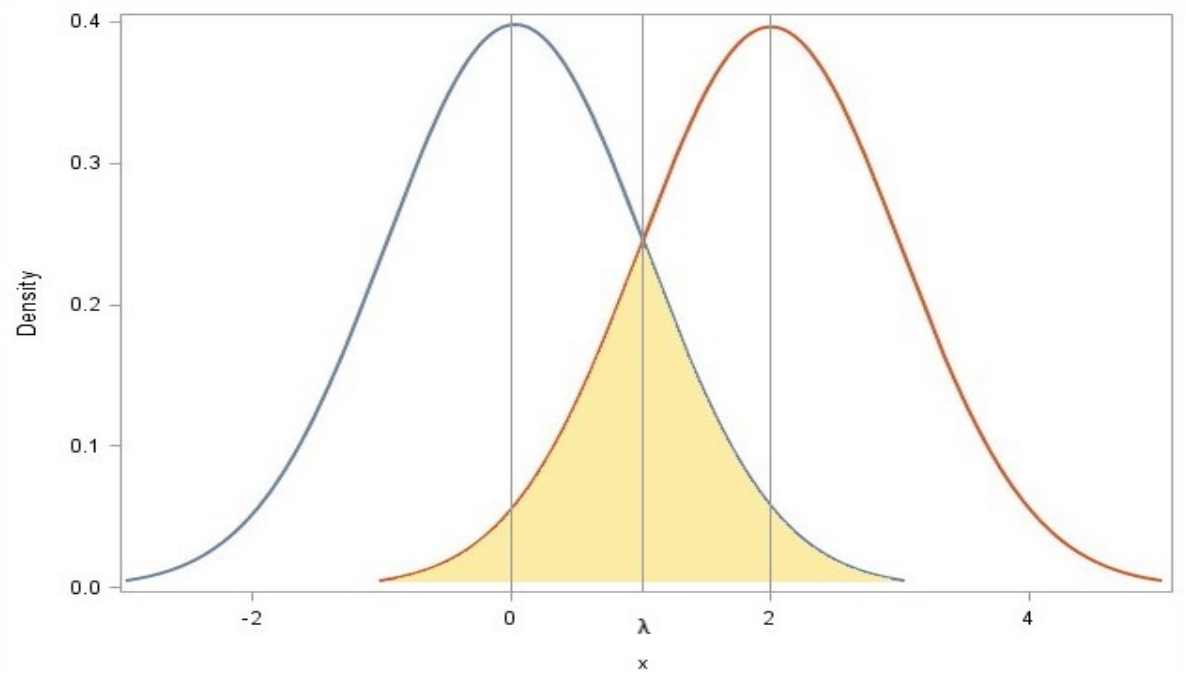


Figure 1: Graphical depiction of Bayes error

This process is repeated for 1000 iterations and the mean and standard deviation of the observed misclassification rate as well as the minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier are recorded.

In this case the Bayes error (following from equation 7) is calculated as

$$\begin{aligned}
\text{Bayes Error} &= \frac{1}{2}\left[1-\Phi\left(\frac{\Delta}{2}\right)+\Phi\left(\frac{\Delta}{2}-\triangle\right)\right] \\
&= \frac{1}{2}\left[1-\Phi\left(\frac{\Delta}{2}\right)+\Phi\left(-\frac{\Delta}{2}\right)\right] \\
&= \frac{1}{2}\left[1-\Phi\left(\frac{\Delta}{2}\right)+\left(1-\Phi\left(\frac{\Delta}{2}\right)\right)\right] \\
&= 1-\Phi\left(\frac{\Delta}{2}\right)
\end{aligned}$$

Various methods of classification are tested, the results of which can be found in section 2.3 of Potgieter [5]. From the results, it is evident that in many cases the maximum sample size M is reached at least once for the combinations of ε and $\triangle$ that are tested. It can also be seen that the observed misclassification rate for various combinations is lower than the Bayes error, this should not be the case as this violates the definition of Bayes error. In addition to this there are numerous occasions where the sequentially trained classifier yields a misclassification rate smaller than that of the pre-specified ε, indicating that the classifier is over trained. Instead of attempting to obtain a possible unrealistic specified error rate, the goal should be to pursue the minimum feasible rate of error. [5].

3 A sequential procedure for estimating the Bayes error

The approach suggested by Fu *et al.* [2] is unable to account for those situations where the pre-specified maximum misclassification rate ε is smaller than the Bayes error, causing the sequential procedure to continue indefinitely.

In order to address this problem Potgieter [5] suggests a sequential procedure which trains a classifier to estimate the Bayes error and ensures that the rate of misclassification of the classifier remains within a pre-specified level of this error.

In practice it is often the case that the Bayes error is unable to be calculated since the underlying distribution of the data is unknown, it is therefore desirable to train a classifier using a sequential method which stops only when the misclassification rate falls within a certain range of the Bayes error [5]. Using the notation previously defined in section 2, where $Q_i = 1$ if an observation is misclassified and $Q_i = 0$ otherwise, the ratio $\hat{p} = \frac{\sum Q_i}{n}$ is thus an estimate of the Bayes error. In order to train a classifier to estimate the proportion, p, of misclassified observations within a certain level of accuracy it is first necessary to determine a fixed-width confidence interval.

3.1 Obtaining a fixed-width confidence interval for a proportion

When conducting fixed sample size simulation studies or designed experiments it is often the desire of the researcher to be able to estimate a proportion p as accurately as possible, however depending on the number of successes observed during the experiment the confidence intervals for p will differ. It is also necessary to conduct an extremely large amount of trials in order to obtain accuracy and this can become computationally intensive and costly.

Frey [1] suggests four methods of determining a fixed-width sequential confidence interval for a proportion. These fixed-width confidence intervals allow the researcher to specify a target half-width for the interval so that the estimate for p lies within a certain pre-defined level of accuracy.

These four methods, although each making use of different stopping criteria, have a common algorithm based on a number of independent and identically distributed Bernoulli trials with probability of success p. The fixed-width confidence intervals for p have the general form

$$[\max(0, \hat{p} - h), \min(1, \hat{p} + h)]$$

where h is the user-defined half-width.

These confidence intervals ensure a coverage probability of $(1 - \alpha)$ or more while requiring the least number of observations. Through simulation study it is seen that the method based on an adapted version of the Wald confidence interval for a proportion performs the best [1]. An outline of this method will be

Algorithm 2 Sequential Method of Obtaining a fixed-width confidence interval for p

1. Select a desired half-width $h \epsilon \{0.1, 0.05, 0.01\}$ and the associated values of a and α.
2. Conduct a Bernoulli trial with probability of success p.
3. Compute the adapted Wald confidence interval $\hat{p} \pm Z_{\frac{\alpha}{2}} \sqrt{\frac{\widetilde{p}_a(1-\widetilde{p}_a)}{n}}$.
4. If this confidence interval falls entirely within the fixed-width interval $\hat{p} \pm h$ then the sequential process is stopped and the resulting confidence interval for p is given by

   $$[max(0, \hat{p} - h), min(1, \hat{p} + h)]$$

   otherwise return to step 2 and continue conducting these independent and identically distributed Bernoulli trials until the computed Wald confidence interval falls within $\hat{p} \pm h$ (this is equivalent to stopping the process when $\frac{\widetilde{p}_a(1-\widetilde{p}_a)}{n} \leq (\frac{h}{Z_{\frac{\alpha}{2}}})^2$).

discussed and a simple simulation study will be conducted in order to illustrate this method.

3.1.1 Sequential method

Let $X_1, X_2, ..., X_n$ be independent and identically distributed Bernoulli random variables with probability of success p, $X_i \sim BIN(1, p)$, and define $\hat{p} = \frac{\sum x_i}{n}$.

A $100(1-\alpha)\%$ $(0 < \alpha \leq 1)$ Wald confidence interval for a proportion is given as

$$\hat{p} \pm Z_{\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{8}$$

where $Z_{\frac{\alpha}{2}}$ is the upper $\frac{\alpha}{2}$ quantile of the standard normal distribution.

A complication arises however when $x = 0$ or when $x = n$ as this causes $\hat{p}(1-\hat{p}) = 0$ resulting in an interval length of zero. Since this will always be the case when $n = 1$ by replacing $\frac{\hat{p}(1-\hat{p})}{n}$ in equation 8 with a non-zero variance estimate by substituting $\hat{p}$ with $\widetilde{p}_a = \frac{\sum x_i + a}{n+2a}$, where a is a positive whole number, $\hat{p}$ will effectively be pulled towards $\frac{1}{2}$ and prevents us from ending up with an undesirable confidence interval width of zero. This results in the adapted Wald confidence interval

$$\hat{p} \pm Z_{\frac{\alpha}{2}} \sqrt{\frac{\widetilde{p}_a(1-\widetilde{p}_a)}{n}} \tag{9}$$

Frey used previously developed path counting ideas [6][3] to calculate critical values as well as suitable values of a in order to obtain the desired $100(1-\alpha)\%$ confidence intervals. These values can be found in Table 2 of Frey [1].

The sequential method of obtaining the fixed-width confidence intervals for the proportion p is summarized in algorithm 2.

a	α	Coverage probability	E(N)
2	0.0225	0.966	155.939
4	0.0356	0.972	99.368
6	0.0374	0.962	98.277

Table 1: Comparison of coverage probabilities, CP and expected trial numbers, E(N) over the three choices of a.

3.1.2 Simulation study

To illustrate the convergence of the fixed-width sequential confidence intervals found when applying the method outlined in Algorithm 2 a simple experiment is conducted.

Consider the event where a fair dice is rolled and where a success is counted if the resulting number is either a one or a two, giving $X_i \sim BIN(1, \frac{1}{3})$.

Using the various choices of critical value α and the corresponding value of a given in table 2 of Frey [1] 95% sequential confidence intervals for p of width $h = 0.1$ are obtained.

Figure 2 depicts the convergence of the adapted Wald confidence interval towards the fixed-width interval, displayed for two iterations of the sequential procedure for $a = \{2, 4, 6\}$ and $\alpha = \{0.0225, 0.0356, 0.0374\}$ respectively. The dark solid line represents the true value of p, the dotted lines represent the lower and upper limits of the adapted Wald confidence interval and the innermost solid lines represent the fixed-width lower and upper limits. As is evident in the graphs, the true value of p falls within the resulting confidence intervals in all 6 iterations. This is to be expected as the method used ensures a coverage probability of 95% or more.

The sequential process is stopped when both the upper and the lower limits of the Wald confidence interval fall within the fixed-width limits. This experiment is repeated for1000 iterations to verify that the coverage probability of the obtained confidence intervals is at least $(1 - \alpha)100\%$.

Table 1 compares the coverage probabilities as well as the expected number of trials required for each of the three choices of a. Both the table and the graphs suggest that the best choice for a is 6 as this leads to the smallest expected number of trials as well as the coverage probability closest to 95%.

3.2 Sequential wrapping procedure

Using the method proposed by Frey [1] a wrapping algorithm, summarised in algorithm 3 is used to sequentially train a classifier to estimate the Bayes error and ensure that the observed rate of misclassification remains within a pre-specified level of this error [5].

This method is beneficial primarily due to the fact that it allows the researcher to decide which classification method to use and when the sequential procedure should terminate. The process repeatedly samples, trains and classifies observations until the misclassification rate lies within h of the Bayes error with a certain probability, ensuring that the classifier is always at optimum performance. In addition to

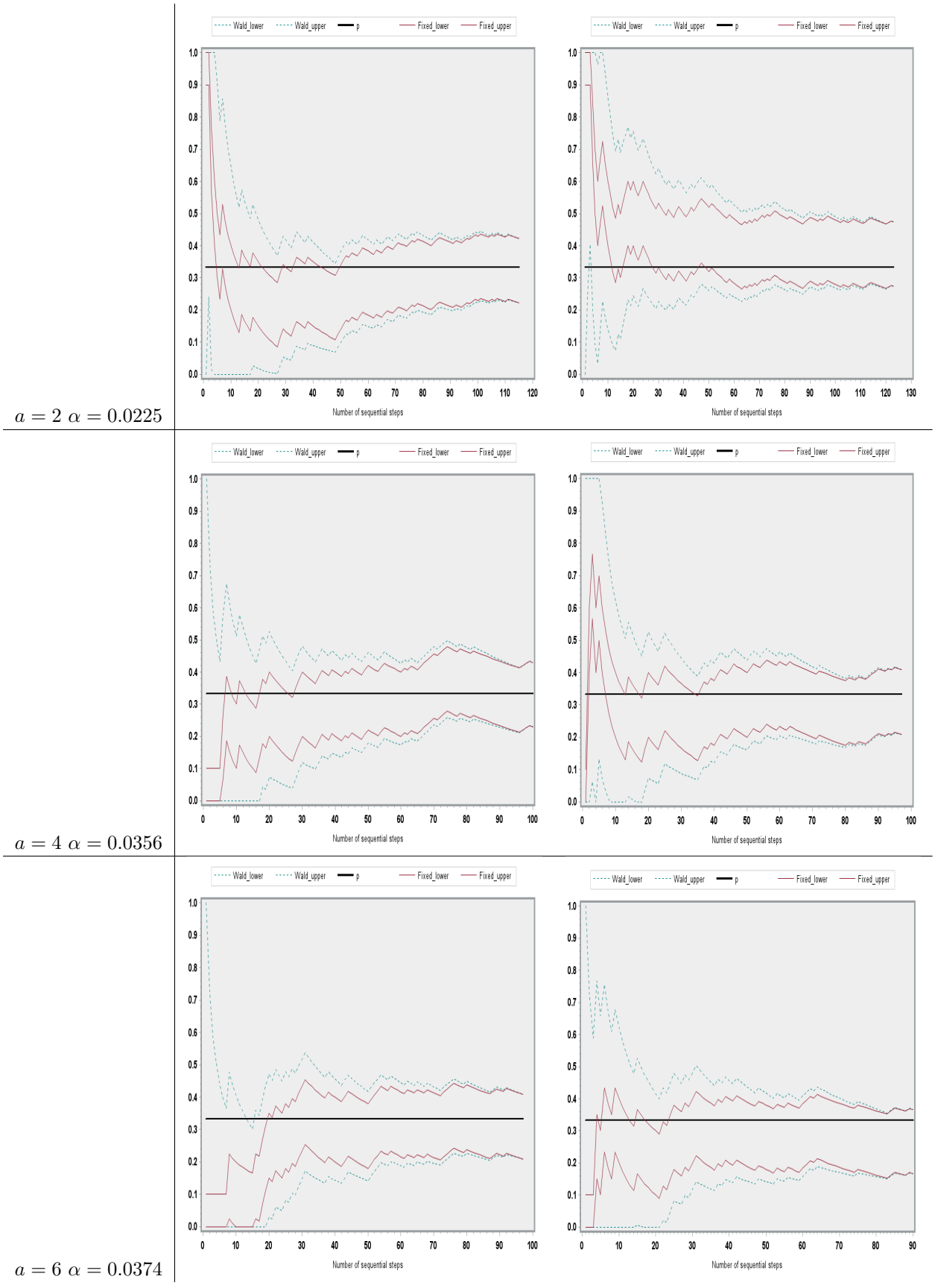


Figure 2: Convergence of the Wald confidence interval towards the fixed-width confidence intervals.

Algorithm 3

1. Obtain an initial sample of size S_0 and select a desired half-width $h \epsilon \{0.1, 0.05, 0.01\}$ and the associated values of a and α.
2. At the i^{th} iteration, train the classifier using all observations obtained thus far.
3. Sample an additional random observation and classify it using the classifier trained in step 2.
4. Test whether or not the new observation has been correctly classified. If a correct classification is made set $Q_i = 0$. If the observation is misclassified set $Q_i = 1$.
5. Calculate $\hat{p}$, the proportion of observations misclassified thus far, and as discussed in algorithm 2 evaluate the stopping rule: $\frac{\widetilde{p}_a(1-\widetilde{p}_a)}{n} \leq (\frac{h}{Z_{\frac{\alpha}{2}}})^2$. If the stopping rule is met, stop the sequential training procedure, if not repeat steps 2 to 5.

this the sequential procedure proposed by Potgieter [5] does not result in a classifier that is over trained and the sequential procedure never continues indefinitely.

3.3 Simulation study and analysis of results

To test the performance of the sequential method proposed by Potgieter [5] various simulations are run using the same simulation design as discussed in section 2.4, the only difference being the method used to train the classifier i.e algorithm 3. To imitate the results that are obtained in Potgieter [5], simulations are run using both LDA and KNN classification, the results of which can be found in tables 2, 3 and 4. In addition, further simulations are run using QDA classification (results given in table 5) and the coverage probability for all of the tested scenarios is calculated.

When $h = 0.1$ in table 2 the mean of the observed error rate is slightly higher than the Bayes error for all four choices of $\triangle$, never being lower than it. As $\triangle$ is increased from 1 to 4 the standard deviation of the observed error decreases from 0.0082022 when $\triangle = 1$ to 0.0024705 when $\triangle = 4$ implying that for greater values of $\triangle$ there is less variance in the observed error rate. As expected the average number of sequential steps needed to train the classifier decreases from 97.387 when $\triangle = 1$ to 40.946 when $\triangle = 4$ since the observations are easier to classify when $\triangle$ is larger.

As h is decreased from 0.1 to 0.05 it can be observed for all choices of $\triangle$ that while the mean of the observed error rate decreases slightly, the standard deviation of the observed error rate, average number of sequential steps required to train the classifier as well as the standard deviation in required step number all increase drastically. In all cases the coverage probability is higher than the desired $1 - \alpha = 0.95$ tending towards 1 for larger values of $\triangle$.

A maximum average step number of 97.387 when $h = 0.1$ or 354.9 when $h = 0.05$ is therefore required to ensure with a minimum probability of 0.95 that the trained classifier will have a misclassification error rate that is within h of the Bayes error.

Comparing the results given in table 3 to those in table 4 it can be seen that using $k = 7$ in the KNN

$\triangle$	**Bayes Error**	**Summary Statistics**	$h = 0.1$	$h = 0.05$
1	**0.3085**	Mean error	0.3127517	0.3098295
		Standard deviation error	0.0082022	0.0051270
		$\bar{n}$	97.387	354.9
		Standard deviation step number	6.7448294	14.3321810
		Coverage Probability	0.954	0.951
2	**0.1587**	Mean error	0.161760	0.1594909
		Standard deviation error	0.0055238	0.0037932
		$\bar{n}$	69.799	236.879
		Standard deviation step number	10.0067344	23.4117592
		Coverage Probability	0.964	0.954
3	**0.0668**	Mean error	0.0691845	0.067667
		Standard deviation error	0.0041446	0.0029772
		$\bar{n}$	49.787	148.049
		Standard deviation step number	7.6118047	19.7018975
		Coverage Probability	0.997	0.99
4	**0.0228**	Mean error	0.0240292	0.0232403
		Standard deviation error	0.0024705	0.001717
		$\bar{n}$	40.946	108.684
		Standard deviation step number	4.1291808	11.6744352
		Coverage Probability	1	1

Table 2: Summary Statistics of observed misclassification rate (Mean error and Standard deviation error) and required number of sequential steps ($\bar{n}$, standard deviation step number) as well as the calculated coverage probability obtained from the performance test of the sequentially trained LDA classifier

classification performs better than $k = 3$. When $h = 0.1$ in table 4 the mean of the observed error rate is higher than the Bayes error for all four choices of $\triangle$ (more so than is the case with the trained LDA classifier discussed above), never being lower than it. As $\triangle$ is increased from 1 to 4 the standard deviation of the observed error rate decreases from 0.0243099 when $\triangle = 1$ to 0.0047235 when $\triangle = 4$ implying that for greater values of $\triangle$ there is less variance in the observed error rate. The average number of sequential steps needed to train the classifier decreases from 100.809 when $\triangle = 1$ to 41.749 when $\triangle = 4$.

As h is decreased from 0.1 to 0.05 it can be observed for all choices of $\triangle$ that while the mean of the observed error rate decreases slightly, the standard deviation of the observed error rate, average number of sequential steps required to train the classifier as well as the standard deviation in required step number all increase drastically. Only in the case of $\triangle \geq 2$ for both $h = 0.1$ and $h = 0.05$ is the coverage probability higher than the desired $1 - \alpha = 0.95$, this differs from the trained LDA classifier possibly because of the difference in distance from the average observed error rate to the Bayes error between the two methods of classification and is to be expected as Linear Discriminant Analysis is a Bayes classifier and as such should be better able to estimate the Bayes error.

When $h = 0.1$ in table 5 the mean of the observed error rate is slightly higher than the Bayes error for all four choices of $\triangle$ (less so than is the case with the trained LDA classifier discussed earlier), never being lower than it. As $\triangle$ is increased from 1 to 4 the standard deviation of the observed error decreases from 0.0083795 when $\triangle = 1$ to 0.0046626 when $\triangle = 4$ implying that for greater values of $\triangle$ there is less

△	Bayes Error	Summary Statistics	$h = 0.1$	$h = 0.05$
1	**0.3085**	Mean error	0.3680616	0.3675528
		Standard deviation error	0.0222350	0.0124539
		$\bar{n}$	103.06	380.399
		Standard deviation step number	6.0093687	13.3504319
		Coverage Probability	0.777	0.371
2	**0.1587**	Mean error	0.1900856	0.1907666
		Standard deviation error	0.0242417	0.0121940
		$\bar{n}$	75.091	264.525
		Standard deviation step number	11.1248587	26.7912887
		Coverage Probability	0.913	0.726
3	**0.0668**	Mean error	0.0796523	0.0795092
		Standard deviation error	0.0139642	0.0097851
		$\bar{n}$	51.96	157.732
		Standard deviation step number	8.6015340	24.1168364
		Coverage Probability	0.98	0.932
4	**0.0228**	Mean error	0.0277258	0.0273362
		Standard deviation error	0.0079546	0.0060407
		$\bar{n}$	41.839	110.8
		Standard deviation step number	5.1054034	13.6112919
		Coverage Probability	0.999	0.994

Table 3: Summary Statistics of observed misclassification rate (Mean error and Standard deviation error) and required number of sequential steps ($\bar{n}$, standard deviation step number) as well as the calculated coverage probability obtained from the performance test of the sequentially trained 3-NN classifier

△	Bayes Error	Summary Statistics	$h = 0.1$	$h = 0.05$
1	**0.3085**	Mean error	0.3451829	0.3447459
		Standard deviation error	0.0243099	0.0128402
		$\bar{n}$	100.809	370.561
		Standard deviation step number	6.8859230	15.1838344
		Coverage Probability	0.855	0.659
2	**0.1587**	Mean error	0.1747711	0.1752994
		Standard deviation error	0.0164576	0.0169783
		$\bar{n}$	72.389	248.266
		Standard deviation step number	10.8214039	10.4079965
		Coverage Probability	0.951	0.96
3	**0.0668**	Mean error	0.0735859	0.078396
		Standard deviation error	0.0099096	0.0078396
		$\bar{n}$	50.505	152.65
		Standard deviation step number	8.0437975	22.0495279
		Coverage Probability	0.995	0.966
4	**0.0228**	Mean error	0.0257467	0.0251590
		Standard deviation error	0.0047235	0.0036835
		$\bar{n}$	41.749	110.365
		Standard deviation step number	4.5343753	13.1857362
		Coverage Probability	1	0.997

Table 4: Summary Statistics of observed misclassification rate (Mean error and Standard deviation error) and required number of sequential steps ($\bar{n}$, standard deviation step number) as well as the calculated coverage probability obtained from the performance test of the sequentially trained 7-NN classifier

$\triangle$	**Bayes Error**	**Summary Statistics**	$h = 0.1$	$h = 0.05$
1	**0.3085**	Mean error	0.3130873	0.3095449
		Standard deviation error	0.0083795	0.0048039
		$\bar{n}$	98.237	355.173
		Standard deviation step number	6.6920561	14.3751771
		Coverage Probability	0.952	0.95
2	**0.1587**	Mean error	0.1614599	0.1596404
		Standard deviation error	0.0057934	0.0038332
		$\bar{n}$	70.688	0.1596404
		Standard deviation step number	9.6130675	22.3887888
		Coverage Probability	0.971	0.968
3	**0.0668**	Mean error	0.0699172	0.0680353
		Standard deviation error	0.0054637	0.003007
		$\bar{n}$	50.933	148.732
		Standard deviation step number	7.2227889	19.8116804
		Coverage Probability	0.997	0.992
4	**0.0228**	Mean error	0.0255584	0.023989
		Standard deviation error	0.0046626	0.002291
		$\bar{n}$	41.445	109.89
		Standard deviation step number	4.3870515	12.0526830
		Coverage Probability	1	0.994

Table 5: Summary Statistics of observed misclassification rate (Mean error and Standard deviation error) and required number of sequential steps ($\bar{n}$, standard deviation step number) as well as the calculated coverage probability obtained from the performance test of the sequentially trained QDA classifier

variance in the observed error rate. The average number of sequential steps needed to train the classifier decreases from 98.237 when $\triangle = 1$ to 41.445 when $\triangle = 4$.

As h is decreased from 0.1 to 0.05 it can be observed for all choices of $\triangle$ that while the mean of the observed error rate decreases slightly, the standard deviation of the observed error, average number of sequential steps required to train the classifier as well as the standard deviation in required step number all increase drastically. In all cases it can be noted that the coverage probability is at least the desired $1 - \alpha = 0.95$.

A maximum average of 98.237 when $h = 0.1$ or 355.173 when $h = 0.05$ is therefore required to ensure with a minimum probability of 0.95 that the trained classifier will have a misclassification error rate that is within h of the Bayes error.

Overall, the results in table 2, 3, 4 and 5 suggest that using LDA classification yields the best results and although using $h = 0.05$ yields lower mean error rates, due to the extreme increase in standard deviation and required step number for sequential training it is better to use $h = 0.1$ since the difference in mean error rates is only slight.

To test the sensitivity of the suggested procedure to differing sampling probabilities another simulation study is conducted using LDA classification for $h = 0.1$ this time with $\hat{p}_1 = 0.8$ and $\hat{p}_2 = 0.2$, the results of which are given in table 6. For smaller values of $\triangle$ the observed average number of sequential steps necessary to train the classifier is much smaller than in the case of equal sampling probabilities. When

$\triangle$	**Bayes Error**	**Summary Statistics**	$h = 0.1$
1	**0.18616**	Mean error	0.4054810
		Standard deviation error	0.0403615
		$\bar{n}$	79.094
		Standard deviation step number	9.0153447
		Coverage Probability	0.955
2	**0.11207**	Mean error	0.2027102
		Standard deviation error	0.0261420
		$\bar{n}$	61.48
		Standard deviation step number	8.9984761
		Coverage Probability	0.985
3	**0.04983**	Mean error	0.0829697
		Standard deviation error	0.0134277
		$\bar{n}$	47.104
		Standard deviation step number	6.5644708
		Coverage Probability	1
4	**0.0174**	Mean error	0.0281888
		Standard deviation error	0.0059504
		$\bar{n}$	40.461
		Standard deviation step number	3.9050901
		Coverage Probability	1

Table 6: Summary Statistics of observed misclassification rate (Mean error and Standard deviation error) and required number of sequential steps ($\bar{n}$, standard deviation step number) as well as the calculated coverage probability obtained from the performance test of the sequentially trained LDA classifier with sampling probabilities $\hat{p}_1 = 0.8$ and $\hat{p}_2 = 0.2$

80% of the data is generated from a $N(0, 1)$ distribution a maximum average of 79.094 is required to ensure with a minimum probability of 0.95 that the trained classifier will have a misclassification error rate that is within h of the Bayes error, this is 18.293 less than when there are equal sampling probabilities. For all four choices of $\triangle$ the standard deviation of the error rate is larger than in the case of equal sampling probabilities indicating that there is a larger spread in the observed error rates for unequal sampling probabilities. For all choices of $\triangle$ the coverage probability is higher than the desired $1-\alpha = 0.95$ tending towards 1 for larger values of $\triangle$ as was the case when there were equal sampling probabilities.

4 Handwritten digit recognition application

The sequential wrapping method of training a classifier discussed in this paper can be applied to many real-world data sets, as an example the procedure is applied to a set of handwritten digits from the ZIP codes on envelopes from the U.S. postal mail [4]. As discussed in Hastie *et al.*[4] based on the pixel intensity of 16x16 eight bit greyscale images of single digits the goal is to train a classifier using the least amount of observations to classify an image of a handwritten digit into one of the groups $0, 1, 2, 3, 4, 5, 6, 7, 8$ or 9 as accurately as possible. For the purpose of this study two data sets are used, a training data set consisting of a spread of 2000 handwritten digits ranging from 0 to 9 and a testing data set consisting of 5196 digits. For comparison purposes various classifiers are trained using the full

Classification Method Used	Average misclassification rate
LDA	0.1383757
5-NN	0.1853349
7-NN	0.202271

Table 7: Comparison of average misclassification rates obtained using LDA and KNN classification methods

Initial Sample Size	Summary Statistics	h=0.1	h=0.05
150	Mean error	0.2308494	0.1869669
	Standard deviation error	0.0139988	0.0097419
	Min step number; Max step number	49; 103	233; 334
	$\bar{n}$	87.76	286.76
	Standard deviation step number	8.1138543	17.6979429
200	Mean error	0.2122646	0.1811650
	Standard deviation error	0.0111866	0.0091327
	Min step number; Max step number	55; 101	223; 319
	$\bar{n}$	81.4066667	273.18
	Standard deviation step number	8.6599415	20.0043854

Table 8: Summary Statistics of observed misclassification rate (Mean error and Standard deviation error) and required number of sequential steps (Min step number; Max step number, $\bar{n}$, standard deviation step number) obtained from the simulation study of the sequentially trained LDA classifier on the handwritten digit data set

2000 observations of the training data set and are then used to classify the remaining 5196 observations. The average misclassification rate is observed and the results given in table 7. Using algorithm 3 the procedure is applied to the training data set using both LDA and KNN classification methods, once the stopping rule is met the sequential procedure is stopped and the resulting classifier used to classify the remaining observations in the testing data set. The simulations are run on an initial training sample of 150 hand written digits (15 of each digit) as well as on an initial sample of 200 hand written digits (20 of each digit) the results can be seen in table 8, 9 and 10.

Various values of k were used for the sequentially trained KNN classifiers and as can be seen when comparing the results given in table 9 to that of those in table 10, smaller values of k yield better results.

Initial Sample Size	Summary Statistics	h=0.1	h=0.05
150	Mean error	0.4036977	0.3286855
	Standard deviation error	0.0189215	0.0135625
	Min step number; Max step number	97; 111	351; 408
	$\bar{n}$	108.25333	384
	Standard deviation step number	3.0809801	10.7853249
200	Mean error	0.3862933	0.3535681
	Standard deviation error	0.0159321	0.0128691
	Min step number; Max step number	94; 111	369; 409
	$\bar{n}$	107.1866667	397.32
	Standard deviation step number	3.554724	7.4493305

Table 9: Summary Statistics of observed misclassification rate (Mean error and Standard deviation error) and required number of sequential steps (Min step number; Max step number, $\bar{n}$, standard deviation step number) obtained from the simulation study of the sequentially trained 5-NN classifier on the handwritten digit data set

Initial Sample Size	Summary Statistics	h=0.1	h=0.05
150	Mean error	0.4352874	0.3535681
	Standard deviation error	0.0220439	0.0128691
	Min step number; Max step number	101; 111	369; 409
	$\bar{n}$	109.4466667	397.32
	Standard deviation step number	2.0774813	7.4493305
200	Mean error	0.4221722	0.3488376
	Standard deviation error	0.0189825	0.0149421
	Min step number; Max step number	96; 111	363; 407
	$\bar{n}$	109.0333333	395.43
	Standard deviation step number	2.7181139	8.1491835

Table 10: Summary Statistics of observed misclassification rate (Mean error and Standard deviation error) and required number of sequential steps (Min step number; Max step number, $\bar{n}$, standard deviation step number) obtained from the simulation study of the sequentially trained 7-NN classifier on the handwritten digit data set

In both the LDA and KNN cases and for both $h = 0.1$ and $h = 0.05$ the mean of the observed error rate and the standard deviation of the error rate are similar for either choice of initial sampling size but the number of sequential steps needed to train the classifier decreases as the initial sampling size of each digit is increased from 15 to 20.

When comparing the minimum error observed during these simulations to that obtained when the full 2000 observations of the training data set were used (table 7) it can be seen that the sequentially trained LDA classifier performs the best. In this case it was possible to obtain, with only a maximum of 301 training observations when $h = 0.1$ or 519 training observations when $h = 0.05$, an observed misclassification rate that is within h of that given in table 7. This is quite a substantial difference to the 2000 training observations needed to obtain the result in table 7. For all four combinations of factors (classification method, value of h, and initial sample size) the classifiers are fully trained, never reaching the maximum amount of observations available with which to train the classifier and never resulting in the sequential procedure continuing indefinitely. The sequential method allowed the researcher to train the classifier using either classification method and to decide on a desired level of accuracy for the classifier to estimate the minimum feasible error rate.

In addition, using the LDA classification method, heat maps of each digit were generated using SAS, these can be seen in figure 3 where the first row of images depict the heat maps of the original observed digits from the testing data set and the second row depicts the heat maps of the predicted/classified digits for a simulation case of the procedure if $h = 0.1$ and an initial sample size of 200. As can be seen by only using the minimum number of observations (roughly 281) the classifier was able to predict the digits relatively accurately.

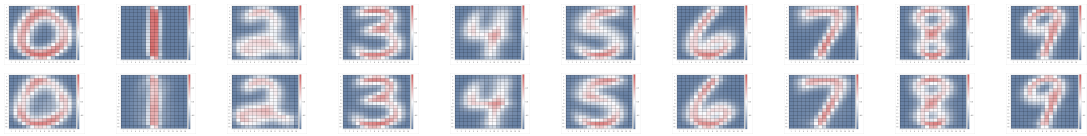


Figure 3: Heat maps of original observed digits from testing data set (top row) and predicted digits for one simulation of the procedure for $h = 0.1$ and initial sample size of 200 (bottom row)

5 Conclusion

In experimental cases it is often not possible to obtain large amounts of data with which to accurately train a classifier. A sequential method of training a classifier to estimate the Bayes error, as proposed in Potgieter [5] is discussed. This sequential training method is able to ensure with a certain level of confidence that the probability of the classifier making an error is within a pre-specified level of the absolute minimum feasible error, the Bayes error, whilst only requiring the smallest possible number of observations. The method is ideal as it gives the researcher more control over the process by specifying when the sequential procedure should be stopped, it is also not restricted to any single method of classification due to the constant updating of classification rules at each step. This classifier training method can prove useful in many real-world situations, saving on required observations and computational time. Further research may be conducted to investigate why, when using KNN classification, the results are not as expected.

References

[1] J. Frey. Fixed-width sequential confidence intervals for a proportion. *The American Statistician*, 64(3):242–249, 08 2010.

[2] W.J. Fu, E.R. Dougherty, B. Mallick, and R. J. Carroll. How many samples are needed to build a classifier: a general sequential approach. *Bioinformatics*, 21(12005):63–70, 2005.

[3] M. A. Girshick, F. Mosteller, and L. J. Savage. Unbiased estimates for certain binomial sampling problems with applications. *Annals of Mathematical Statistics*, 17:13–23, 1946.

[4] T Hastie, R Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics, 2009.

[5] R. Potgieter. Minimum sample size for estimating the bayes error at a predetermined level. Master's thesis, University of Pretoria, 2013.

[6] J.R. Schultz, F.R. Nichol, G.L. Elfring, and S.D. Weed. Multiple-stage procedures for drug screening. *Biometrics*, 29:293–300, 1973.

Appendix

SAS/IML code used in the simulation study of subsubsection 3.1.2

```
proc iml;
    p = 2/6;
    do experiment=1 to 1000;
    flag = 0;
    x = {};
    CI = {};
    do until(flag);
    x1 = int(6*ranuni(0)) + 1; *Outcome of dice roll;
    if x1<=2 then x2=1; *It is counted as a success if we obtain a number of 2 or less;
    else x2=0; x = x // (x1||x2);
    ** CHOOSE VALUE FOR a (2, 4 or 6) **;
    a = 4;
    h = 0.1; *Target half-width;
    if a = 2 then gamma = 0.0225; if a = 4 then gamma = 0.0356;
    if a = 6 then gamma = 0.0374;
    Z = probit(1-(gamma/2));
    n = nrow(x);
    phat = sum(x[,2])/n; *Number of successes divided by number of observations;
    p_a = (sum(x[,2])+a) / (n+(2*a)); *Adapted Wald confidence interval;
    *Upper and lower limits for the adapted Wald confidence interval;
    Wald_lower = max(0,phat - Z*sqrt(p_a*(1-p_a)*(1/n)));
    Wald_upper = min(1,phat + Z*sqrt(p_a*(1-p_a)*(1/n)));
    *Upper and lower limits for the fixed-width confidence interval;
    Fixed_lower = max(0, phat - h);
    Fixed_upper = min(1, phat + h);
    CI = CI // (n||Wald_lower||Wald_upper||Fixed_lower||Fixed_upper||p);
    w1 = p_a*(1-p_a)*(1/n);
    w2 = (h/Z)**2;
    *Stopping rule;
    if w1<=w2 then flag=1;
    end;
```

```
*Testing if the CI contains the true value of p;
p=2/6;
ind = Fixed_lower <= p & p <= Fixed_upper;
limits = limits//(Fixed_lower || Fixed_upper || phat || n || ind);
end;
nl = nrow(limits);
Coverage_probability = sum(limits[,5])/nl;
Expected_trial_no = sum(limits[,4])/nl;
print nl Coverage_probability Expected_trial_no;
cn = {'n' 'Wald_lower' 'Wald_upper' 'Fixed_lower' 'Fixed_upper' 'p'};
create convergence from CI[colname=cn];
append from CI;
run;
goptions reset=all cback=white border htext=10pt htitle=12pt;
title1 "Convergence process of the Wald interval towards the fixed width interval";
symbol1 interpol=join color=vibg line=2;
symbol2 interpol=join color=vibg line=2;
symbol3 interpol=join width=2 color=black;
symbol4 interpol=join color=depk;
symbol5 interpol=join color=depk;
axis1 value=(font='Arial/bold' height=11pt) width=2 label=('Number of sequential steps');
axis2 value=(font='Arial/bold' height=11pt) width=2 label=none;
legend1 label=none position=top frame;
proc gplot data=convergence;
plot Wald_lower*n Wald_upper*n p*n Fixed_lower*n Fixed_upper*n
/ overlay legend=legend1 haxis=axis1 vaxis=axis2 cframe=greyee;
run;
```

SAS/IML Studio code used in the simulation study of subsection 3.3

For LDA and QDA Classification:

```
delta = 1;
BE = 1 - probnorm(delta/2);
h = 0.05; *Chosen target half_width h={0.1, 0.05, 0.01};
if h = 0.1 then gamma = 0.0356; if h = 0.05 then gamma = 0.0433; if h = 0.01 then gamma = 0.0487;
```

```
if h = 0.1 then a=4; if h = 0.05 then a=6; if h= 0.01 then a=10;
Z = probit(1 - (gamma/2));
do sim=1 to 1000;
S_0 = 5;
x_0 = j(S_0, 2, 0);
x_delta = j(S_0, 2, delta);
do i=1 to S_0;
x_0[i,2] = rannor(0);
x_delta[i,2] = delta + rannor(0);
end;
sample = x_0//x_delta;
flag = 0;
free misclassification;
free AM;
free x_new;
do until(flag);
/* TRAINING DATA SET */
sample = sample//x_new;
cn1 = {'Group' 'Value'};
create sampledata from sample[colname=cn1];
append from sample;
close sampledata;
/* NEW OBSERVATION TO BE CLASSIFIED */
phat1 = 0.5; *Probability of new observation being from a N(0,1)distribution;
phat2 = 1-phat1; *Probability of new observation being from a N(delta,1) distribution;
x = rand('Bernoulli', phat1);
if x = 1 then value = rannor(0); else value = delta + rannor(0);
if x = 1 then truegroup = 0; else truegroup = delta;
totest = truegroup||value;
create newobs from totest[colname=cn1];
append from totest;
close newobs;
/* CLASSIFYING THE NEW OBSERVATION */
submit;
```

```
proc discrim data=sampledata outstat=calib method=normal pool=yes noprint;
class group; *IF QDA THEN POOL=NO;
priors prop;
run;
proc discrim data=calib testdata=newobs testout=grouping noprint;
class group;
run;
endsubmit;
/* VERIFYING WHETHER OR NOT A CORRECT CLASSIFICATION WAS MADE */
use grouping;
read all into test;
close grouping;
if test[1,5] = truegroup then Qi = 0; else Qi = 1;
misclassification = misclassification // Qi;
/* EVALUATING THE STOPPING CRITERIA */
phat = misclassification[:]; n = nrow(misclassification);
p_a = (misclassification[+]+a)/(n+2*a);
Fixed_lower = max(0, phat - h); Fixed_upper = min(1, phat + h);
Wald_lower = phat - Z*sqrt(p_a*(1-p_a)/n);
Wald_upper = phat + Z*sqrt(p_a*(1-p_a)/n);
w1 = p_a*(1-p_a)/n; w2 = (h/Z)**2;
if w1<=w2 then flag=1; else x_new = truegroup||test[1,2];
end;
ind = Fixed_lower <= BE & BE <= Fixed_upper;
limits = limits//(Fixed_lower||Fixed_upper||phat||n||ind);
call randseed(0);
xh0=j(5000,1); xhdelta=j(5000,1);
call randgen(xh0, "Normal", 0, 1); call randgen(xhdelta, "Normal", delta,1);
yh0 = j(5000,1,0); yhdelta = j(5000,1,delta);
finaltest = (yh0||xh0)//(yhdelta||xhdelta);
create FT from finaltest[colname=cn1];
append from finaltest;
close FT;
submit;
```

```
proc discrim data=calib testdata=FT testout=grouping2 noprint;
class group;
run;
endsubmit;
use grouping2;
read all into Ftest;
close grouping2;
nt=nrow(FTEST);
do i=1 to nt;
if Ftest[i,1] = Ftest[i,5] then Fi = 0; else Fi = 1; AM = AM // Fi;
end;
ACTUAL_ERROR = AM[:];
errortest = errortest//actual_error;
end;
nl = nrow(limits);
CP = sum(limits[,5])/nl;
Expected_trial_number = sum(limits[,4])/nl;
print CP;
resSteps = limits[,4];
create steps from resSteps[colname='stepno'];
append from resSteps;
close steps;
submit;
proc means data=steps min max mean std;
var stepno;
title 'steps';
run;
endsubmit;
create results from errortest[colname='error'];
append from errortest;
close results;
submit;
proc means data=results mean std;
var error;
```

```
title 'error';
run;
endsubmit;
```

For KNN Classification:

```
delta = 1;
BE = 1 - probnorm(delta/2);
h = 0.05; *Chosen target half_width h={0.1, 0.05, 0.01};
if h = 0.1 then gamma = 0.0356; if h = 0.05 then gamma = 0.0433; if h = 0.01 then gamma = 0.0487;
if h = 0.1 then a=4; if h = 0.05 then a=6; if h= 0.01 then a=10;
Z = probit(1 - (gamma/2));
do sim=1 to 1000;
S_0 = 5;
x_0 = j(S_0, 2, 0);
x_delta = j(S_0, 2, delta);
do i=1 to S_0;
x_0[i,2] = rannor(0);
x_delta[i,2] = delta + rannor(0);
end;
sample = x_0//x_delta;
flag = 0;
free misclassification;
free AM;
free x_new;
do until(flag);
/* TRAINING DATA SET */
sample = sample//x_new;
cn1 = {'Group' 'Value'};
create sampledata from sample[colname=cn1];
append from sample;
close sampledata;
/* NEW OBSERVATION TO BE CLASSIFIED */
phat1 = 0.5; *Probability of new observation being from a N(0,1)distribution;
phat2 = 1-phat1; *Probability of new observation being from a N(delta,1) distribution;
```

```
x = rand('Bernoulli', phat1);
if x = 1 then value = rannor(0); else value = delta + rannor(0);
if x = 1 then truegroup = 0; else truegroup = delta;
totest = truegroup||value;
create newobs from totest[colname=cn1];
append from totest;
close newobs;
/* CLASSIFYING THE NEW OBSERVATION */
submit;
proc discrim data=sampledata method=npar k=3 testdata=newobs testout=grouping noprint;
class group;
run;
endsubmit;
/* VERIFYING WHETHER OR NOT A CORRECT CLASSIFICATION WAS MADE */
use grouping;
read all into test;
close grouping;
if test[1,5] = truegroup then Qi = 0; else Qi = 1;
misclassification = misclassification // Qi;
/* EVALUATING THE STOPPING CRITERIA */
phat = misclassification[:]; n = nrow(misclassification);
p_a = (misclassification[+]+a)/(n+2*a);
Fixed_lower = max(0, phat - h); Fixed_upper = min(1, phat + h);
Wald_lower = phat - Z*sqrt(p_a*(1-p_a)/n);
Wald_upper = phat + Z*sqrt(p_a*(1-p_a)/n);
w1 = p_a*(1-p_a)/n; w2 = (h/Z)**2;
if w1<=w2 then flag=1; else x_new = truegroup||test[1,2];
end;
ind = Fixed_lower <= BE & BE <= Fixed_upper;
limits = limits//(Fixed_lower||Fixed_upper||phat||n||ind);
call randseed(0);
xh0=j(5000,1); xhdelta=j(5000,1);
call randgen(xh0, "Normal", 0, 1); call randgen(xhdelta, "Normal", delta,1);
yh0 = j(5000,1,0); yhdelta = j(5000,1,delta);
```

```
finaltest = (yh0||xh0)//(yhdelta||xhdelta);
create FT from finaltest[colname=cn1];
append from finaltest;
close FT;
submit;
proc discrim data=sampledata method=npar k=3 testdata=FT testout=grouping2 noprint;
class group;
run;
endsubmit;
use grouping2;
read all into Ftest;
close grouping2;
nt=nrow(FTEST);
do i=1 to nt;
if Ftest[i,1] = Ftest[i,5] then Fi = 0; else Fi = 1; AM = AM // Fi;
end;
ACTUAL_ERROR = AM[:];
errortest = errortest//actual_error;
end;
nl = nrow(limits);
CP = sum(limits[,5])/nl;
Expected_trial_number = sum(limits[,4])/nl;
print CP;
resSteps = limits[,4];
create steps from resSteps[colname='stepno'];
append from resSteps;
close steps;
submit;
proc means data=steps min max mean std;
var stepno;
title 'steps';
run;
endsubmit;
create results from errortest[colname='error'];
```

```
append from errortest;
close results;
submit;
proc means data=results mean std;
var error;
title 'error';
run;
endsubmit;
```

SAS/IML code used in section 4

```
* VALUES FOR STOPPING CRITERIA ;
h = 0.05; *Chosen target half_width h={0.1, 0.05, 0.01};
if h = 0.1 then gamma = 0.0356;
if h = 0.05 then gamma = 0.0433;
if h = 0.01 then gamma = 0.0487;
if h = 0.1 then a=4;
if h = 0.05 then a=6;
if h= 0.01 then a=10;
Z = probit(1 - (gamma/2));
*IMPORTING DATA SETs ;
submit;
proc import OUT= WORK.Digit DATAFILE= "C:\Users\Catherine\Desktop\Training.csv" DBMS=CSV
REPLACE; GETNAMES=YES; DATAROW=2;
run;
proc import OUT= WORK.EvTestSample DATAFILE= "C:\Users\Catherine\Desktop\Test.csv"
DBMS=CSV REPLACE; GETNAMES=YES; DATAROW=2;
run;
endsubmit;
* STRAIGHT CLASSIFICATION ;
submit;
proc discrim data=digit outstat=calib method=normal pool=yes noprint;
class y;
priors prop;
run;
```

```
proc discrim data=calib testdata=EvTestSample testout=grouping0 noprint;
class y;
run;
endsubmit;
use grouping0;
read all into tester;
close grouping0;
n=nrow(tester);
nc=ncol(tester);
free T; do i=1 to n;
if tester[i,1] = tester[i,nc] then Ti = 0;
else Ti = 1; T = T // Ti;
end;
TEST_ERROR = T[:];
print TEST_ERROR;
free step_no; free errortest;
* OBTAINING INITIAL SAMPLE DATA ;
submit;
proc sort data=digit;
by y;
run;
proc surveyselect data=digit method=srs n=15 seed=0 out=InitialSample noprint;
strata y;
run;
/*title 'Initial Sample Data';
proc print data=InitialSample;
run;*/ endsubmit;
use InitialSample;
read all into sample_a;
close InitialSample;
del = ncol(sample_a)-2;
sample = sample_a[,1:del];
flag = 0; free misclassification; free EV; free x_new; do i=1 to 1;
do until(flag);
```

```
* TRAINING DATA SET ;
sample = sample//x_new;
cn1 = {'y'};
create sampledata from sample[colname=cn1];
append from sample;
close sampledata;
/*submit; title 'Training data set';
proc print data=sampledata;
run;
endsubmit;
* NEW OBSERVATION TO BE CLASSIFIED ;
submit;
proc surveyselect data=digit method=srs seed=0 n=1 out=totest noprint;
run;
endsubmit;
use totest;
read all into x;
close totest;
create totest_ from x[colname=cn1];
append from x;
close totest_;
/*submit; title 'Observation to be classified';
proc print data=totest_;
run;
endsubmit;
* CLASSIFYING THE NEW OBSERVATION ;
submit;
proc discrim data=sampledata outstat=calib method=normal pool=yes noprint;
class y;
priors prop;
run;
proc discrim data=calib testdata=totest_ testout=grouping noprint;
class y;
run;
```

```
/*title 'Classification results';
proc print data=grouping;
run;*/
endsubmit;
*VERIFYING WHETHER OR NOT A CORRECT CLASSIFICATION WAS MADE;
use grouping;
read all into test;
close grouping;
nc2=ncol(test);
if test[1,1] = test[1,nc2] then Qi = 0;
else Qi = 1;
misclassification = misclassification // Qi;
* EVALUATING THE STOPPING CRITERIA ;
phat = misclassification[:];
nm = nrow(misclassification);
p_a = (misclassification[+]+a)/(nm+2*a);
Fixed_lower = max(0, phat - h);
Fixed_upper = min(1, phat + h);
Wald_lower = phat - Z*sqrt(p_a*(1-p_a)/nm);
Wald_upper = phat + Z*sqrt(p_a*(1-p_a)/nm);
w1 = p_a*(1-p_a)/nm;
w2 = (h/Z)**2;
en = nc2-11;
if w1<=w2 then flag=1; else x_new = test[1,1:en];
/*title 'Stopping Criteria';
print w1 w2;*
*print 'end of loop';
end;
/*title 'Misclassification rate';
print misclassification;*/
Step_no = Step_no // nm;
* Testing accuracy of trained classifier ;
use EvTestSample;
read all into x0;
```

```
close EvTestSample;
create EvTestSample_ from x0[colname=cn1];
append from x0;
close EvTestSample_;
submit;
proc discrim data=calib testdata=EvTestSample_ testout=grouping2 noprint;
class y;
run;
/*title 'Classification results';
proc print data=grouping2;
run;*/
endsubmit;
use grouping2;
read all into Evtest;
close grouping2;
*print Evtest;
nc3=ncol(Evtest);
graph = Evtest[,nc3];
print graph; nt=nrow(Evtest);
free EV;
do j=1 to nt;
if Evtest[j,1] = Evtest[j,nc3] then Fi = 0; else Fi = 1;
EV = EV // Fi;
end;
ACTUAL_ERROR = EV[:]; errortest = errortest//ACTUAL_ERROR;
res = step_no||errortest;
cn2 = {'Steps' 'Error'};
create results from res[colname=cn2];
append from res;
close results;
submit;
proc means data=results min max mean std;
var error;
title 'error';
```

```
run;
proc means data=results min max mean std;
var steps;
title 'error';
run;
endsubmit;
```

Spatial density estimation

Moise Kabwe wa Kabwe 13078179

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisors: Dr I Fabris-Rotelli, Ms C Kraamwinkel

Department of Statistics, University of Pretoria

30 October 2017

Abstract

The `density.ppp` function in the `spatstat` package in `R` uses kernel density estimation to create a continuous intensity function for a spatial point pattern data set. This report examines how the function uses kernel density estimation to create these intensity functions, focusing on the main mathematical theory of kernel density estimation in a spatial context. The function will then be tested under various conditions and defaults.

Declaration

I, Moise Kabwe wa Kabwe, declare that this essay, submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics, at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____

Moise Kabwe wa Kabwe

_____

Inger Fabris-Rotelli, Christine Kraamwinkel

_____

Date

Acknowledgements

I thank my supervisors Dr Inger Fabris-Rotelli and Miss Christine Kraamwinkel for their advice, patience and encouragement while helping me on this report. I have learned so much under their guidance and I couldn't have ever asked for better supervisors.

I would like to acknowledge the financial support from the Center for AI Research, Meraka Institute, CSIR.

Contents

List of Figures

List of Tables

1 Introduction

Spatial statistics is the study of phenomena whose spatial location is either of interest or contributes to a stochastic model for that particular phenomenon [11]. The first person to study the implications of spatial dependence was R. A. Fisher [10] in his work on developing inferences for agricultural field experiments at the Rothamsted Experimental station in Hertforshire, England, where he developed methodologies for the analysis of data from these experiments [11].

To this day, there have been extensive contributions in the field of spatial statistics and the standard references include Diggle [9], Ripley [16, 17] and Cressie [6].

Spatial data can be explained as observations from a stochastic process $\{Z(s) : s \in D\}$, where D is a random set in $\mathbb{R}^d$ [6]. Spatial areas can be classified into the following [14]:

- if D is a fixed subset of $\mathbb{R}^d$ and $Z(s)$ is a random vector at $s \in D$, then we have geostatistical data;
- if D is a fixed collection of countably many points of $\mathbb{R}^d$ and $Z(s)$ is a random vector at $s \in D$, then we have lattice data;
- if D is a point process in $\mathbb{R}^d$ and $Z(s)$ is a random set, then we have spatial objects;
- if D is a point process in $\mathbb{R}^d$ and $Z(s)$ is a random vector at $s \in D$, then we have point patterns.

Diggle [9] defines a spatial point pattern as a set of points that are distributed within a designated region in space, where the points are presumed to have been generated by a stochastic mechanism. Examples include locations of seedlings in a section of a forest, houses in a neighbourhood or locations of ant nests in a specific region. Figure 1 below shows an example of a spatial point pattern in a square region.[1]

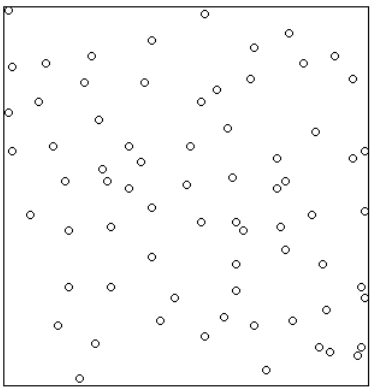


Figure 1: Locations of pine saplings in a Swedish forest [16, 21].

Suppose we want to create a continuous intensity function of the observed discrete point process. This is done by using kernel density estimation [9]. Kernel density estimation is a method of estimating the

[1] Acquired from the `spatstat` package in `R`

probability density function of a random variable in a nonparametric way. The first paper to deal with probability density estimation was by Rosenblatt [18] who also discussed the kernel estimator. Parzen [15] is also credited to have created density estimation independently.

Silverman [20] and Scott [19] gave detailed explanations of density estimation. Silverman also discussed many important applications of density estimation while Scott focused on the multivariate aspects of density estimation.

The `spatstat` package is a package in `R` for the statistical analysis of spatial point pattern data in two dimensions [4]. The package supports several functions including creation, manipulation, plotting, simulation and model-fitting of spatial point patterns. The `density.ppp` function is a function in `spatstat` that computes a kernel smoothed intensity function from a point pattern data set using kernel density estimation [1]. Figure 2 shows a kernel estimated intensity function for the spatial point pattern in Figure 1 that was applied using this function.

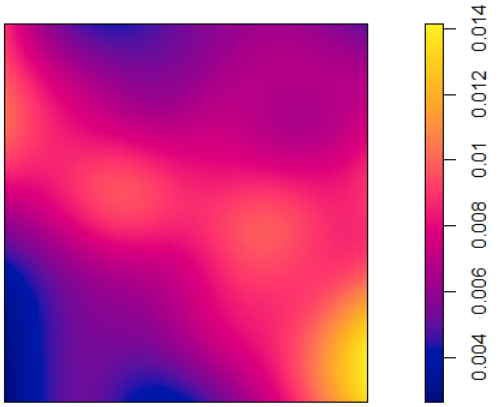


Figure 2: Kernel estimated intensity graph for the locations of pine saplings in a Swedish forest (Figure 1).

In [3], Baddeley and Turner give us a brief introduction to `spatstat`, which includes, among other things, an overview of available data sets, techniques in generating random point patterns as well as different ways of analyzing them. [1] is a detailed set of notes that Baddeley made for a workshop where he presented practical techniques in the statistical analysis of spatial point patterns. [2] is a book by Baddeley et al. that contains techniques for analysing spatial point patterns and is currently the main reference for the `spatstat` package.

The purpose of this research report will be to examine the mathematics of kernel density estimation in a spatial context that the function `density.ppp` uses, as well as to test the function under diverse conditions.

2 Kernel Density Estimation

Density estimation is a method that allows us to make an estimation of the density function of an observed sample of data. Suppose that we want to fit a probability density function on an observed set of data with an unknown probability density where we have no assumption of the nature of the distribution. Having no assumption of the nature of the distribution means that we will have a nonparametric approach to estimating the density function. Several methods and techniques exist for nonparametric estimation, as can be read in Silverman [20].

The most common and probably the first nonparametric density estimator is the histogram. Although the histogram is easily constructed and has an advantage of displaying the distribution of the observed data in a straightforward manner, it is not a reliable estimator in the sense that it loses a large amount accuracy due to the lack of smoothness in its shape. In a histogram, its shape is largely dependent on the width of the sub-intervals in which the whole data interval is divided and the end points of these sub-intervals. The kernel density estimator, another well known and widely used nonparametric density estimator, alleviates this problem by centering a kernel function at each data point and then summing them to create an overall smooth density estimate. It is widely used mainly because of its smoothing properties (see also [12]). A comparison of a histogram and a kernel density estimate is given in Figure 3, where we can see that the kernel density estimate is a more accurate estimate.

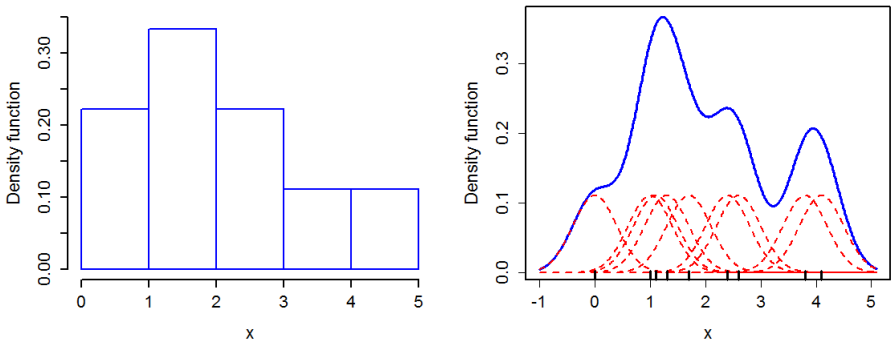


Figure 3: Comparison of a histogram and a kernel density estimate using the same data.

2.1 Kernel Density Estimation in a One-dimensional Space

The kernel density estimator was introduced in 1956 by Rosenblatt [18] and is defined by

$$\hat{f}(x) = \frac{1}{nh(n)} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h(n)}\right) \tag{1}$$

where $x_1, x_2, ..., x_n$ are independent identically distributed observations, K is a kernel function and $h(n)$ is the bandwidth for a chosen number of sub-intervals n that tends to 0 as n tends to infinity. The kernel

function is defined below.

Definition 1. A kernel is a real-valued integrable function K that is non-negative [20]. K should satisfy the following conditions:

- $\int_{-\infty}^{\infty} K(x)dx = 1.$
- $K(x) = K(-x)$ for all x.

These two conditions ensure that our kernel functions are symmetrical probability density functions.

Examples of kernel functions that `density.ppp` uses are the Gaussian, Epanechnikov and quartic (or biweight) functions. Table 1 presents each of these kernel functions' formulas and their graphs. Figures 5, 6 and 7 show density functions fitted with these kernels together with a histogram.[2]

The kernel estimator $\hat{f}(x)$ is a sum of kernel functions placed at each observed data point. The kernel function K will determine the shape of the density and the bandwidth $h(n)$ will determine the width of the kernel at each point [20]. The resulting shape of the kernel density estimate depends more on the choice of $h(n)$ than it does on K. It is important to choose an appropriate $h(n)$ as a value that is too small will result in a shape that is undersmoothed, and a value that is too large will result in a shape that is oversmoothed. Figure 4 below shows a comparison between a kernel density estimate that is undersmoothed, oversmoothed and one that is optimally smoothed respectively. Take note of how the size of $h(n)$ differs in each graph.

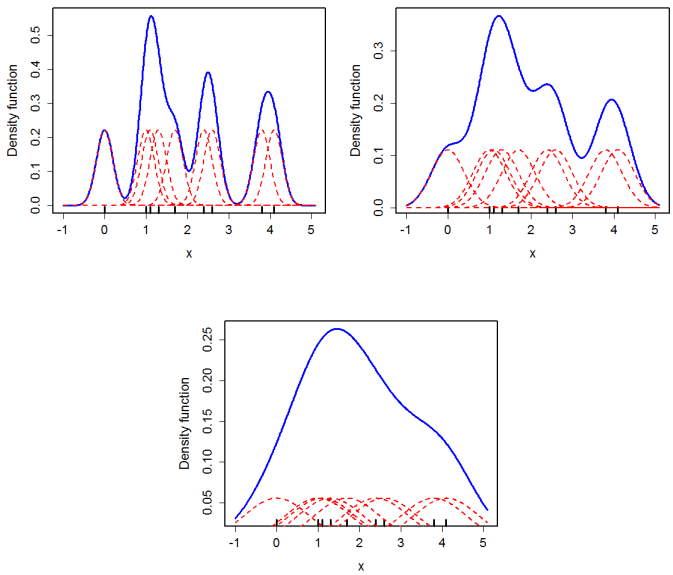


Figure 4: A comparison between a kernel density estimate that is undersmoothed, oversmoothed and one that is optimally smoothed.

[2]The data used to create these graphs was acquired from the `PlantGrowth` dataset in `R`.

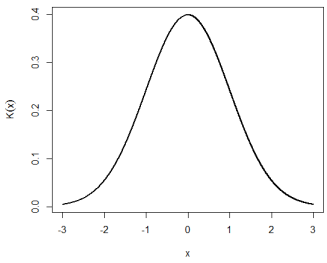
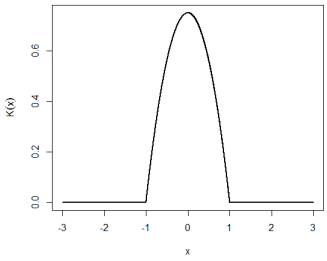
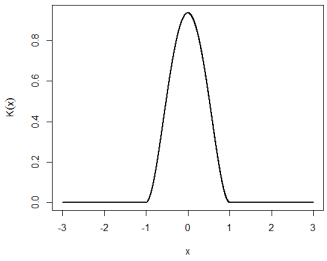
Kernel	$K(x)$	Graph
Gaussian	$K(x) = \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{x^2}{2}\right)}$	
Epanechnikov	$K(x) = \begin{cases} \frac{3}{4}(1-x^2) & \text{for } \lvert x\rvert \leq 1 \\ 0 & \text{otherwise} \end{cases}$	
Biweight	$K(x) = \begin{cases} \frac{15}{16}(1-x^2)^2 & \text{for } \lvert x\rvert \leq 1 \\ 0 & \text{otherwise} \end{cases}$	

Table 1: Some kernel functions.

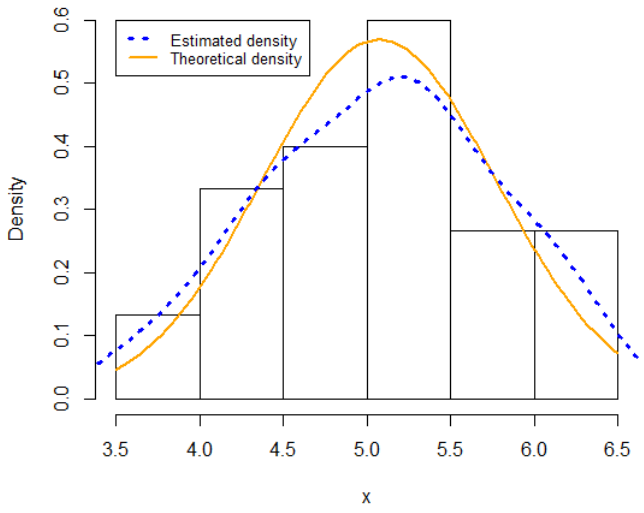


Figure 5: Density function fitted with the Gaussian kernel

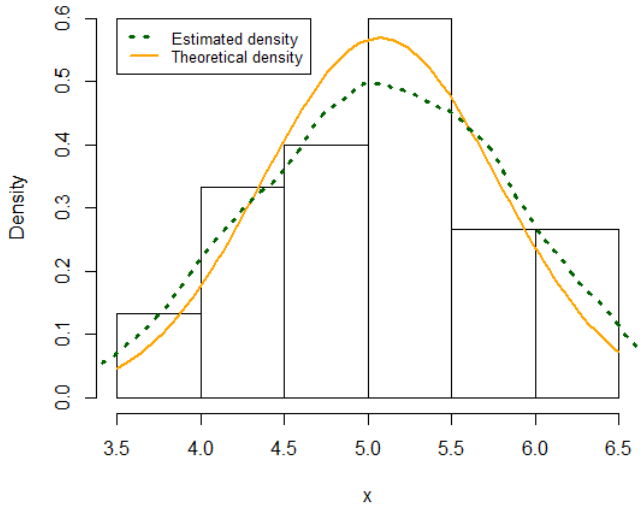


Figure 6: Density function fitted with the Epanechnikov kernel

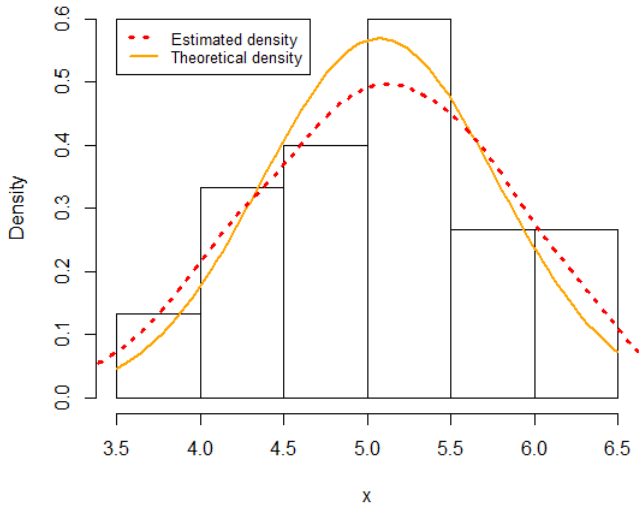


Figure 7: Density function fitted with the biweight kernel

2.1.1 Error functions for $\hat{f}(x)$

When studying the measure of difference between the density estimator $\hat{f}$ and the true density f at a single point, a commonly used measure is the mean square error (MSE), defined by

$$\mathrm{MSE}_x(\hat{f}) = E[(\hat{f}(x) - f(x))^2]. \tag{2}$$

By the standard properties of mean and variance,

$$\mathrm{MSE}_x(\hat{f}) = \Big(E[\hat{f}(x)] - f(x)\Big)^2 + \mathrm{var}(\hat{f}(x)), \tag{3}$$

which is the sum of the squared bias and the variance at x.

The first and most widely used method of measuring the global accuracy of $\hat{f}$ as an estimator of f, introduced by Rosenblatt [18], is the mean integrated square error (MISE) and is defined by,

$$\text{MISE}(\hat{f}) = \text{E}\left[\int (\hat{f}(x) - f(x))^2 dx\right]. \tag{4}$$

We can rewrite (4) as:

$$\begin{aligned}
\text{MISE}(\hat{f}) &= E\left[\int (\hat{f}(x) - f(x))^2 dx\right] \\
&= \int E[(\hat{f}(x) - f(x))^2] dx \\
&= \int \text{MSE}_x(\hat{f})\, dx \\
&= \int \left(E[\hat{f}(x)] - f(x)^2\right) dx + \int \text{var}(\hat{f}(x)) dx,
\end{aligned} \tag{5}$$

which is equal to the sum of the integrated square bias and the integrated variance at x [20].

Let $X_1, X_2, ..., X_n$ be independent and identically distributed random variables with common density f. Then for a kernel function K, we have that the expectation of $\hat{f}$ is equal to

$$\begin{aligned}
E[\hat{f}(x)] &= \frac{1}{nh} \sum_{i=1}^{n} E\left[K\left(\frac{x - X_i}{h}\right)\right] \\
&= \frac{1}{h} E\left[K\left(\frac{x - X_i}{h}\right)\right] \\
&= \frac{1}{h} \int K\left(\frac{x - y}{h}\right) f(y) dy
\end{aligned} \tag{6}$$

and the variance is given by

$$\begin{aligned}
\text{var}(\hat{f}(x)) &= \text{var}\left(\frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)\right) \\
&= \frac{1}{n^2} \text{var}\left(\frac{1}{h} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)\right) \\
&= \frac{1}{n} \text{var}\left(\frac{1}{h} K\left(\frac{x - X_i}{h}\right)\right) \\
&= \frac{1}{n}\left[E\left(\left[\frac{1}{h} K\left(\frac{x - X_i}{h}\right)\right]^2\right) - \left(E\left[\frac{1}{h} K\left(\frac{x - X_i}{h}\right)\right]\right)^2\right]
\end{aligned} \tag{7}$$

Suppose that the kernel K is a symmetric function satisfying

$$\int K(t)dt = 1, \int tK(t)dt = 0, \text{ and } \int t^2 K(t)dt = \sigma_k^2 > 0. \tag{8}$$

Then the bias is given by

$$\text{bias}_h(x) \approx \frac{1}{2}h^2 f''(x)\sigma_k^2. \tag{9}$$

(See [20] for the derivation). The integrated square bias is given by

$$\int \text{bias}_h(x)^2 dx \approx \frac{1}{4}h^4\sigma_k^4 \int f''(x)^2 dx. \tag{10}$$

A Taylor series approximation shows that the variance is approximately equal to

$$\text{var}(\hat{f}(x)) \approx \frac{1}{nh} f(x) \int K(t)^2 dt. \tag{11}$$

Integrating over x gives the approximation

$$\int \text{var}(\hat{f}(x)) dx \approx \frac{1}{nh} \int K(t)^2 dt. \tag{12}$$

From (10) and (12), the approximate mean square integrated error is given by

$$\text{AMISE}(\hat{f}) = \frac{1}{4}h^4\sigma_k^4 \int f''(x)^2 dx + \frac{1}{nh} \int K(t)^2 dt. \tag{13}$$

Comparing the two terms in the AMISE, we can see that eliminating the bias by using a small value for h increases the size of the integrated variance. On the contrary, using a large value for h will decrease the size of the variance while increasing the overall bias. The choice of h will always imply a trade-off between random error and bias [20].

One way of acquiring an ideal value for h is by minimizing the AMISE given in (13). Parzen [15] has shown that the ideal value for h in this context is equal to h^*, where

$$h^* = \left[\frac{\int K(t)^2 dt}{\sigma_k^4 \int f''(x)^2 dx}\right]^{\frac{1}{5}} n^{-\frac{1}{5}}. \tag{14}$$

From (14) we see that h^* will converge to 0 at a very slow rate as the sample size n increases.

If we substitute the value of h^* back into the AMISE in (13), it can be shown that the ideal approximate mean square integrated error (AMISE*) will be

$$\text{AMISE}^* = \frac{5}{4}\left[\sigma_k \int K(t)^2 dt\right]^{\frac{4}{5}} \left[\int f''(x)^2 dx\right]^{\frac{1}{5}} n^{-\frac{4}{5}}. \tag{15}$$

To obtain a small value for the MISE, we should choose a kernel function K with a small value of $\sigma_k \int K(t)^2 dt$. The kernel that minimizes $\int K(t)^2 dt$ subject to the constraints $\int K(t) dt = 1$ and $\int t^2 K(t) dt = 1$ is the Epanechnikov kernel. The efficiency of any other symmetric kernel K is obtained

by comparing it to the Epanechnikov kernel [20].

2.2 Kernel Density Estimation in a Two-dimensional Space

The bivariate kernel density estimator with kernel K and bandwidth h is defined by

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^2} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \tag{16}$$

where $\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$ and $\mathbf{x}_i = (x_{i1}, x_{i2})^{\mathrm{T}}$, $i = 1, 2, ..., n$. Here the kernel function $K(\mathbf{x})$ is a bivariate kernel satisfying

$$\int_{\mathbb{R}^2} K(\mathbf{x}) \mathrm{d}\mathbf{x} = 1.$$

It is also a symmetric unimodal probability density function. Examples of K include the standard bivariate normal density function

$$K(\mathbf{x}) = \frac{1}{2\pi} e^{(-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{x})} \tag{17}$$

and the bivariate Epanechnikov kernel

$$K(x) = \begin{cases} \frac{2}{\pi}(1 - \mathbf{x}^{\mathrm{T}}\mathbf{x}) & \text{if} \, \mathbf{x}^{\mathrm{T}}\mathbf{x} < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

To ensure the kernel placed at each data point is equally scaled in all directions, a single bandwidth h is used in (16). Sometimes it may be more appropriate to use a vector or matrix of bandwidths, such as in cases where the spread of the data points is greater in one direction than the others [20].

Figure 8 shows a sample data set in a two-dimensional space, its kernel density estimate and a 3D perspective of the density estimate. The bivariate Gaussian kernel with bandwidth $h = 1$ was used.

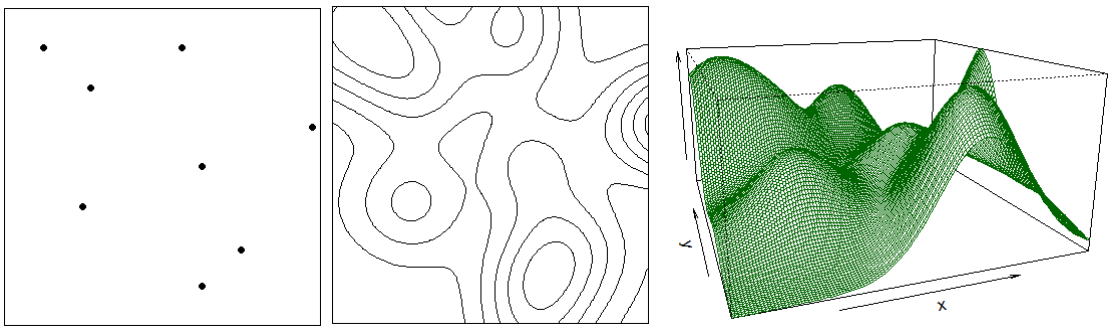


Figure 8: Kernel density estimate (middle) of a sample data set (left) and a 3D perspective (right).

The theoretical properties of the two-dimensional kernel density estimator can be derived in the same manner as in Section 2.1.1 [20]. Using the two-dimensional form of Taylor's theorem, we acquire the

approximations

$$\text{bias}_h(\mathbf{x}) \approx \frac{1}{2}h^2 \left(\int t_1^2 K(\mathbf{t}) \mathrm{d}\mathbf{t} \right) \nabla^2 f(\mathbf{x}) \tag{19}$$

and

$$\text{var}(\hat{f}(\mathbf{x})) \approx \frac{f(\mathbf{x}) \int K(\mathbf{t})^2 \mathrm{d}\mathbf{t}}{nh^2}. \tag{20}$$

We combine (19) and (20) to get the AMISE

$$\text{AMISE} = \frac{1}{2}h^4 \left(\int t_1^2 K(\mathbf{t}) \mathrm{d}\mathbf{t} \right)^2 \int (\nabla^2 f(\mathbf{x}))^2 \mathrm{d}\mathbf{x} + \frac{\int K(\mathbf{t})^2 \mathrm{d}\mathbf{t}}{nh^2}. \tag{21}$$

The ideal value for the bandwidth h that is acquired by minimizing the AMISE is given by h^*, where

$$h^* = \left[\frac{2 \int K(\mathbf{t})^2 \mathrm{d}\mathbf{t}}{n \left(\int t_1^2 K(\mathbf{t}) \mathrm{d}\mathbf{t} \right)^2 \left(\int (\nabla^2 f(\mathbf{x}))^2 \mathrm{d}\mathbf{x} \right)} \right]^{\frac{1}{6}}. \tag{22}$$

3 The `density.ppp` function

Intensity is a basic descriptive characteristic of a point process that is described as the average number of random points per unit area [2]. It is essential that the intensity of a point pattern be examined before any further data analysis can take place. There are parametric and nonparametric ways of examining the intensity of a point pattern. The intensity function $\lambda(u)$ of the point process that generates a point pattern data set can be estimated nonparametrically by kernel estimation. The `density.ppp` in `spatstat` function calculates this estimate. It is a method for the generic function `density`, where the result that `density.ppp` produces is not a probability density but an estimate of the intensity function. This chapter will look at the `density.ppp` in more detail.

Usage

The function is used in the following way:

```
density.ppp(x, sigma=NULL, ..., weights=NULL, edge=TRUE, varcov=NULL, at="pixels",
leaveoneout=TRUE, adjust=1, diggle=FALSE, se=FALSE, kernel="gaussian",
scalekernel=is.character(kernel), positive=FALSE)
```

where `x` is a point pattern object.

3.1 Kernel estimation

Edge effects

Diggle [9] states that edge-effects occur when an observed region A on which a spatial point pattern is observed is part of a larger region that contains the same underlying process. This causes the difficulty to take into account the possibility of unobserved events outside A interacting with the observed events within A. This creates a strong negative bias close to the boundary when creating an intensity function over the whole region, since the points outside the boundary do not contribute to the sum in the equation (23) below. The kernel estimators for the intensity function [2, 7, 8] are given by

$$\hat{\lambda}^{(1)}(u) = \sum_{i=1}^{n} \kappa(u - x_i) \tag{23}$$

$$\hat{\lambda}^{(2)}(u) = \frac{1}{e(u)} \sum_{i=1}^{n} \kappa(u - x_i) \tag{24}$$

$$\hat{\lambda}^{(3)}(u) = \sum_{i=1}^{n} \frac{1}{e(x_i)} \kappa(u - x_i) \tag{25}$$

for any spatial location u inside the window W, where $\kappa(u) = \frac{1}{h^2} K\left(\frac{u}{h}\right)$ with K a kernel function and

$$e(u) = \int_W \kappa(u - v) dv \tag{26}$$

is a correction for the bias caused by edge-effects.

The uniform correction (24) and Diggle's correction (25) are designed to address the problem of edge-effects that arise when a point process is observed inside a window [2]. The raw estimate (23) doesn't take edge-effects into account and thus should only be used in situations where there are no edge-effects.

The kernel estimators have a slight bias because of the smoothing that takes place in the intensity function [2]. We are going to understand the statistical properties of the kernel estimators using Campbell's formula. Let $f(u)$ be a function of a spatial location u, and let T be the random sum

$$T = \sum_i f(x_i)$$

of the value of f at each data point x_i in a point process $\mathbf{X}$. Campbell's formula states that

$$E\left[\sum f(x_i)\right] = \int_{\mathbb{R}^2} f(u)\lambda(u) du = 1 \tag{27}$$

where $\lambda(u)$ is the intensity function of $\mathbf{X}$.

Applying Campbell's formula to kernel estimation, let u be a fixed spatial location and let $f(v) =$

$\kappa(u-v)$ if v is in the window W and $f(v)=0$ if it's not. Then

$$\sum_i f(x_i) = \sum_i \kappa(u-x_i) = \hat{\lambda}^{(1)}(u).$$

By Campbell's formula (27)

$$E\left[\hat{\lambda}^{(1)}(u)\right] = E\left[\sum f(x_i)\right] = \int f(v)\lambda(v)dv = \int_W \kappa(u-v)\lambda(v)dv.$$

The expected value of $\hat{\lambda}^{(1)}(u)$ is not equal to to the true value of $\lambda(u)$. Now suppose that the true intensity value is a constant, say β, we get

$$E\left[\hat{\lambda}^{(1)}(u)\right] = \int_W \kappa(u-v)\beta dv = \beta \int_W \kappa(u-v)dv = \beta e(u)$$

where $e(u)$ is the same as (26). From this we can see that the uniformly corrected estimator $\hat{\lambda}^{(2)}(u)$ is unbiased when the intensity is constant, i.e. if $\lambda(u) \equiv \lambda$, then $E\left[\hat{\lambda}^{(2)}(u)\right] = \lambda$. In general, $\hat{\lambda}^{(2)}(u)$ is a biased estimator of $\lambda(v)$ since

$$E\left[\hat{\lambda}^{(2)}(u)\right] = \frac{1}{e(u)} \int_W \kappa(u-v)\lambda(v)dv,$$

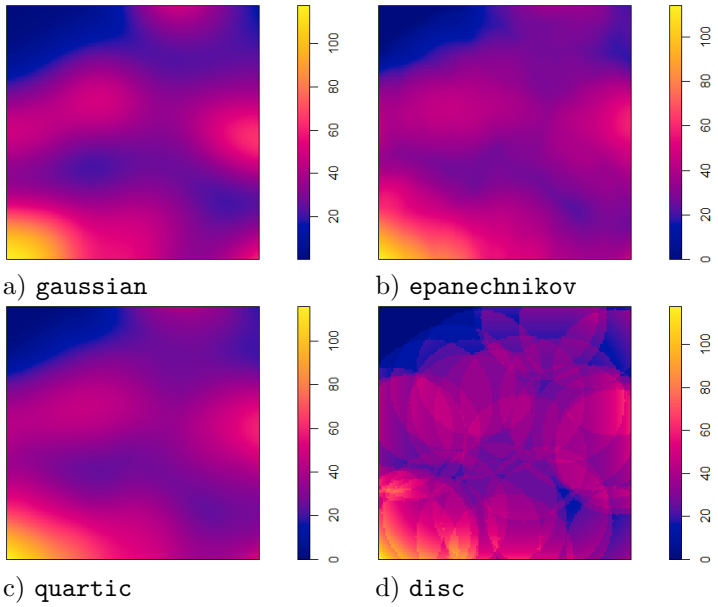
which is a smoothed version of the true intensity $\lambda(u)$.

Passing the argument `edge=TRUE` to `density.ppp` will correct for edge-effects in one of the following ways:

- If `diggle=FALSE` (default) then the uniform correction is used.
- If `diggle=TRUE` then Diggle's correction is used. This method has a better performance but is slower to compute [2].

`kernel`

The argument `kernel` selects the kernel that should be used. The current options are `"gaussian"`, `"epanechnikov"`, `"quartic"` or `"disc"`, or a pixel image, or an `R` function. The default is the convolution of the isotropic Gaussian kernel with standard deviation `sigma`. Figure 9 displays the density estimate of the same data set for different kernel options.



a) `gaussian` b) `epanechnikov`

c) `quartic` d) `disc`

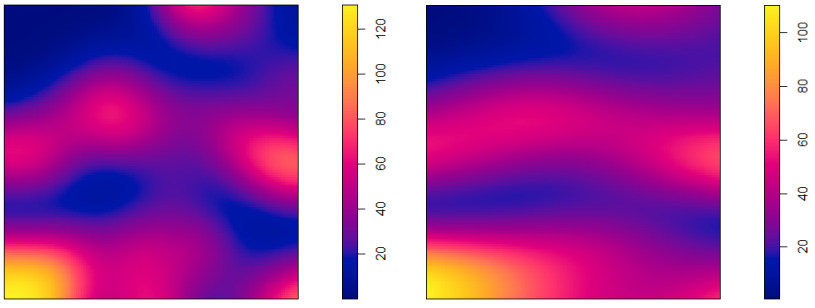
Figure 9: Density estimates with different options for `kernel`.

3.2 Bandwidth selection

`sigma`

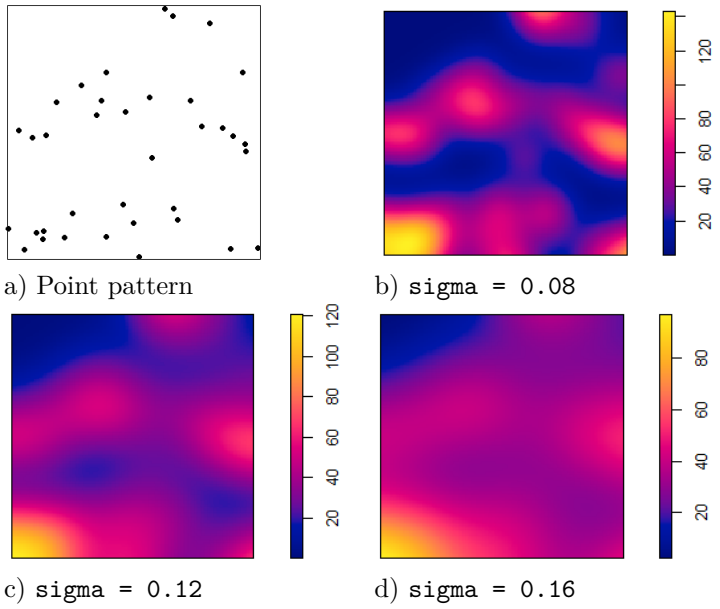
The standard deviation of the kernel is the smoothing bandwidth h [2]. The argument `sigma` determines the value of the bandwidth. It can be a single numerical value, a pair of numerical values (one specifying the standard deviation in the x direction and the other in the y direction), or a function that will automatically calculate the bandwidth. If `sigma` is a numerical value, it will be taken as the standard deviation of the isotropic Gaussian kernel. By default, if not specified, the value of `sigma` will be equal to one-eighth of the shortest side of the enclosing rectangle. This rule of thumb may produce unsatisfactory results in most cases. Figure 10 shows a comparison of density estimates with different bandwidth values.

In Figure 11, we see a comparison of two density estimates where where one has a numerical value for `sigma` and the other a pair of numerical values given by `vec` $= \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix}$.



a) `sigma = 0.1` b) `sigma = vec`

Figure 11: Density estimates with one and two numerical values for `sigma`.



a) Point pattern b) `sigma = 0.08`

c) `sigma = 0.12` d) `sigma = 0.16`

Figure 10: Density estimates with different bandwidths.

Various functions exist to calculate the value of `sigma`. These include `bw.diggle` for Diggle and Berman's mean square error cross-validation method and `bw.ppl` for the likelihood cross-validation method. Other functions are based on a fast rule of thumb, including `bw.scott` for Scott's rule of thumb and `bw.frac` for a fast rule of thumb for bandwidth selection based on the window's geometry. We look at this four functions in more detail below.

`bw.diggle`

In this method, the bandwidth h is selected so that it minimises the mean-square error criterion

$$\mathrm{MSE}(h) = \lambda_2(0) + \lambda\left(1 - 2\lambda K(h)\right) / \left(\pi h^2\right) + \left(\pi h^2\right)^{-2} \int\int \lambda_2\left(\|x - y\|\right) dydx$$

defined by Diggle [7], on the assumption that the underlying point process is a stationary isotropic Cox process with intensity $\lambda = \mu$ and second-order intensity $\lambda_2(u) = \gamma(u) - \mu^2$ where μ and $\gamma(u)$ are the expectation and covariance of the driving intensity of a Cox process respectively. Also, πh^2 is the area of a circle of radius h, $\|\cdot\|$ is the Euclidean distance and $K(\cdot)$ in this case is known as the K-function of a stationary process and can be defined as

$$K(t) = 2\pi\lambda^{-2} \int_0^t \lambda_2(s) s ds$$

and should not be confused with a kernel function.

The `bw.diggle` function then uses the method from [5] to calculate the quantity

$$\mathrm{M}(h) = \frac{\mathrm{MSE}(h) - \lambda_2(0)}{\lambda^2}.$$

The result is the selected bandwidth. It should be noted that the value returned by `bw.diggle` is equal to $\frac{r}{2}$, where r is the bandwidth described in [5, 9]. This adjustment is required to equate the variances of the kernels, since the kernel used in those references is a uniform density on a circle with radius r, while the kernel in `density.ppp` is the isotropic Gaussian density with standard deviation `sigma`. Thus `sigma` $= \frac{r}{2}$. Figure 12 is a density estimate where the bandwidth is computed using `bw.diggle`.

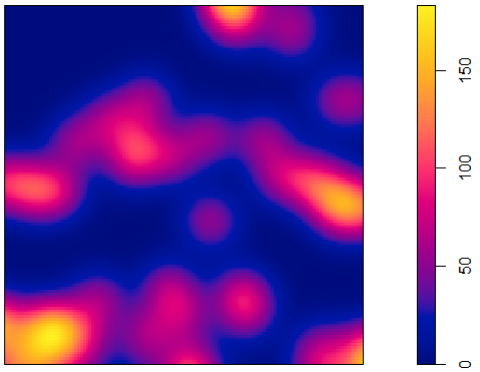


Figure 12: Density estimate with bandwidth computed using `bw.diggle`.

`bw.ppl`

The bandwidth h is selected to maximise the likelihood cross validation criterion, for the point process, defined by

$$\mathrm{LCV}(h) = \sum_{i}^{n} log\hat{\lambda}_{-i}(x_i) - \int_{W} \hat{\lambda}(u)du$$

where $\hat{\lambda}_{-i}(x_i)$ is the leave-one-out estimate (see Section 3.3) and $\hat{\lambda}(u)$ is the estimate of the intensity at u [13].

The value returned by $\mathrm{LCV}(h)$ is calculated for a specified number of h values between a specified range (these specifications are optional). In Figure 13, a density estimate is given where the bandwidth is computed using `bw.ppl` where h is searched within the range $(0.01, 0.2)$.

`bw.scott`

The bandwidth h is calculated using Scott's rule of thumb [19] given by

$$\hat{h}_i = \hat{\sigma}_i n^{-\frac{1}{6}}$$

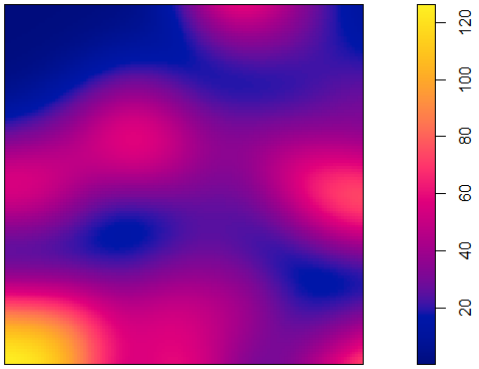


Figure 13: Density estimate with bandwidth computed using `bw.ppl` for the range $(0.01, 0.2)$.

in $\mathbb{R}^2$ for $i = 1, 2$, where $\hat{\sigma}_i$ is an estimate of the standard deviation of a bivariate normal distribution.

Scott's rule of thumb produces a larger bandwidth than `bw.diggle`, and it is beneficial for estimating trends that are gradual. The value returned is a vector of two numerical values, one for the x direction and one for the y direction. In Figure 14 the bandwidth for the density estimate is computed using `bw.scott`. Notice how the bandwidth is large.

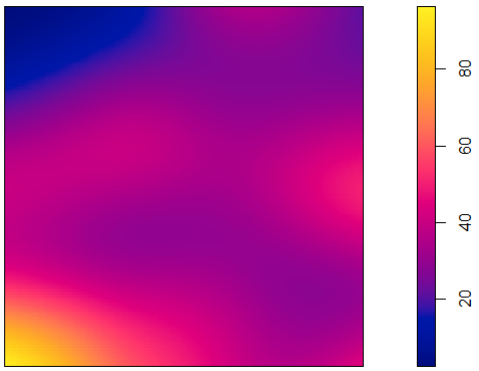


Figure 14: Density estimate with bandwidth computed using `bw.scott`.

`bw.frac`

The bandwidth h is selected so that it is equal to a quantile of the distance between two independent points in the window, that are uniformly distributed. The value returned is r such that $F(r) = f$, where $F(r)$ is the cumulative distribution of the distance between the two points and f is the probability of the quantile (that can be specified). In Figure 15 the bandwidth is computed using `bw.frac` with $f = 0.05$.

In summary, here is a list of available options for `sigma`:

- a numerical value
- a pair of numerical values
- a function such as:

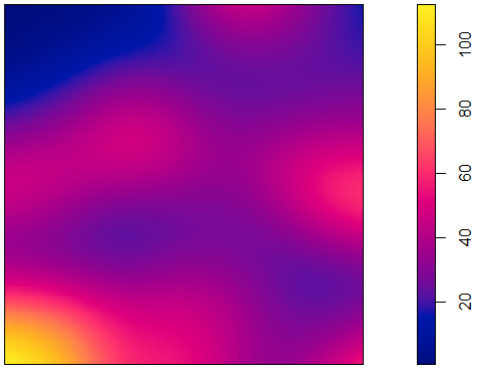
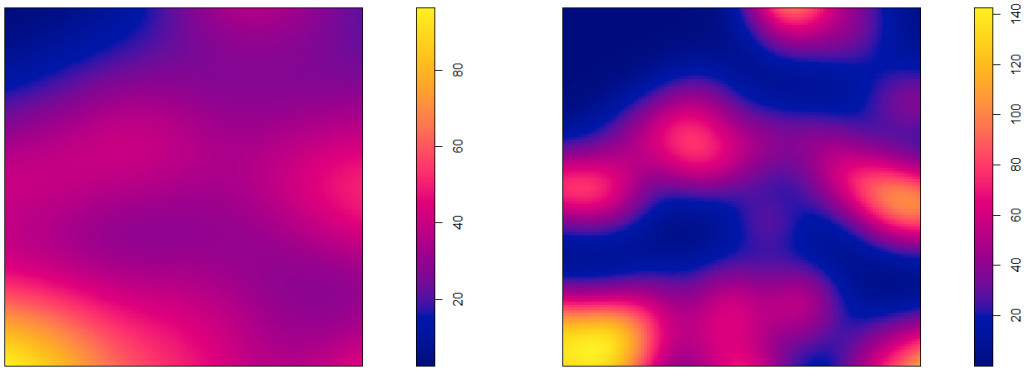


Figure 15: Density estimate with bandwidth computed using `bw.frac` with $f = 0.05$.

- `bw.diggle`
- `bw.scott`
- `bw.ppl`
- `bw.frac`

`adjust`

The bandwidth calculated by any of the methods above can easily be adjusted using the argument `adjust`, which is a numerical value that is multiplied with the bandwidth. For example, `adjust=2` will multiply the value of `sigma` by 2. In Figure 16, we've adjusted the bandwidth for the density estimate in Figure 14 by 0.5.



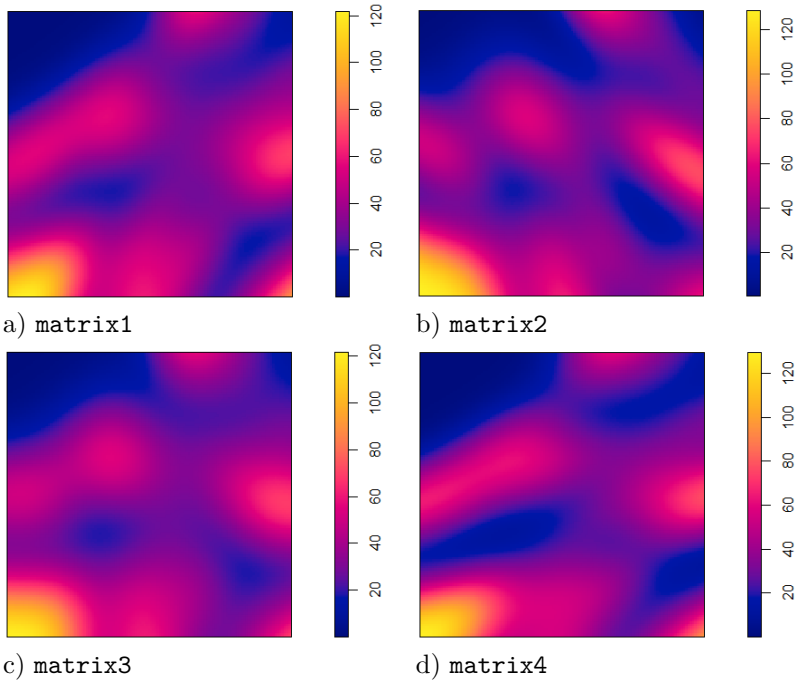
a) No adjustment b) `adjust = 0.5`

Figure 16: Density estimate with bandwidth computed using `bw.scott` with an adjustment of 0.5.

`varcov`

The kernel can be specified to be any Gaussian kernel by giving a variance-covariance matrix in the `varcov` argument. If the argument `varcov` is used, then `sigma` cannot be used as well. As can be seen in

Figure 17, different variance-covariance matrices can give different looking intensity estimates. In Figure 17, the density estimates are created using the following variance-covariance matrices for the `varcov` argument: `matrix1` $= \begin{bmatrix} 0.01 & 0.005 \\ 0.005 & 0.015 \end{bmatrix}$, `matrix2` $= \begin{bmatrix} 0.01 & -0.005 \\ -0.005 & 0.015 \end{bmatrix}$, `matrix3` $= \begin{bmatrix} 0.01 & 0 \\ 0 & 0.015 \end{bmatrix}$, and `matrix4` $= \begin{bmatrix} 0.015 & 0 \\ 0 & 0.01 \end{bmatrix}$.

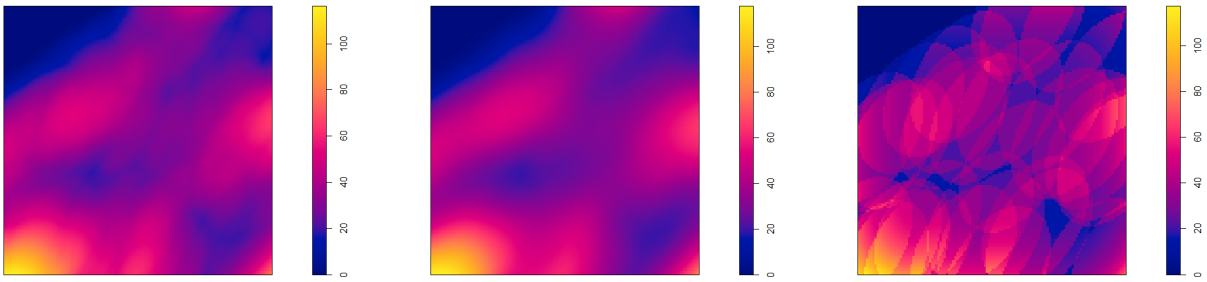


a) `matrix1` b) `matrix2`

c) `matrix3` d) `matrix4`

Figure 17: Density estimates with different matrices in the `varcov` argument using the `gaussian` kernel.

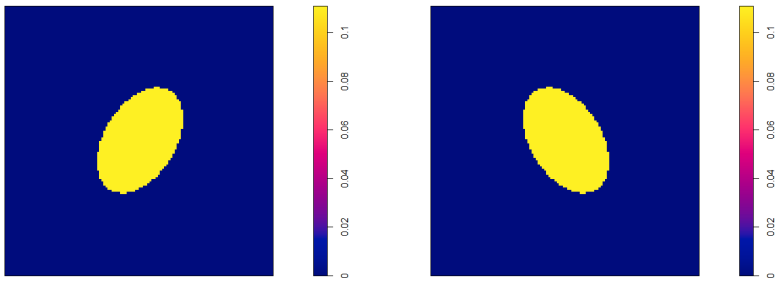
It should be noted that the `varcov` argument can also be used when `kernel` is a non-Gaussian kernel. We can see this in Figure 18, where we used `matrix1` as above.



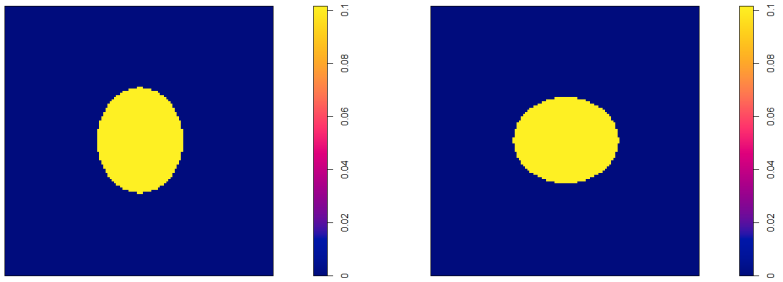
a) `epanechnikov` b) `quartic` c) `disc`

Figure 18: Using the `varcov` argument for the `epanechnikov`, `quartic` and `disc` kernels respectively.

To test how this works, we are first going to use the argument with the above matrices for a disc kernel on a single point. This is shown in Figure 19.



a) `matrix1` b) `matrix2`

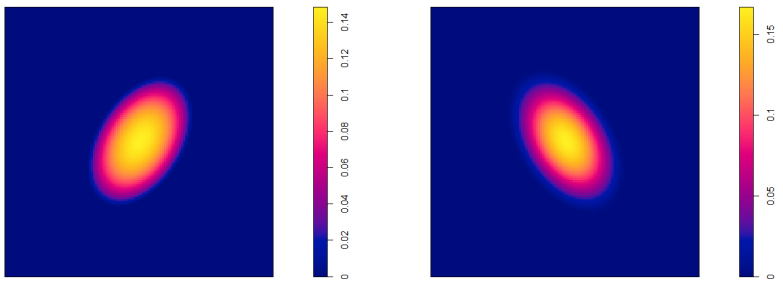


c) `matrix3` d) `matrix4`

Figure 19: Using the `varcov` argument for the `disc` kernel on a single point.

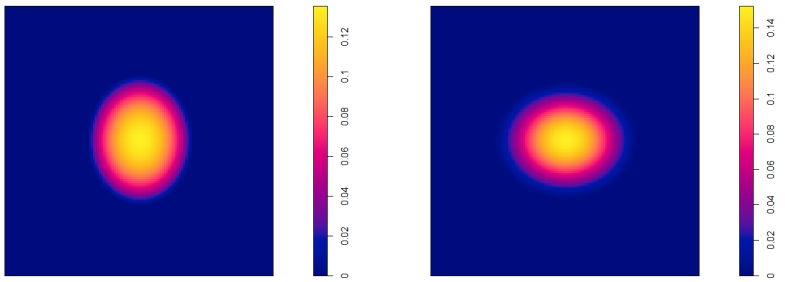
This result gives us a clearer insight on how `varcov` works. A Gaussian kernel is placed on top of the point so that what we see is the base of the kernel (an ellipse) on the window. The variance-covariance matrix affects the distribution of points surrounded by the ellipse. In `matrix1`, the covariances are positive, indicating that the data has a positive linear relationship; `matrix2` on the other hand, has negative covariances and hence a negative linear relationship is observed; in `matrix3`, the relationship is neither positive nor negative, and that is why the ellipse is not tilted in any direction; and `matrix4` yields a similar result to `matrix3` but the only difference is that the ellipse is wide in the x-direction due to a larger variance in that direction.

If we use the `epanechnikov` or `quartic` kernels, we will get the same results, as can be seen in Figures 20 and 21.



a) `epanechnikov` with `matrix1` b) `quartic` with `matrix2`

Figure 20: Using the `varcov` argument for the `epanechnikov` and `quartic` kernels on a single point.

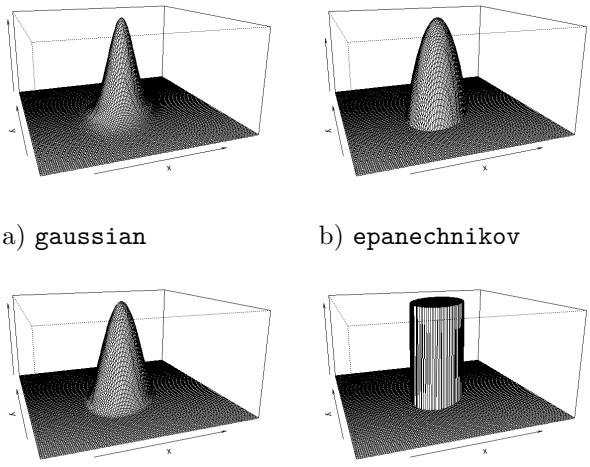


a) `epanechnikov` with `matrix3` b) `quartic` with `matrix4`

Figure 21: Using the `varcov` argument for the `epanechnikov` and `quartic` kernels on a single point.

This observation relieves any confusion on how `varcov` (and even `sigma`) works. Figures 20 and 21 tell us that `varcov` works the same way irrespective of the kernel specified and that it is just a variance-covariance matrix that will create the two-dimensional shape of the base of a Gaussian kernel on the window, and then any specified kernel's shape will be placed on top of this base, so to speak.

This result is further confirmed by looking at a 3D perspective of the different kernels using `matrix1` in Figure 22.



a) `gaussian` b) `epanechnikov`

c) `quartic` d) `disc`

Figure 22: A 3D perspective of the four different kernels on a single point.

Notice how the only difference is the shape of the kernel while the base and everything else remains the same.

Relationship between `varcov` and `sigma`

The value of `sigma` is equivalent to `varcov = diag(rep(sigma^2,2))`, where `rep(x,times)` replicates `x` for a number of times specified by `times`, and `diag(x)` returns a diagonal of `x`.

For example, if `sigma` $= 0.14$ then `varcov` $= \begin{bmatrix} 0.0196 & 0 \\ 0 & 0.0196 \end{bmatrix}$ or if `sigma` $= \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix}$ then `varcov` $= \begin{bmatrix} 0.04 & 0 \\ 0 & 0.01 \end{bmatrix}$.

`scalekernel`

Setting `scalekernel=TRUE` will rescale the kernel with the bandwidth that is given by `sigma` or `varcov`. This will be done automatically when `kernel` is a character string. When `kernel` is a function or pixel image, the default behaviour is `scalekernel=FALSE`, which will ignore `sigma` and `varcov`.

Standard errors (`se`)

Standard errors and confidence intervals for the intensity function can be calculated by setting `se=TRUE`.

3.3 Estimation of intensity at the data points

`leaveoneout`

Sometimes calculating the intensity $\lambda(x_i)$ at each data point x_i is required. Passing the argument `at="points"` to `density.ppp` will compute intensity estimates at the data points. The estimates $\hat{\lambda}^{(2)}(x_i)$ and $\hat{\lambda}^{(3)}(x_i)$ have large positive biases because of the term $\kappa(u - x_i) = \kappa(x_i - x_i) = \kappa(0)$ that appears in the sum in (24) and (25) [2]. The `leaveoneout` argument deals with this problem by estimating $\lambda(x_i)$ over all the data points except x_i:

$$\hat{\lambda}^{(2)}_{-i}(x_i) = \frac{1}{e(x_i)} \sum_{j \neq i}^{n} \kappa(x_i - x_j) \tag{28}$$

$$\hat{\lambda}^{(3)}_{-i}(x_i) = \sum_{j \neq i}^{n} \frac{1}{e(x_j)} \kappa(x_i - x_j). \tag{29}$$

The default is `leaveoneout=TRUE`.

4 Application

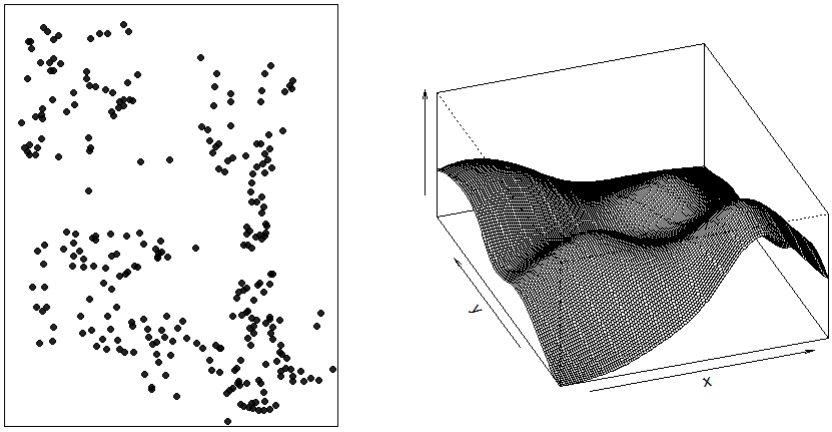
4.1 Introduction

In Section 2 we had an in-depth look at kernel density estimation and in Section 3 we studied the `density.ppp` function in detail and how it uses kernel density estimation to create an intensity function of a point process. In this section, we are going to create intensity functions of point patterns derived from actual real-world data. Our data consists of households in 5 different villages from the Mara province in

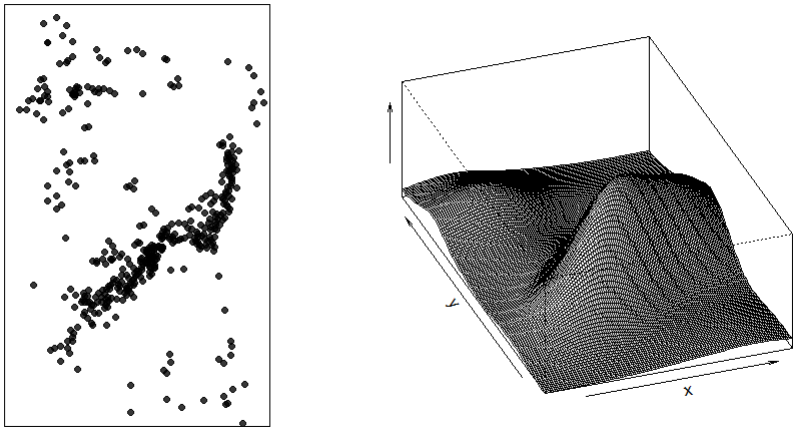
Northern Tanzania[3,4]. The households of these villages will be the points inside the window. In Section 4.2 we are going to apply the `density.ppp` function on these villages using a standard rectangular window, and then in Section 4.3 we are going to test the function under window effects, specifically the `convexhull` argument with edge effects.

4.2 Intensity for each village with standard window

The villages will be studied are Buchanchari, Kitembere, Magatini, Monuna and Nyiberekera and their point patterns are given in Figures 23 and 24.



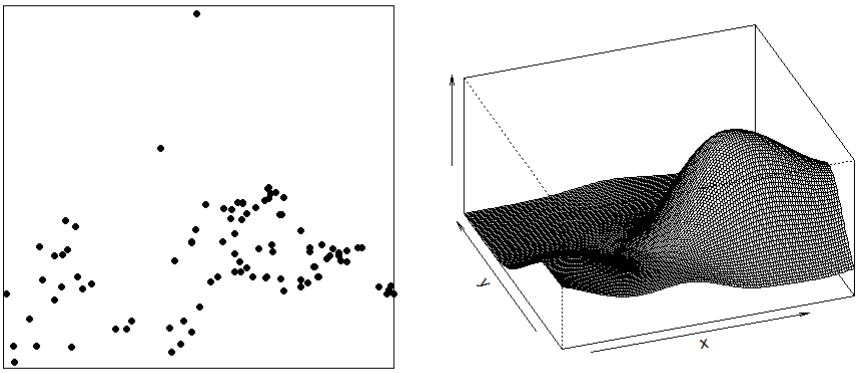
a) Buchanchari



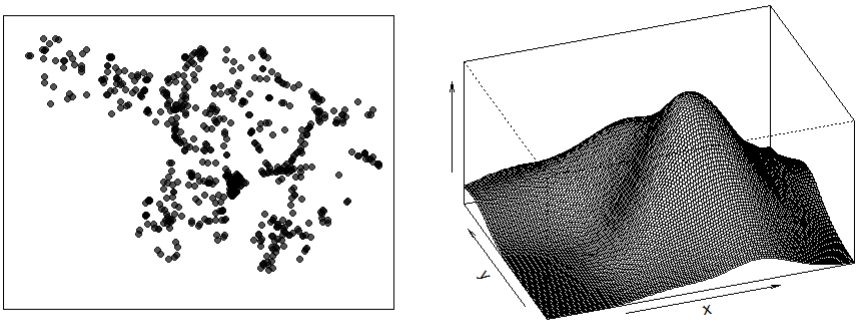
b) Kitembere

Figure 23: Point patterns of the different villages with 3D perspectives.

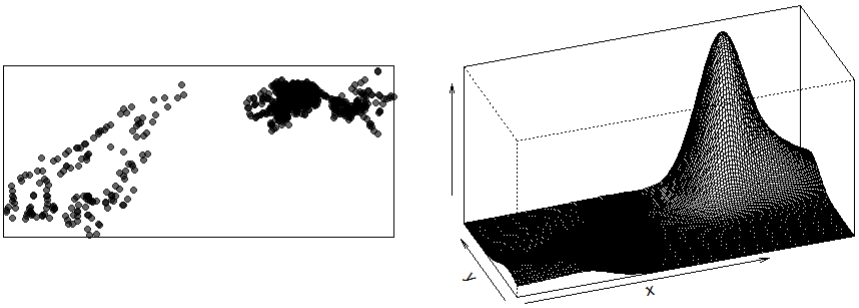
[3]http://www.gla.ac.uk/researchinstitutes/bahcm/staff/katiehampson/
[4]http://www.katiehampson.com/#intro



c) Magatini



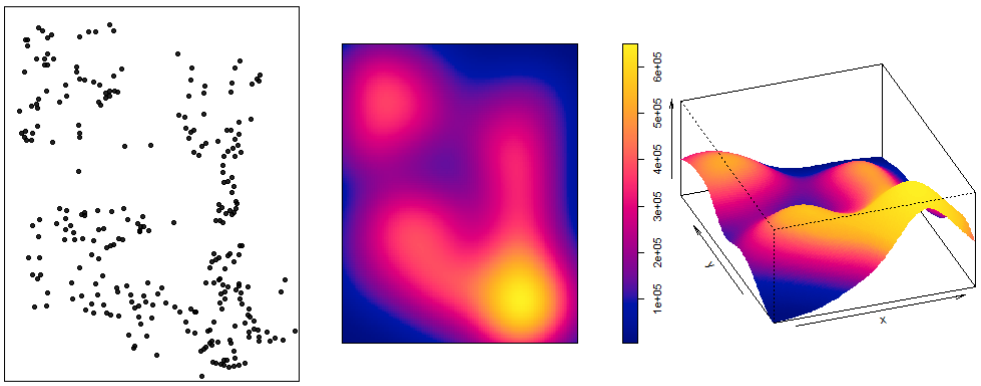
d) Monuna



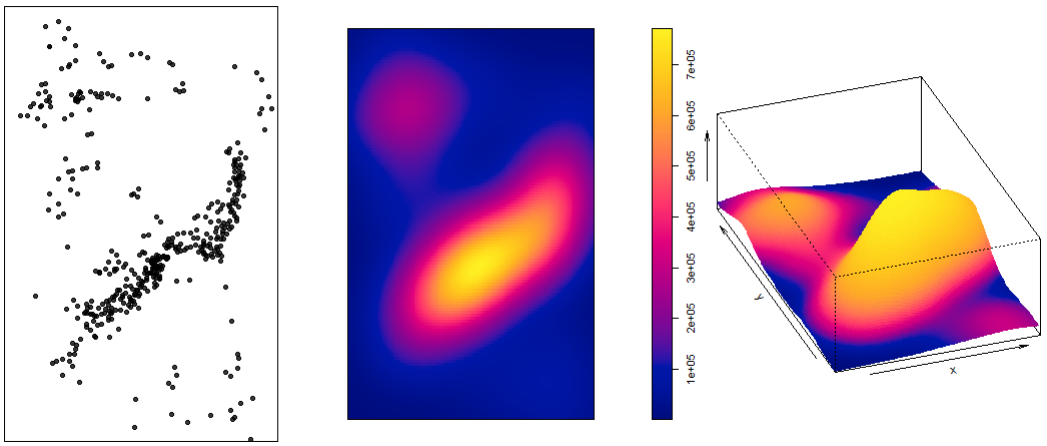
e) Nyiberekera

Figure 24: Point patterns of the different villages with 3D perspectives.

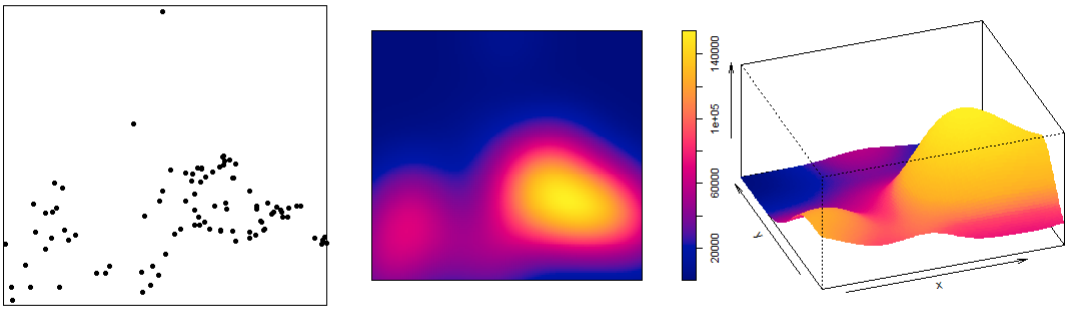
The `density.ppp` function will be applied to compute intensity functions of the villages' point patterns. These can be seen in Figures 25 and 26 where the `gaussian` kernel is used with Diggle's correction for edge effects.



a) Buchanchari

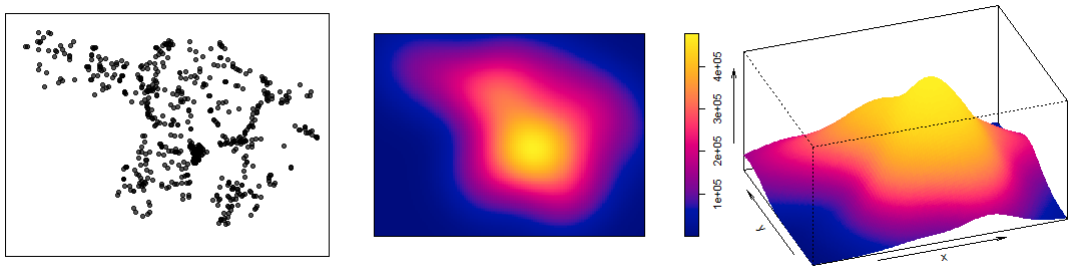


b) Kitembere

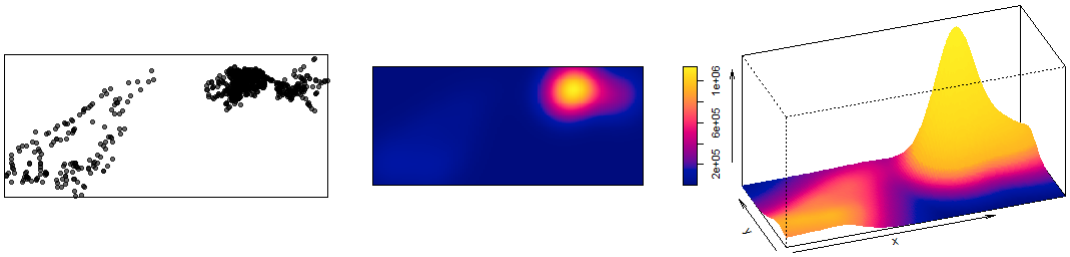


c) Magatini

Figure 25: Point patterns of the different villages and their intensity functions using the `gaussian` kernel with the default bandwidth selection.



d) Monuna



e) Nyiberekera

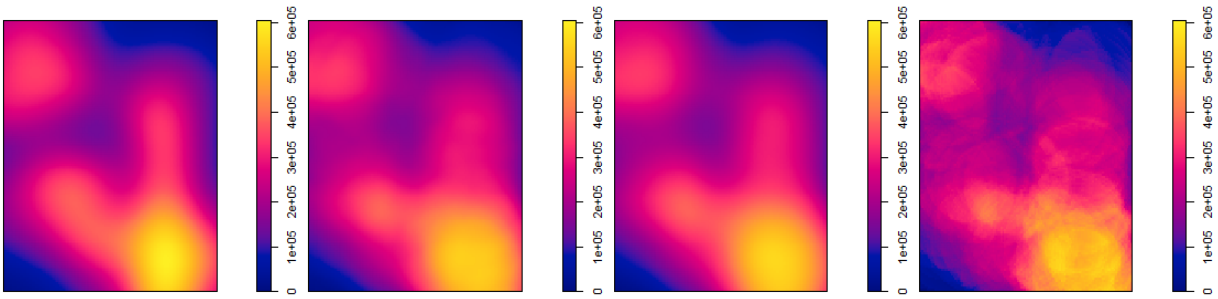
Figure 26: Point patterns of the different villages and their intensity functions using the `gaussian` kernel with the default bandwidth selection.

From the intensity functions we notice how the closeness of the points in a window affects the appearance of the intensity function. For example, if you look at the intensity of Buchanchari, you will notice that the points cover the entire window, creating an appropriate intensity function that gives an almost accurate representation of how the households are spread out across the village. In contrast, if we compare Buchanchari to Kitembere or Nyiberekera we see distinct results. In the latter point patterns, we observe clusters of points with little or no space between them and the clusters are surrounded by points that are more separated. The resulting intensity function shows high intensities in the regions of the clusters. In fact so high that the surrounding points have been smoothed out and do not have any significant contribution to the intensity estimate and they don't even appear in the intensity function.

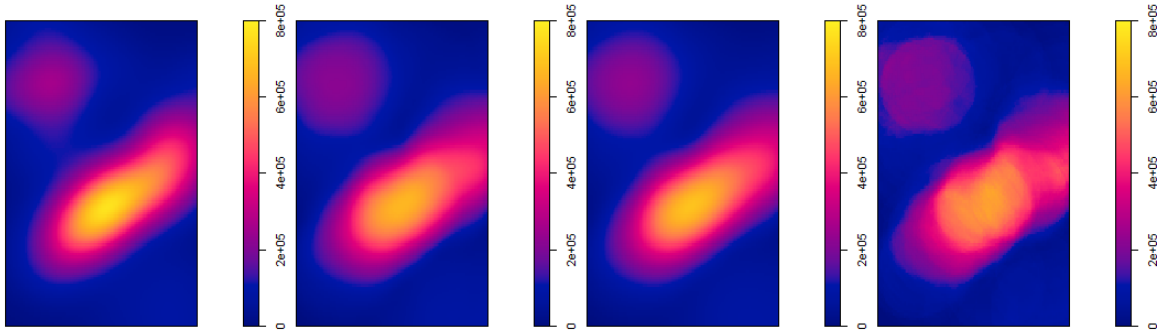
Thus we see that the intensity functions can be misleading because they are not accurate representations of how the households are spread out across the villages.

Intensity functions with different kernels

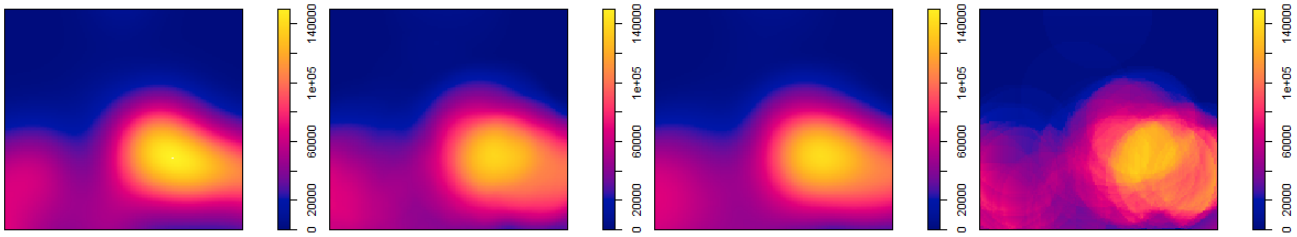
In Figures 27 and 28 it can be seen how the different intensities appear with the different types of kernels with no correction for edge effects. The intensity functions fitted with the `disc` kernel look visibly different from the intensity functions fitted with the other kernels. From this it can seen that these other kernels produce more favorable (if not superior) results than the `disc` kernel with regards to how we expect an intensity function to look like. The `gaussian`, `epanechnikov` and `quartic` kernels look similar to each other with slight differences, the most obvious one being that the `gaussian` kernel has the most intensity.



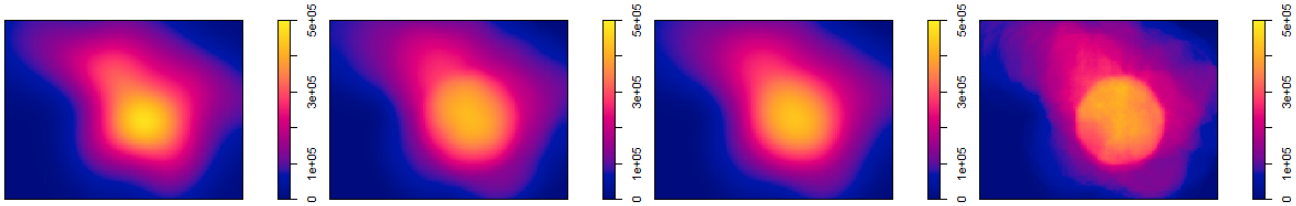
a) Buchanchari



b) Kitembere

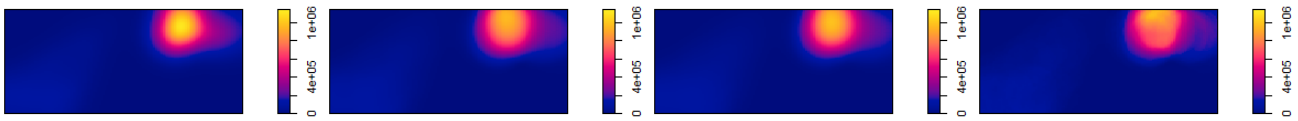


c) Magatini



d) Monuna

Figure 27: Intensity functions of the villages with the `gaussian`, `epanechnikov`, `quartic`, and `disc` kernels respectively without edge effects.

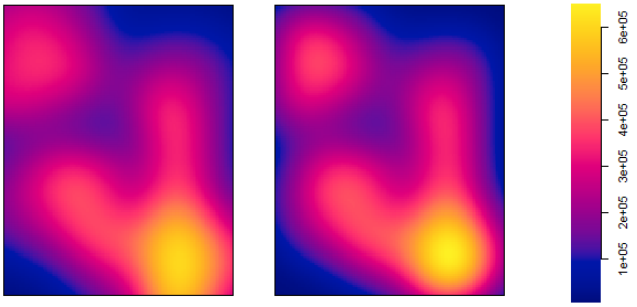


e) Nyiberekera

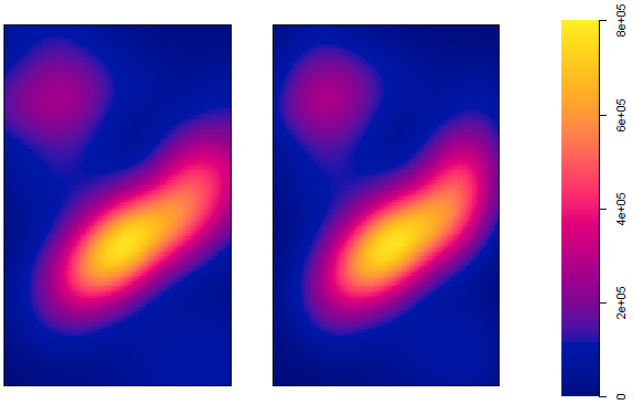
Figure 28: Intensity functions of Nyiberekera with the `gaussian`, `epanechnikov`, `quartic`, and `disc` kernels respectively without edge effects.

Intensity functions with different options for edge effect correction

In Figures 29 and 30 it can be seen how the different intensities appear with and without Diggle's correction for edge effects. Notice how the negative bias caused by edge effects is corrected so that the intensity functions only take into account the points within the window.

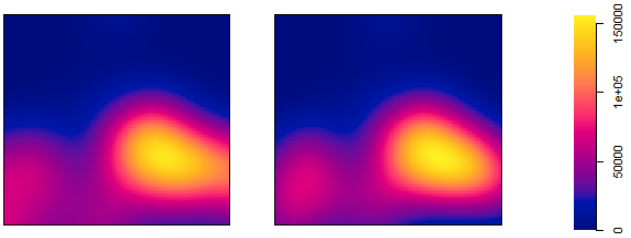


a) Buchanchari

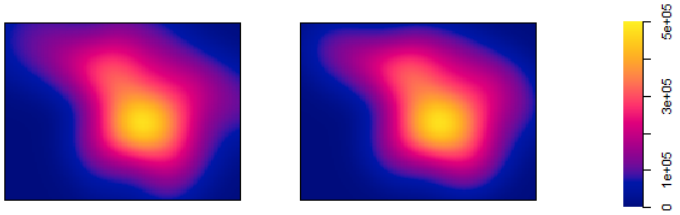


b) Kitembere

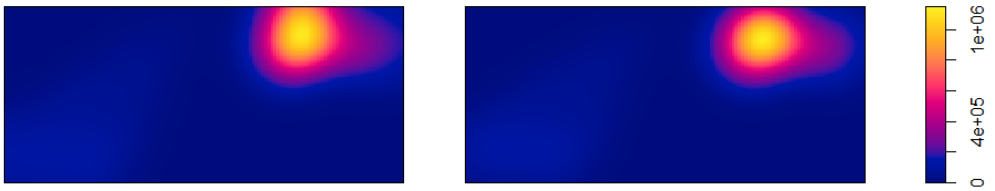
Figure 29: Intensity functions of the villages with `diggle = FALSE` and `diggle = TRUE` respectively.



c) Magatini



d) Monuna

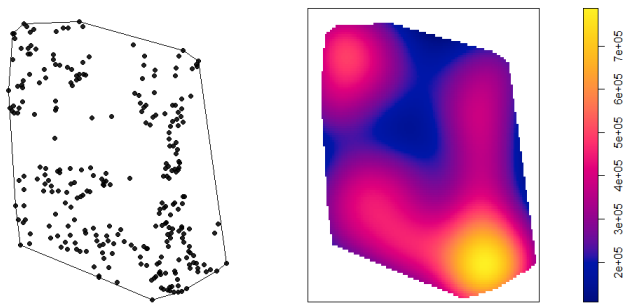


e) Nyiberekera

Figure 30: Intensity functions of the villages with `diggle = FALSE` and `diggle = TRUE` respectively.

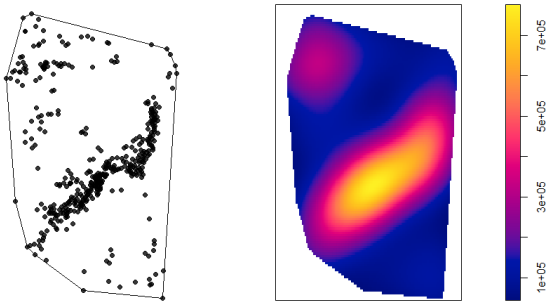
4.3 Intensity for each village with `convexhull` window

In this section the same approach to Section 4.2 is used, but this time changing the window effect to `convexhull` to see how this change will affect the fitted intensity functions. We observe this in Figures 31 and 32 where the `gaussian` kernel was used with Diggle's correction for edge effects.

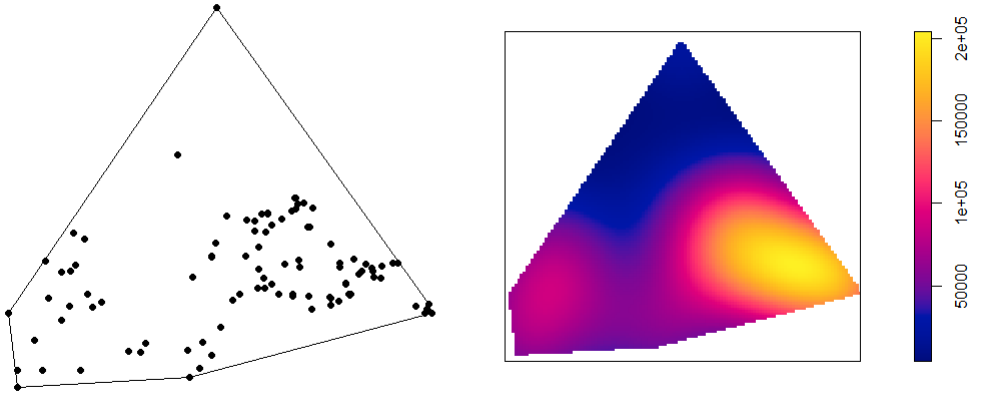


a) Buchanchari

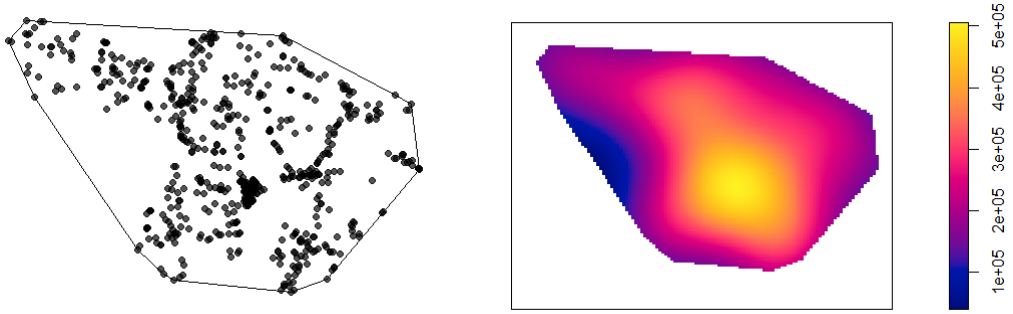
Figure 31: Point patterns of the different villages and their intensity functions with a `convexhull` window.



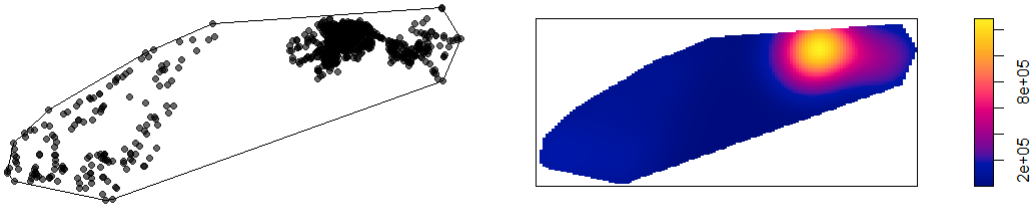
b) Kitembere



c) Magatini



d) Monuna



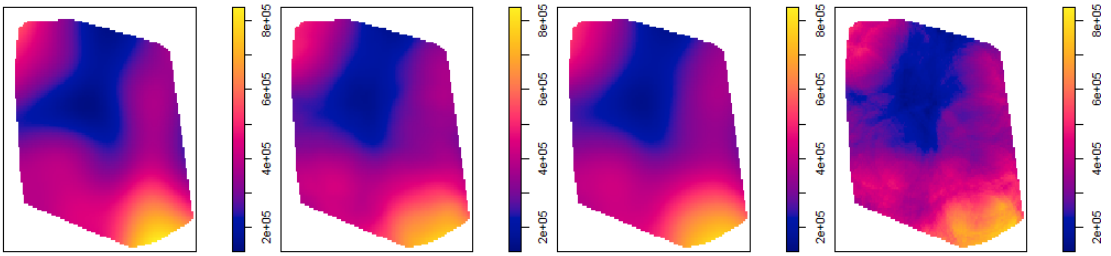
e) Nyiberekera

Figure 32: Point patterns of the different villages and their intensity functions with a `convexhull` window.

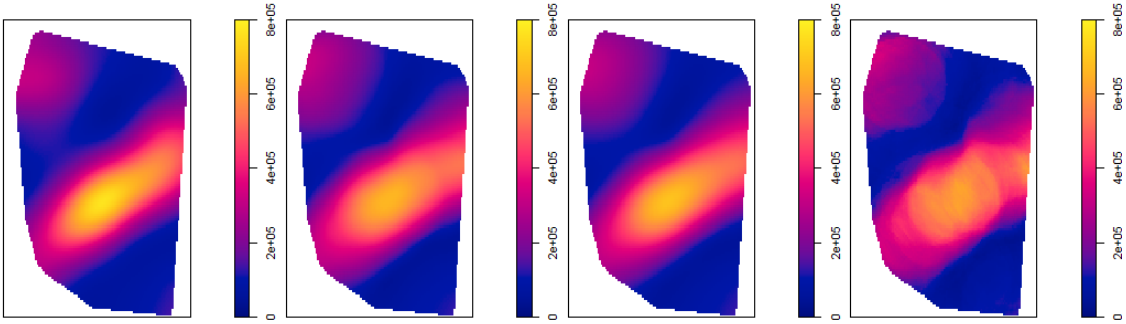
Here we see that changing the window effect to `convexhull` does not significantly affect the appearance of the intensity function.

Intensity functions with different kernels

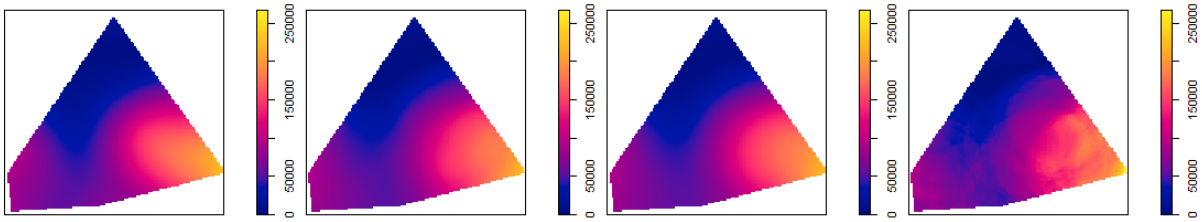
In Figures 33 and 34 the four different kernels without edge effect corrections are applied. We get similar looking results to the intensity functions created with a standard window.



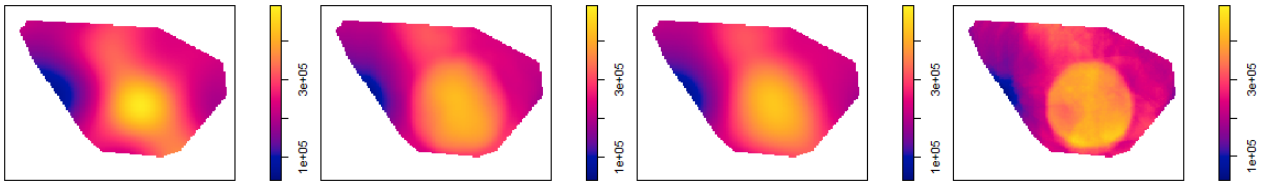
a) Buchanchari



b) Kitembere

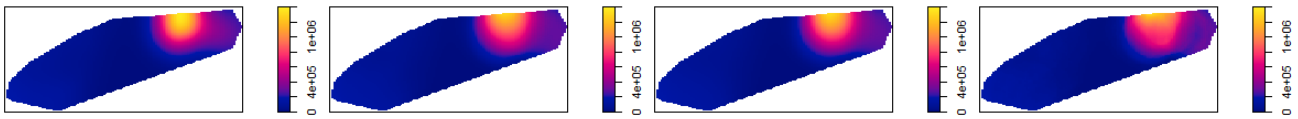


c) Magatini



d) Monuna

Figure 33: Intensity functions of the villages with the `gaussian`, `epanechnikov`, `quartic`, and `disc` kernels respectively without edge effects.

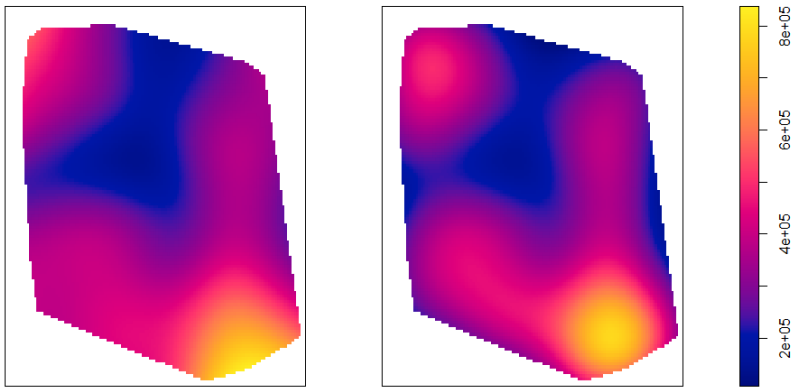


e) Nyiberekera

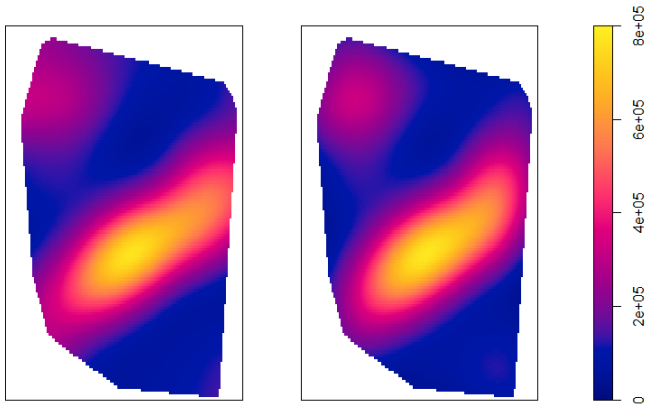
Figure 34: Intensity functions of Nyiberekera with the `gaussian`, `epanechnikov`, `quartic`, and `disc` kernels respectively without edge effects.

Intensity functions with different options for edge effect correction

In Figures 35 and 36 it can be seen how the intensities appear with and without Diggle's correction for edge effects.

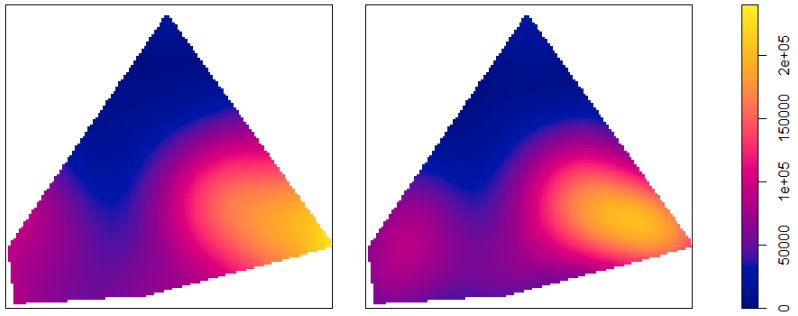


a) Buchanchari

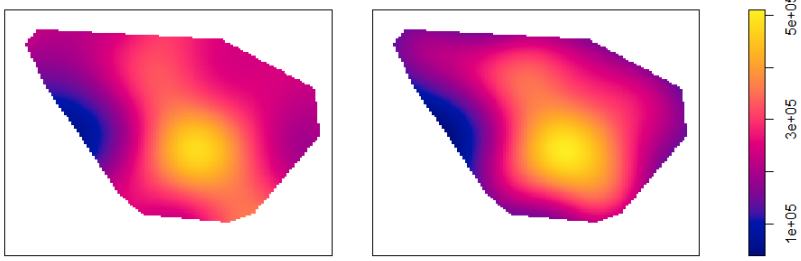


b) Kitembere

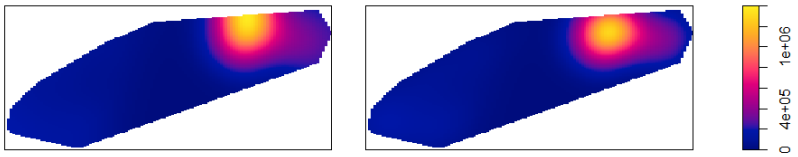
Figure 35: Intensity functions of the villages with `diggle = FALSE` and `diggle = TRUE` respectively with the `gaussian` kernel.



c) Magatini



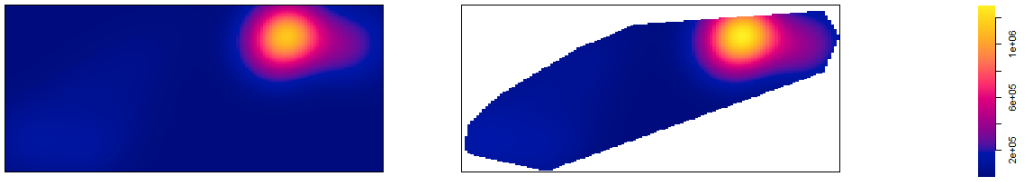
d) Monuna



e) Nyiberekera

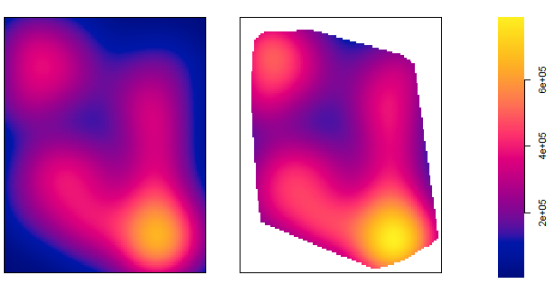
Figure 36: Intensity functions of the villages with `diggle = FALSE` and `diggle = TRUE` respectively with the `gaussian` kernel.

Changing the window will not change how the edge effects are corrected. What does change as an effect of changing the window is the appearance of the intensity functions, as seen in Figures 38 and 37 where the intensities have been plotted with equal range of values on the bar scale for each village. Notice how the intensities in the standard window differ from those in the `convexhull` window. This shows that the selection of a window is important and should be taken into account when plotting intensity functions.

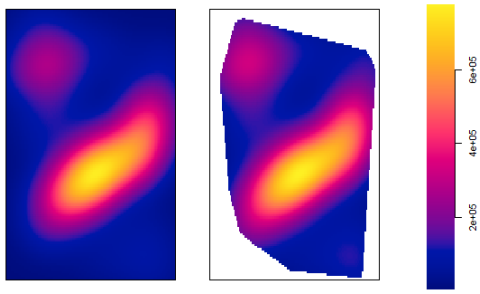


a) Nyiberekera

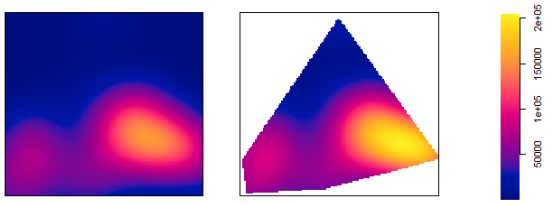
Figure 37: Comparing intensities between the standard window and the `convexhull` window using the `gaussian` kernel with edge effects for Nyiberekera.



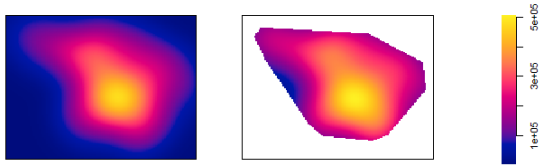
b) Buchanchari



c) Kitembere



d) Magatini



e) Monuna

Figure 38: Comparing intensities between the standard window and the `convexhull` window using the `gaussian` kernel with edge effects for each village.

4.4 Kernel density estimation in SAS software

SAS software can perform univariate and bivariate kernel density estimation using the `KDE` procedure. This procedure only uses the Gaussian kernel with a simple method to adjust the bandwidth accordingly. It can also give summary statistics, percentiles and levels of kernel density estimation. It does not, however, have the ability to change the window effect nor does it have the ability to make corrections for edge effects. What follows is an example of kernel density estimation on a randomised point pattern done in SAS software[5]. The code is given below and the output is given in Figures 39 to 42.

SAS Code:

```
data a;
n = 50;
do i = 1 to n;

    x = 50*ranuni(121);
    y = 50*ranuni(212);
    output;

end;
symbol1 value = dot;
proc gplot;
plot y*x;
run;
```

[5]The code and output for this subsection was generated using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2002-2012 by SAS Institute Inc. SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc., Cary, NC, USA.

```
ods graphics on;
proc kde data = a;
univar x y / plots = (histdensity);
bivar x y / plots = (contour surface);
run;
ods graphics off;
```

Selected output:

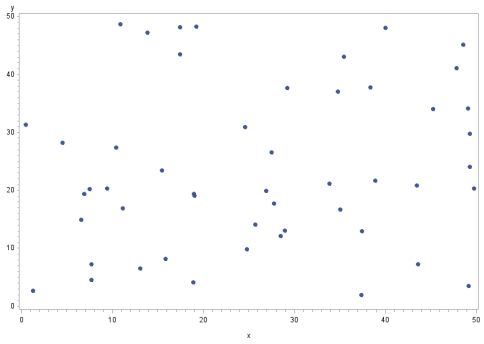


Figure 39: Plot of the `x` values against the `y` values.

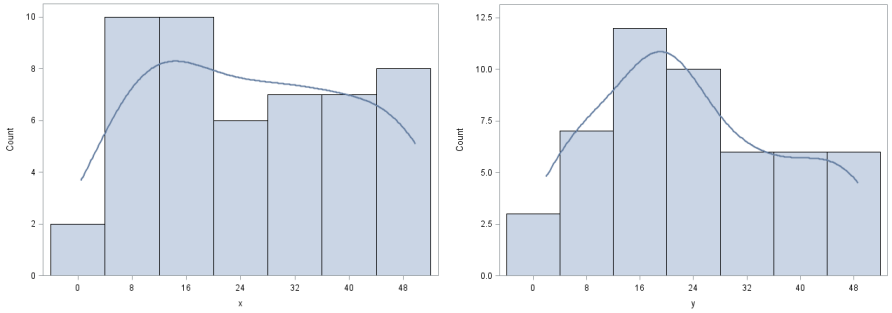


Figure 40: Histogram and kernel density for the `x` and `y` values respectively.

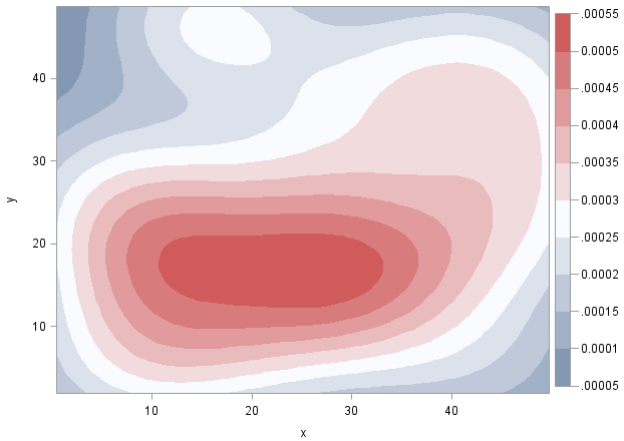


Figure 41: Kernel density estimation for `x` and `y`.

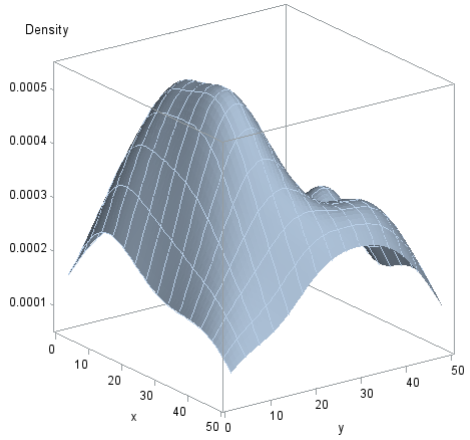


Figure 42: 3D perspective of the kernel density estimation for `x` and `y`.

5 Conclusion

In this report, we have looked at kernel density estimation in the one- and two-dimensional space and how the `density.ppp` function in the `spatstat` package in `R` uses it to create intensity functions of spatial point patterns. We also looked into the `density.ppp` function and studied its different parameters and arguments, including adjusting for the edge effects and the different types of bandwidth selection.

In the application we applied the `density.ppp` function on five different maps of villages which served as point patterns with the households being the points. We tested the function under two different window selections: the standard rectangular window and the `convexhull` window, with and without Diggle's correction for edge effects. We observed that the appearance of an intensity function was greatly influenced by the distribution of points in the window, and that points were not all well represented by the intensity function because of that.

Another result was that despite changing the effect of the window, edge effect correction still worked as it was supposed to work. The appearance of the intensity functions, however, did change as a result of changing the window (which should be expected due to the nature of edge effect correction). For this reason, window selection should be a significant step to consider before creating intensity functions. A recommendation would be to have more research done on the appropriate selection of a window.

We also looked at an example of kernel density estimation done in SAS software using the `KDE` procedure.

References

[1] Adrian Baddeley. Analysing spatial point patterns in `R`. Technical Report Version 4.1, CSIRO and University of Western Australia, 2010.

[2] Adrian Baddeley, Ege Rubak, and Rolf Turner. *Spatial Point Patterns: Methodology and Applications with R*. CRC Press, 2015.

[3] Adrian Baddeley and Rolf Turner. Introduction to `spatstat`. *Help manual for the* `R` *package* `spatstat`, 2003.

[4] Adrian Baddeley and Rolf Turner. `spatstat`: an `R` package for analyzing spatial point patterns. *Journal of Statistical Software*, 12(6):1–42, 2005.

[5] Mark Berman and Peter Diggle. Estimating weighted integrals of the second-order intensity of a spatial point process. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 81–92, 1989.

[6] Noel Cressie. *Statistics for Spatial Data*. Wiley, 1993.

[7] Peter Diggle. A kernel method for smoothing point process data. *Applied Statistics*, pages 138–147, 1985.

[8] Peter Diggle. *Handook of Spatial Statistics*, chapter 18, pages 299–316. CRC Press, 2010.

[9] Peter J Diggle. *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns*. CRC Press, 2014.

[10] R. A. Fisher. *The Design of Experiments*. Oliver And Boyd; Edinburgh; London, 1937.

[11] Alan E Gelfand, Peter Diggle, Peter Guttorp, and Montserrat Fuentes. *Handbook of Spatial Statistics*. CRC Press, 2010.

[12] Alan Julian Izenman. Recent developments in nonparametric density estimation. *Journal of the American Statistical Association*, 86(413):205–224, 1991.

[13] Clive Loader. *Local Regression and Likelihood*. Springer Science & Business Media, 2006.

[14] J. Mateu and F. Montes, editors. *Spatial Statistics Through Applications*. Advances in Ecological Sciences. WIT Press, 2002.

[15] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[16] Brian D Ripley. *Spatial Statistics*. Wiley, 1981.

[17] Brian D Ripley. *Statistical Inference for Spatial Processes*. Cambridge University Press, 1991.

[18] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.

[19] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley Series in Probability and Statistics. Wiley, 1992.

[20] Bernard W Silverman. *Density Estimation for Statistics and Data Analysis*. CRC Press, 1986.

[21] L Strand. A model for stand growth. In *IUFRO Third Conference Advisory Group of Forest Statisticians*, volume 72, page 3, 1972.

Alternative $\alpha - \mu$ fading models

Michaela Laidlaw 14012384

WST795 Research Report

Submitted in partial fulfilment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Prof. Andriëtte Bekker, Co-supervisor: Mr. Johan Ferreira

Department of Statistics, University of Pretoria

30 October 2017

Abstract

This study introduces the $\alpha - \mu$ fading model within the elliptical class. First the well-known $\alpha - \mu$ distribution will be studied. The latter distribution's characteristics will be revisited and its feasibility as a fading model in wireless communications systems will be investigated. Secondly the extension under the elliptical umbrella will be proposed and the contribution as a fading model will be demonstrated.

Keywords: average bit-error rate, elliptical class, fading model, generalized gamma distribution, outage probability, signal-to-noise ratio.

Software packages used: Mathematica, SAS.

Declaration

I, *Michaela Laidlaw,* declare that this essay, submitted in partial fulfilment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Michaela Laidlaw

_ _

Prof. Andriëtte Bekker

_ _

Mr. Johan Ferreira

_ _

Date

Acknowledgements

Thank you to Prof. Andriëtte Bekker and Mr. Johan Ferreira for their unwavering support and guidance throughout the year.

This work is based upon research supported by the National Research Foundation (NRF), South Africa (ref CPRR 160403161466 grant nr 105840). Opinions expressed and conclusions arrived at in this research, are those of the author and are not necessarily to be attributed to the NRF.

Lastly I would like to thank STATOMET for their financial support.

Contents

List of Figures

List of Tables

1 Introduction

In the field of communication systems, fading channels are characterized as statistical distributions used to describe the signal degradation from the transmitter to receiver of wireless signals.

To explain how fading of a signal works and how a fading model's performance is evaluated the fundamental theory of wireless systems needs to be understood. This includes how modelling is approached as well as the performance metrics that are used in order to compare the efficiency of systems, all of which is described in [17, 18]. In recent times, there have been improvements on many approaches to the derivation of the fading models and their performance metrics. These advances in the field of wireless communication have resulted in models that are more relevant and expressions for existing models that have been improved computationally, all of which is summarized by Paris [14]. One such new development is the $\alpha - \mu$ distribution introduced in 2002 by Yacoub [20] with the purpose of modelling fading in non-linear environments where surfaces that cause diffusion and scattering are spatially correlated [21]. The $\alpha - \mu$ distribution has been shown to be Stacy's distribution that has been reparameterized. Expressions to evaluate the performance of a system, together with joint statistics have been derived for the $\alpha - \mu$ distribution [21]. The Stacy distribution is a generalization of the gamma distribution and has three parameters as opposed to the traditional gamma's two. Stacy formally derived this distribution as well as some properties such as the distribution of the sum of independent random variables from the generalized gamma distribution [19], the earliest form as this distribution was seen in 1925 [1]. The $\alpha - \mu$ distribution is also very broad since it includes many distributions as special cases, the Nakagami-m being one example, and since Yacoub found the expressions in terms of the physical fading parameters, statistics of the special cases can be found directly from the $\alpha - \mu$ distribution's derived expressions [21].

The expression of the Laplace transform as derived by Yacoub is not of a convenient form computationally, but work has been done to find alternative closed form expressions. The first method makes use of Meijer's G-functions [9]. These expressions are expanded to find an expression for the bit-error rate (BER) which is a performance measure traditionally calculated from the Laplace transform. More recent developments yielded an alternative which may be computationally easier than the G-functions and is found by approximating the exponential functions in the Laplace transforms [16].

It has been remarked that a more general assumption than the underlying normal may not be far from reality [12], and so this study will focus on the $\alpha - \mu$ distribution and its characteristics and then propose the distribution within the elliptical class [4]. The elliptical class is used to allow us to consider the $\alpha - \mu$ distribution with a underlying t distribution, this allows us to see how a distribution with heavier tails will perform when modelling fading. The $\alpha - \mu$ distribution with underlying t distribution will be found by substituting the relevant weighting function, as represented by Chu [4], into the expressions for the $\alpha - \mu$ type.

The study is compiled as follows, first some background theory of some concepts which will be used and investigated throughout the remainder of this report are given. Section 3 considers the construction of the $\alpha - \mu$ distribution from a physical point of view [21]. The probability density function (PDF), cumulative distribution function (CDF) and moment generating function (MGF) [9, 16] of the $\alpha - \mu$ model will receive attention. In section 4 the $\alpha - \mu$ distribution will be proposed within the elliptical class and alternative $\alpha - \mu$ models will be introduced, resulting in the $\alpha - \mu$ type. These alternative $\alpha - \mu$ models are of interest since they may yield better results than existing models. Section 5 will consider the performance of the $\alpha - \mu$ type comparing the underlying normal- and t case.

2 Background

Some necessary elements for this study will be described below.

2.1 The fading model

When a signal is transmitted there is fluctuation in the power due to objects between the transmitter and the receiver. These fluctuations occur over a short period of time and are known as short-term fading, or simply, fading. The fading is due to the objects causing the signal to scatter, reflect and diffract and this results in there being more than one way for the signal to go from the transmitter to the receiver [17], this is shown in Figure 1. The fading envelope, also known as the amplitude or magnitude, is the smooth curve outlining the extremes of an oscillating signal and will be denoted by the random variable R. When deciding on a model to describe the fading envelope's behaviour, the environment in which the propagation occurs must be taken into consideration. The fading distribution will always form part of the exponential class but the complexity of the distribution may vary [18]. In the event that the signal encounters multiple objects between the transmitter have long-term fading occurs, also known as shadowing which might be described by the lognormal distribution [5]. In this study, attention will be given to short-term fading.

2.2 Signal-to-noise ratio (SNR)

The instantaneous SNR, denoted by γ, is a performance measure taken at the receiver and so is related to the receiver's ability to detect data. The SNR is a good indication of the system's overall accuracy and of the performance metrics considered, will be the easiest to compute in most cases. It is expressed as the ratio between the power of the signal and that of the noise at the receiver's output [18]. Let R be the random variable representing the envelope of the fading channel with $\hat{r} = \sqrt[\alpha]{\mathrm{E}\left(R^{\alpha}\right)}$ thus $\hat{r}^{\alpha} = \mathrm{E}\left(R^{\alpha}\right)$. Then the instantaneous SNR expressed in terms of the channel envelope is $\gamma = \hat{\gamma}\left(\frac{R}{\hat{r}}\right)^2 = \hat{\gamma}R^2\frac{1}{\mathrm{E}(R^2)}$ where

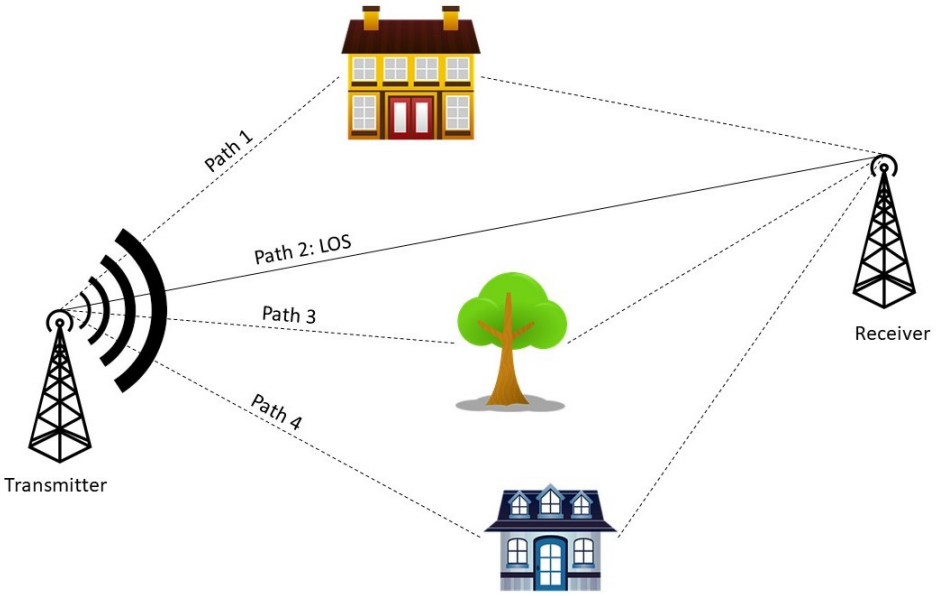


Figure 1: Illustration of how a signal can have many paths to travel from the transmitter to the receiver. The Line-Of-Sight (LOS) path is the only one where there is no scattering, reflecting or diffraction of the signal as it makes its way to the receiver.

$\hat{\gamma} = \mathrm{E}\left(\hat{r}^2\right) \frac{E_b}{N_o}$, E_b is the energy per bit and N_0 the noise spectral density [9].

2.3 Outage probability (OP)

The outage probability is a performance measure for wireless communication channels and is a criterion with which comparisons between models can be made by considering their capability of maintaining a specific SNR [17]. The outage probability can be defined as the probability that the received SNR drops below a predetermined threshold [18]. In statistical terms the outage probability is the CDF of the SNR, thus OP $= \int_0^{\gamma_{OP}} f_\gamma(\gamma)\, d\gamma$ where γ_{OP} is the threshold and $f_\gamma(\gamma)$ is the PDF of the SNR. It is worth noting that a MGF based approach does exits, [18], but will not be considered in this study.

2.4 Average bit-error rate (ABER)

Of the performance metrics considered in this study this is the most challenging but gives the most insight into the system's attributes. When data travels through wireless systems the digitized form at the transmitter is transformed into 0's and 1's known as bits. These bits form a sequence when combined, when a sequence consists of a single bit one will detect binary signals [17]. Noise enters the system between the transmitter and receiver. The noise in the system affects the system by corrupting the signal resulting in errors between what was received and what was transmitted. ABER quantifies this error caused by the noise. When binary shift-phase-keying (BPSK), as coherent detection method, is used the ABER can be written as a Gaussian Q-function:

$$P_b(E) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right),$$

where E_b is the energy per bit and N_0 the noise spectral density [18]. The ABER can also be found by evaluating the finite integral [6, 18]:

$$P_b(E) = \frac{a_m}{\pi}\int_0^{\frac{\pi}{2}} L_\gamma\left(\frac{b_m^2}{2\sin^2(\theta)}\right)d\theta, \tag{1}$$

when coherent detection of BPSK is used. In this case $a_m = 1$ and $b_m^2 = 2\frac{E_b}{N_0}$.

2.5 Elliptical class

A random variable X is said to be a member of the elliptical class, denoted as $X \sim E(\mu, \sigma^2, h)$ $h(\cdot)$ being a generator function, with mean $\mu > 0$ and variance $\sigma^2 > 0$ only when its PDF is a function of a quadratic form, hence

$$f_X(x) = h(-\frac{1}{2\sigma^2}(x-\mu)^2). \tag{2}$$

It is worth noting that the PDF of a random variable with an elliptical distribution can be written as the integral of a set of normal densities. Thus (2) can be expanded as:

$$f_X(x) = \int_0^\infty W(t) f_{N(\mu, t^{-1}\sigma^2)}(x)dt$$

where $W(t)$ is known as the weighting function, dependent only upon $t \in (0, \infty)$, and $f_{N(\mu,t^{-1}\sigma^2)}(x)$ is the PDF of a normally distributed random variable with mean μ and variance $t^{-1}\sigma^2$ [4]. The different choices of the weighting function results in a large variety of functions, making the elliptical class very flexible . This flexibility makes it possible to implement distributions with different characteristics than the normal distribution, such as heavier or lighter tails [2, 13].

3 $\alpha - \mu$ model

The commonly used fading models such as the Rayleigh distribution assume that the received signal can be found by the addition of vector sums representing scattering, diffraction and reflection from

different objects. These models have a drawback in that they can only model the fading accurately when the scattering is homogeneous and require a large number of scatters to apply the central limit theorem (CLT). The $\alpha-\mu$ model is a unified model (also known as a general fading model), these models work under the assumption that the received signal is found through modelling the objects as clusters, rather than individually, Figure 2 illustrates this clustering. Unified models are statistically complex in their explanation of fading while also being diverse. Signal in wireless channels is more likely to be heterogeneous in nature with the number of obstacles between the transmitter and receiver finite. This heterogeneity leads to the signal conducting itself in a non-linear manner [17]. The number of scatters required to implement the CLT will likely be larger than the number of clusters, which adds to unified model's relevance.

The $\alpha-\mu$ distribution was introduced by Yacoub [21] with the purpose of modelling fading in non-linear environments where there exists a spatial correlation between the surfaces that are causing the diffusion and scattering. In the model the non-linearity of the propagation medium is represented by α while μ represents the number of multipath clusters [8].

First to be considered is the PDF of R's derivation after which the moments of the fading signal's envelope. These results are then used to find the instantaneous SNR, γ, which is a performance metric but will also be used to find both the outage probability and the ABER. The $\alpha-\mu$ distribution is a very flexible distribution and its relation with other distributions will also be considered.

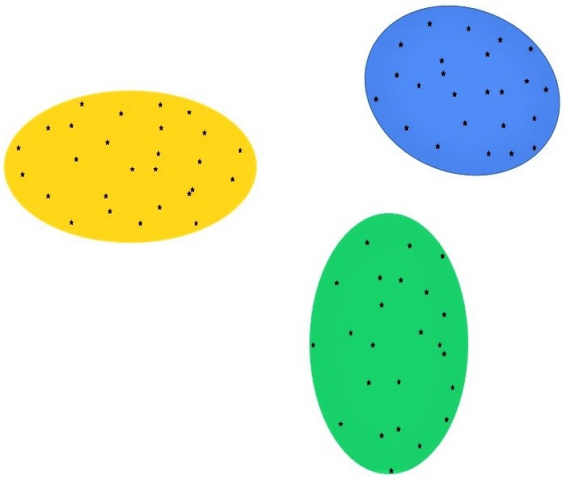


Figure 2: Generalized fading considers clusters of objects as displayed.

3.1 Construction of the model

The derivation will be done as approached by Yacoub [21] by describing the envelope as

$$R^{\alpha}=\sum_{i=1}^{\mu}\left(X_i^2+Y_i^2\right), \tag{3}$$

$\alpha > 0$ and μ a positive integer. This description of the envelope is under the assumption that there is a sufficient number of scatters, or objects, within each cluster. When the number of scatters within the cluster is sufficient it may be assumed that X and Y are independent normal random variables [17].

Theorem 1. *Consider two mutually independent normal random variables X_i and Y_i with $E(X_i) = E(Y_i) = 0$ and $E(X_i^2) = E(Y_i^2) = \frac{\hat{r}^\alpha}{2\mu}$ where $\hat{r} = \sqrt[\alpha]{E(R^\alpha)}$ and for $i = 1, ..., \mu$. $\hat{r}$ is the α-root mean value of the envelope random variable R. The PDF of R, the envelope of the fading signal, is given by*

$$f_R(r) = \frac{\alpha r^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \exp\left[-\mu\left(\frac{r}{\hat{r}}\right)^\alpha\right], \qquad r > 0 \tag{4}$$

where $\Gamma(\cdot)$ is the gamma function (see Appendix R3).

Proof. Suppose $Y_i \sim N\left(0, \frac{\hat{r}^\alpha}{2\mu}\right)$ and $X_i \sim N\left(0, \frac{\hat{r}^\alpha}{2\mu}\right)$ where $i = 1, ..., \mu$. Then $\left(Y_i\sqrt{\frac{2\mu}{\hat{r}^\alpha}}\right)^2 \sim \chi^2(1)$ and similarly $\left(X_i\sqrt{\frac{2\mu}{\hat{r}^\alpha}}\right)^2 \sim \chi^2(1)$ (see Appendix R1). From (3) it follows that

$$R^\alpha\frac{2\mu}{\hat{r}^\alpha} = \sum_{i=1}^{\mu}\left(X_i\sqrt{\frac{2\mu}{\hat{r}^\alpha}}\right)^2 + \sum_{i=1}^{\mu}\left(Y_i\sqrt{\frac{2\mu}{\hat{r}^\alpha}}\right)^2.$$

Resulting in

$$R^\alpha\frac{2\mu}{\hat{r}^\alpha} \sim \chi^2(2\mu) \tag{5}$$

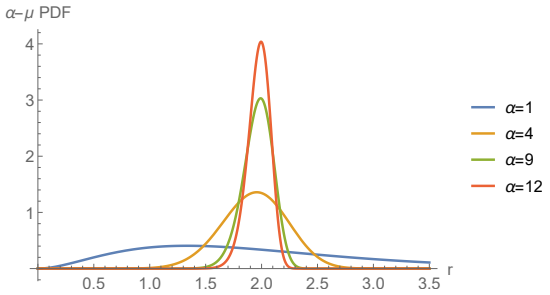
since $\sum_{i=1}^{\mu}\left(Y_i\sqrt{\frac{2\mu}{\hat{r}^\alpha}}\right)^2 \sim \chi^2(\mu)$ and $\sum_{i=1}^{\mu}\left(X_i\sqrt{\frac{2\mu}{\hat{r}^\alpha}}\right)^2 \sim \chi^2(\mu)$ (see Appendix R2). Let $A = R^\alpha\frac{2\mu}{\hat{r}^\alpha}$, then $R = \sqrt[\alpha]{\frac{\hat{r}^\alpha}{2\mu}A}$ and $\frac{dA}{dR} = \alpha R^{\alpha-1}\frac{2\mu}{\hat{r}^\alpha}$. The PDF of A is is given by:

$$f(A) = \frac{1}{2^\mu\Gamma(\mu)}a^{\mu-1}\exp\left(-\frac{a}{2}\right), \qquad a > 0, \tag{6}$$

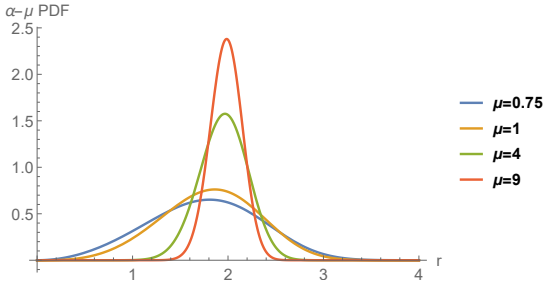
this is a Gamma$(\mu, 2)$ distribution (see Appendix R4). Using (6) the PDF of R can be determined as

$$\begin{aligned} f_R(r) &= f_A(r)\frac{dA}{dR} \\ &= \frac{\left(r^\alpha\frac{2\mu}{\hat{r}^\alpha}\right)^{\mu-1}\exp\left(-\frac{1}{2}(r^\alpha\frac{2\mu}{\hat{r}^\alpha})\right)}{2^\mu\Gamma(\mu)}\frac{2\mu\alpha\left(\sqrt[\alpha]{r^\alpha}\right)^{\alpha-1}}{\hat{r}^\alpha} \\ &= \frac{\alpha r^{\alpha\mu-\alpha}2^{\mu-1}\mu^{\mu-1}\hat{r}^{\alpha(1-\mu)}\exp\left(-\mu\frac{r^\alpha}{\hat{r}^\alpha}\right)2\mu r^{\alpha-1}}{2^\mu\Gamma(\mu)\hat{r}^\alpha} \\ &= \frac{\alpha r^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}}\exp\left[-\mu\left(\frac{r}{\hat{r}}\right)^\alpha\right], \qquad r > 0 \end{aligned}$$

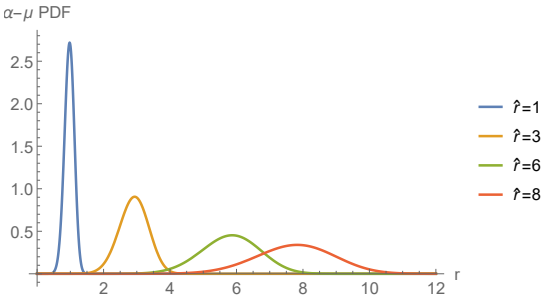
which leaves the final result. □



(a) (4) with $\mu = 3$ and $\hat{r} = 2$ and α varying



(b) (4) with $\alpha = 4$ and $\hat{r} = 2$ and μ varying



(c) (4) with $\mu = 3$ and $\alpha = 4$ and $\hat{r}$ varying

Figure 3: PDFs of the $\alpha - \mu$ distribution.

Figures 3a - 3c illustrate the effect of a change in parameter values on (4).

Remark 2. Consider the following reparameterization of (4):

$$\hat{r} = a\sqrt[p]{\frac{d}{p}}$$

$$\mu = \frac{d}{p}$$

$$\alpha = p$$

The resulting PDF is:

$$f_R(r) = \frac{p}{a^d} r^{d-1} \exp\left[-\left(\frac{r}{a}\right)^p\right] \frac{1}{\Gamma\left(\frac{d}{p}\right)} \qquad a, d, p, r > 0. \tag{7}$$

Equation (7) is the generalized gamma as derived by Stacy [19] and thus the $\alpha - \mu$ distribution is a reparameterization of the Stacy distribution.

Cumulative distribution function (CDF)

The CDF of R using (4) can be determined as:

$$
\begin{aligned}
F_R(r) &= \int_0^r f_R(s)\,ds \\
&= \int_0^r \frac{\alpha s^{\alpha\mu-1}}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}}\mu^{\mu}\exp\left(-\mu\frac{s^{\alpha}}{\hat{r}^{\alpha}}\right)ds \\
&= \frac{\alpha\mu^{\mu}}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}}\int_0^r s^{\alpha\mu-1}\exp\left(-\mu\frac{s^{\alpha}}{\hat{r}^{\alpha}}\right)ds \\
&= \frac{\alpha\mu^{\mu}}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}}\frac{\gamma\left(\mu,\mu\left(\frac{u}{\hat{r}}\right)^{\alpha}\right)}{\alpha\left(\frac{\mu}{\hat{r}^{\alpha}}\right)\mu}, \qquad u>0 \\
&= \frac{\gamma\left(\mu,\mu\left(\frac{u}{\hat{r}}\right)^{\alpha}\right)}{\Gamma(\mu)}, \qquad u>0
\end{aligned} \tag{8}
$$

where $\gamma(\cdot,\cdot)$ is the lower incomplete gamma function (see Appendix R5).

Figures 4a - 4c illustrate the effect a change of parameter values has on the CDF (8).

3.2 Moments

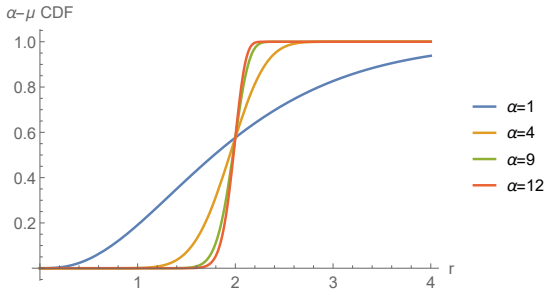
The MGF and the v^{th} moment of the $\alpha-\mu$ distribution will be considered, while the expected value and variance of the fading signal's envelope R will be revisited [21].

3.2.1 Moment generating function

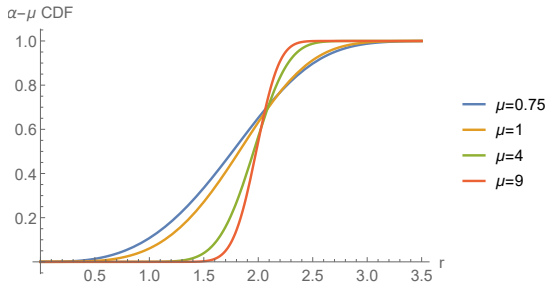
The MGF of the $\alpha-\mu$ distribution is derived as follows:

$$
\begin{aligned}
M_R(c) &= \mathrm{E}\left[\exp(cr)\right] \\
&= \int_0^{\infty} f_R(r)\exp(cr)\,dr \\
&= \int_0^{\infty} \frac{\alpha r^{\alpha\mu-1}}{\Gamma(\mu)\left(\frac{\hat{r}^{\alpha}}{\mu}\right)^{\mu}}\exp\left(-\frac{r^{\alpha}\mu}{\hat{r}^{\alpha}}\right)\exp(cr)\,dr \\
&= \mu^{\mu}\alpha\frac{\hat{r}^{-\alpha\mu}}{\Gamma(\mu)}\int_0^{\infty}\exp\left(-\frac{r^{\alpha}\mu}{\hat{r}^{\alpha}}\right)r^{\alpha\mu-1}\sum_{k=0}^{\infty}\frac{(cr)^{k}}{k!}dr \\
&= \mu^{\mu}\alpha\frac{\hat{r}^{-\alpha\mu}}{\Gamma(\mu)}\sum_{k=0}^{\infty}\frac{c^{k}}{k!}\int_0^{\infty}\exp\left(-\frac{r^{\alpha}\mu}{\hat{r}^{\alpha}}\right)r^{\alpha\mu+k-1}dr \\
&= \mu^{\mu}\alpha\frac{\hat{r}^{-\alpha\mu}}{\Gamma(\mu)}\sum_{k=0}^{\infty}\frac{c^{k}}{k!}\frac{\Gamma\left(\mu+\frac{k}{\alpha}\right)}{\alpha\left(\hat{r}^{-\alpha}\mu\right)^{\mu+\frac{k}{\alpha}}} \\
&= \frac{1}{\Gamma(\mu)}\sum_{k=0}^{\infty}\frac{c^{k}\Gamma\left(\mu+\frac{k}{\alpha}\right)}{k!\left(\hat{r}^{-\alpha}\mu\right)^{\frac{k}{\alpha}}},
\end{aligned} \tag{9}
$$

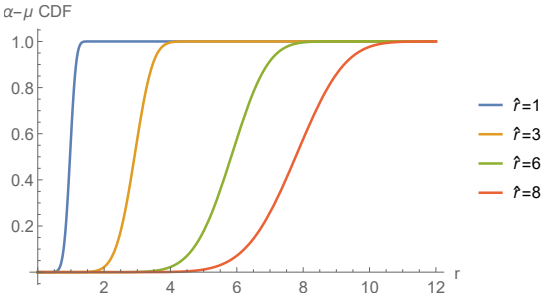
where $\alpha>0$, $\alpha\mu+k-1>0$ and $\hat{r}^{-\alpha}\mu>0$ (see Appendix R6).



(a) (8) with $\mu = 3$, $\hat{r} = 2$ and α varying.



(b) (8) with $\alpha = 4$, $\hat{r} = 2$ and μ varying.



(c) (8) with $\mu = 3$, $\alpha = 4$ and $\hat{r}$ varying.

Figure 4: CDFs of $\alpha - \mu$ distribution.

3.2.2 j^{th} moment

The j^{th} moment of the $\alpha - \mu$ distribution, which is given by $\mathrm{E}\left(R^j\right)$, will now be revisited [20, 21].

$$
\begin{aligned}
m_j &= \mathrm{E}\left(R^j\right) \\
&= \int_0^\infty r^j f_R\left(r\right) dr \\
&= \int_0^\infty r^j \mu^\mu \alpha \hat{r}^{-\alpha} \frac{j^\mu}{\Gamma\left(\mu\right)} \exp\left(-\frac{r^\alpha \mu}{\hat{r}^\alpha}\right) r^{\alpha\mu-1} dr \\
&= \mu^\mu \alpha \frac{\hat{r}^{-\alpha\mu}}{\Gamma\left(\mu\right)} \int_0^\infty r^{j+\alpha\mu-1} \exp\left(-\frac{r^\alpha \mu}{\hat{r}^\alpha}\right) dr \\
&= \frac{\mu^\mu \alpha}{\Gamma\left(\mu\right)\hat{r}^{\alpha\mu}} \frac{\Gamma\left(\frac{\alpha\mu+j}{\alpha}\right)}{\alpha\left(\mu\hat{r}^{-\alpha}\right)^{\frac{\alpha\mu+j}{\alpha}}} \\
&= \frac{\mu^\mu}{\Gamma\left(\mu\right)\hat{r}^{\alpha\mu}} \frac{\Gamma\left(\mu+\frac{j}{\alpha}\right)\hat{r}^{\alpha\mu}\hat{r}^j}{\mu^\mu \mu^{\frac{j}{\alpha}}} \\
&= \frac{\hat{r}^j \Gamma\left(\mu+\frac{j}{\alpha}\right)}{\mu^{\frac{j}{\alpha}}\Gamma\left(\mu\right)},
\end{aligned}
\tag{10}
$$

where $\alpha > 0$, $\hat{r}^\alpha > 0$ and $\alpha\mu + j > 1$ (see Appendix R6).

3.2.3 First and second moment

A general form for the first two moments of the envelope R^α can not be obtained easily from the MGF or the v^{th} moment of the $\alpha-\mu$ distribution as derived above. This is due to both (9) and (10) not being in a form that is easily computable. However the distribution of $R^\alpha \frac{2\mu}{\hat{r}^\alpha}$ is known and is given by (5) hence

$$\begin{aligned} \mathrm{E}\left(R^\alpha \frac{2\mu}{\hat{r}^\alpha}\right) &= 2\mu \\ \mathrm{E}\left(R^\alpha\right) &= 2\mu \frac{\hat{r}^\alpha}{2\mu} \\ \mathrm{E}\left(R^\alpha\right) &= \hat{r}^\alpha. \end{aligned} \tag{11}$$

Solving for $\hat{r}$ in equation (11), $\hat{r} = \sqrt[\alpha]{\mathrm{E}\left(R^\alpha\right)}$.

$$\mathrm{E}\left(\left(R^\alpha\right)^2\right) = \mathrm{var}\left(R^\alpha\right) + \left(\mathrm{E}\left(R^\alpha\right)\right)^2.$$

For the second moment start by considering the variance of (5):

$$\begin{aligned} \mathrm{var}\left(R^\alpha \frac{2\mu}{\hat{r}^\alpha}\right) &= 4\mu \\ \mathrm{var}\left(R^\alpha\right) &= 4\mu \frac{\hat{r}^{2\alpha}}{4\mu^2} \\ &= \frac{\hat{r}^{2\alpha}}{\mu}, \end{aligned}$$

subsequently

$$\begin{aligned} \mathrm{E}\left(\left(R^\alpha\right)^2\right) &= \frac{\hat{r}^{2\alpha}}{\mu} + \hat{r}^{2\alpha} \\ &= \frac{\hat{r}^{2\alpha}\left(\mu+1\right)}{\mu}. \end{aligned}$$

These are the same results obtained in [21].

3.3 SNR

3.3.1 PDF and Outage probability

To find the instantaneous SNR, indicated by γ, the PDF of γ needs to be found, this can be done by performing a transformation on the $\alpha-\mu$ density function (4), as was done in [9].

Probability density function (PDF)

Suppose γ is the instantaneous SNR (see Section 2.2) and that $\gamma = \hat{\gamma}\left(\frac{R}{\hat{r}}\right)^2$ then $r = \hat{r}\sqrt{\frac{\gamma}{\hat{\gamma}}}$ and $\frac{dr}{d\gamma} = \frac{\hat{r}}{2}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}\frac{1}{\hat{\gamma}}$ thus the PDF of γ can be found as

$$\begin{aligned} f_\gamma(\gamma) &= f_\gamma(r)\frac{dr}{d\gamma} \\ &= \frac{\alpha r^{\alpha\mu-1}}{\Gamma(\mu)\left(\frac{\hat{r}^\alpha}{\mu}\right)^\mu}\exp\left(-\frac{r^\alpha\mu}{\hat{r}^\alpha}\right)\frac{\hat{r}}{2}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}\frac{1}{\hat{\gamma}} \\ &= \frac{\alpha\mu^\mu\left(\hat{r}\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}}{\Gamma(\mu)}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha\mu}\exp\left[-\mu\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^\alpha\right]\frac{\hat{r}}{2}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}\frac{1}{\hat{\gamma}} \\ &= \frac{\alpha\mu^\mu\gamma^{\frac{\alpha\mu}{2}-1}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\exp\left[-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right] \end{aligned} \tag{12}$$

where $\gamma > 0$.

Outage Probability

Let γ_{OP} be the threshold being considered then using (12):

$$\begin{aligned} F_\gamma(\gamma_{OP}) &= \int_0^{\gamma_{OP}} f_\gamma(\gamma)\,d\gamma \\ &= \int_0^{\gamma_{OP}} \frac{\alpha\mu^\mu\gamma^{\frac{\alpha\mu}{2}-1}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\exp\left[-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right]d\gamma \\ &= \frac{\alpha\mu^\mu}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^{\gamma_{OP}}\gamma^{\frac{\alpha\mu}{2}-1}\exp\left[-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right]d\gamma \\ &= \frac{\alpha\mu^\mu}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\frac{2\gamma\left(\frac{\alpha\mu}{2}\frac{2}{\alpha},\mu\hat{\gamma}^{-\frac{\alpha}{2}}\gamma_{OP}^{\frac{\alpha}{2}}\right)}{\alpha\left(\mu\hat{\gamma}^{-\frac{\alpha}{2}}\right)^\mu} \\ &= \frac{\gamma\left(\mu,\mu\left(\frac{\gamma_{OP}}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right)}{\Gamma(\mu)}, \end{aligned} \tag{13}$$

where $\gamma_{OP} > 0$ (see Appendix R6).

3.3.2 Laplace transform

The Laplace transform forms part of the calculation of the ABER, the Laplace transform of γ is given by:

$$
\begin{aligned}
L(c) &= \mathrm{E}\left[\exp\left(-c\gamma\right)\right] \\
&= \int_0^\infty f_\gamma(\gamma)\exp\left(-c\gamma\right)d\gamma \\
&= \int_0^\infty \frac{\alpha\mu^\mu\gamma^{\frac{\alpha\mu}{2}-1}}{2\Gamma\left(\mu\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\exp\left[-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right]\exp\left(-c\gamma\right)d\gamma \\
&= \frac{\alpha\mu^\mu}{2\Gamma\left(\mu\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty \gamma^{\frac{\alpha\mu}{2}-1}\exp\left[-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right]\exp\left(-c\gamma\right)d\gamma \qquad (14) \\
&= \frac{\alpha\mu^\mu}{2\Gamma\left(\mu\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty \gamma^{\frac{\alpha\mu}{2}-1}\exp\left[-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right]\sum_{k=0}^{\infty}\frac{\left(-c\gamma\right)^k}{k!}d\gamma \\
&= \frac{\alpha\mu^\mu}{2\Gamma\left(\mu\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\sum_{k=0}^{\infty}\frac{\left(-c\right)^k}{k!}\int_0^\infty \gamma^{\frac{\alpha\mu}{2}+k-1}\exp\left[-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right]d\gamma \\
&= \frac{\alpha\mu^\mu}{2\Gamma\left(\mu\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\sum_{k=0}^{\infty}\frac{\left(-c\right)^k}{k!}\frac{\Gamma\left(\mu+\frac{2k}{\alpha}\right)}{\frac{\alpha}{2}\left(\mu\hat{\gamma}^{-\frac{\alpha}{2}}\right)^{\mu+\frac{2k}{\alpha}}} \\
&= \frac{1}{\Gamma\left(\mu\right)}\sum_{k=0}^{\infty}\frac{\left(-c\right)^k}{k!}\frac{\Gamma\left(\mu+\frac{2k}{\alpha}\right)}{\mu^{\frac{2k}{\alpha}}}\hat{\gamma}^k. \qquad (15)
\end{aligned}
$$

Two approaches will be considered for finding a more convenient form of (15). The first approach will make use of Meijer's G-function [9]. The second method to be considered was introduced in 2015, and the desired expression for the Laplace transform is found through the approximation of the exponential function, $-\exp\left(-x^r\right)$, removing the need for G-functions [16]. The lack of G-functions in the expression leads to a result that is can be manipulated easily and efficiently. Both approaches will result in alternative closed form expressions of the SNR's Laplace transform.

Approach 1

Using (12) an alternative expression for the Laplace transform is derived below.

$$
\begin{aligned}
L(c) &= \mathrm{E}\left[\exp\left(-c\gamma\right)\right] \\
&= \int_0^\infty f_\gamma\left(\gamma\right)\exp\left(-c\gamma\right)d\gamma \\
&= \int_0^\infty \frac{\alpha\mu^\mu\gamma^{\frac{\alpha\mu}{2}-1}}{2\Gamma\left(\mu\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\exp\left(-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right)\exp\left(-c\gamma\right)d\gamma \\
&= \frac{\alpha\mu^\mu}{2\Gamma\left(\mu\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty \gamma^{\frac{\alpha\mu}{2}-1}\exp\left(-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right)\exp\left(-c\gamma\right)d\gamma \\
&= \frac{\alpha\mu^\mu}{2\Gamma\left(\mu\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\times
\end{aligned}
$$

$$\int_0^\infty \gamma^{\frac{\alpha\mu}{2}-1}\left(\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right)^{-1}\left[\left(\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right)^{1}\exp\left(-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right)\right](c\gamma)^{-1}\left[(c\gamma)^{1}\exp\left(-c\gamma\right)\right]d\gamma$$

$$=\frac{\alpha\mu^{\mu}}{2\Gamma\left(\mu\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\times$$

$$\int_0^\infty \gamma^{\frac{\alpha\mu}{2}-1}\left(\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right)^{-1}\left[\left(\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right)^{1}G_{0,1}^{1,0}\left(\frac{\mu}{\hat{\gamma}^{\frac{\alpha}{2}}}\gamma^{\frac{\alpha}{2}}\middle|\begin{matrix}-\\0\end{matrix}\right)\right](c\gamma)^{-1}\left[(c\gamma)^{1}G_{0,1}^{1,0}\left(c\gamma\middle|\begin{matrix}-\\0\end{matrix}\right)\right]d\gamma$$

$$=\frac{\alpha\mu^{\mu}}{2\Gamma\left(\mu\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty \gamma^{\frac{\alpha\mu}{2}-1}G_{0,1}^{1,0}\left(\frac{\mu}{\hat{\gamma}^{\frac{\alpha}{2}}}\gamma^{\frac{\alpha}{2}}\middle|\begin{matrix}-\\0\end{matrix}\right)G_{0,1}^{1,0}\left(c\gamma\middle|\begin{matrix}-\\0\end{matrix}\right)d\gamma \tag{16}$$

$$=\frac{\alpha\mu^{\mu}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\frac{k^{\frac{1}{2}}l^{\frac{\alpha\mu-1}{2}}}{(2\pi)^{\frac{l+k-2}{2}}c^{\frac{\alpha\mu}{2}}}G_{l,k}^{k,l}\left(\left(\frac{\mu}{\hat{\gamma}^{\frac{\alpha}{2}}}\right)^{k}\frac{l^{l}}{c^{l}k^{k}}\middle|\begin{matrix}I\left(l,1-\frac{\alpha\mu}{2}\right)\\I\left(k,0\right)\end{matrix}\right), \tag{17}$$

with $I(n,\epsilon)=\frac{\epsilon}{n},\frac{\epsilon+1}{n},...,\frac{\epsilon+n-1}{n}$ and $\frac{\alpha}{2}$ was defined such that $\frac{\alpha}{2}=\frac{l}{k}$ and to include values for which α is not an integer the greatest common divisor for l and k is one, the G-function for the integral considered can be found in [15, p346] (see Appendix R8).

Approach 2

Using (12) as a departure point the following approximation of the exponential function will be applied [16]:

$$\exp\left(-z^{\frac{1}{\tilde{\alpha}}}\right)\approx\sum_{i=1}^{4}a_i\exp\left(-b_iz\right)$$

is extended to the case

$$\exp\left(-cz^{\frac{1}{\tilde{\alpha}}}\right)\approx\sum_{i=1}^{4}a_i\exp\left(-b_icz\right)$$

with a_i and b_i fitting parameters and $\tilde{\alpha}=\frac{\alpha}{2}$. Rewritten in terms of α, where α still represents the non-linearity of the propagation medium

$$\exp\left(-cz^{\frac{2}{\alpha}}\right)\approx\sum_{i=1}^{4}a_i\exp\left(-b_icz\right). \tag{18}$$

This approximation's value will change only when considering different cases of non-linearity in propagation medium, α [16]. Consider the Laplace transform:

$$L\left(c\right)=\mathrm{E}\left(\exp\left(-c\gamma\right)\right)$$

$$
\begin{aligned}
&= \int_0^\infty f_\gamma(\gamma) \exp(-c\gamma)\, d\gamma \\
&= \int_0^\infty \frac{\alpha\mu^\mu \gamma^{\frac{\alpha\mu}{2}-1}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \exp\left[-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right] \exp(-c\gamma)\, d\gamma \\
&= \frac{\alpha\mu^\mu}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \int_0^\infty \gamma^{\frac{\alpha\mu}{2}-1} \exp\left[-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right] \exp(-c\gamma)\, d\gamma.
\end{aligned}
$$

Consider the transformation $\gamma^{\frac{\alpha}{2}} = z$ thus $\gamma = z^{\frac{2}{\alpha}}$ and with $\frac{d\gamma}{dz} = \frac{2}{\alpha} z^{\frac{2}{\alpha}-1}$.

$$
\begin{aligned}
L(c) &= \frac{\alpha\mu^\mu}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \int_0^\infty \left(z^{\frac{2}{\alpha}}\right)^{\frac{\alpha\mu}{2}-1} \exp\left[-\mu z \frac{1}{\hat{\gamma}^{\frac{\alpha}{2}}}\right] \exp\left(-cz^{\frac{2}{\alpha}}\right) \frac{2}{\alpha} z^{\frac{2}{\alpha}-1} dz \\
&= \frac{\mu^\mu}{\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \int_0^\infty z^{\mu-\frac{2}{\alpha}} \exp\left(-\mu z \frac{1}{\hat{\gamma}^{\frac{\alpha}{2}}}\right) \exp\left(-cz^{\frac{2}{\alpha}}\right) z^{\frac{2}{\alpha}-1} dz \\
&= \frac{\mu^\mu}{\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \int_0^\infty z^{\mu-1} \exp\left(-\mu z \frac{1}{\hat{\gamma}^{\frac{\alpha}{2}}}\right) \exp\left(-cz^{\frac{2}{\alpha}}\right) dz.
\end{aligned}
$$

Substituting (18) in

$$
\begin{aligned}
L(c) &\approx \frac{\mu^\mu}{\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \int_0^\infty z^{\mu-1} \exp\left(-\mu z \frac{1}{\hat{\gamma}^{\frac{\alpha}{2}}}\right) \sum_{i=1}^4 a_i \exp(-b_i c z)\, dz \\
&= \frac{\mu^\mu}{\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \sum_{i=1}^4 a_i \int_0^\infty z^{\mu-1} \exp\left(-z\left(\frac{\mu}{\hat{\gamma}^{\frac{\alpha}{2}}} + b_i c\right)\right) dz \\
&= \frac{\mu^\mu}{\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \sum_{i=1}^4 a_i \frac{\Gamma(\mu)}{\left(\mu\hat{\gamma}^{-\frac{\alpha}{2}} + b_i c\right)^\mu} \\
&= \frac{\mu^\mu}{\hat{\gamma}^{\frac{\alpha\mu}{2}}} \sum_{i=1}^4 a_i \left(\mu\hat{\gamma}^{-\frac{\alpha}{2}} + b_i c\right)^{-\mu}, \qquad (19)
\end{aligned}
$$

where $\mu > 0$ and $\frac{\mu}{\hat{\gamma}^{\frac{\alpha}{2}}} + b_i c > 0$ (see Appendix R6). The values of a_i and b_i are to be fitted. The motivation for this method is that it may be easier to fit these parameters than it is to calculate the G-functions, which can be very complex.

3.4 $\alpha - \mu$'s relationship with other distributions

The $\alpha - \mu$ distribution is a flexible distribution and includes a number of distributions as special cases [21]. These special cases are given in Table 1.

4 $\alpha - \mu$ type model

Suppose that the number of clusters is not sufficient for the assumption of normality between the$X$'s and Y's to hold. The derivation of the $\alpha - \mu$ distribution from the elliptical class is a possible solution,

Distribution	μ	α
Gamma	-	1
Nakagami-m	-	2
Exponential	1	1
Weibull	1	-
One-sided Gaussian	$\frac{1}{2}$	2
Rayleigh	1	1

Table 1: $\alpha-\mu$ distribution's relationship with other distributions with " - " indicating that the parameter may assume any value.

resulting in the $\alpha-\mu$ type model. The $\alpha-\mu$ type's PDF, moments and SNR will be derived with special cases being considered.

4.1 Derivation

Theorem 3. *Let X_i and Y_i be mutually independent elliptical processes with $E(X_i) = E(Y_i) = 0$ and $var(X_i) = var(Y_i) = \frac{\hat{r}^\alpha}{2\mu}$, hence $X_i, Y_i \sim E\left(0, \frac{\hat{r}^\alpha}{2\mu}, h\right)$ where $h(\cdot)$ is a generator function. The $\alpha-$power envelope emanating from the elliptical assumption is defined by:*

$$R^\alpha = \sum_{i=1}^{\mu} \left(X_i^2 + Y_i^2\right),$$

where $\alpha, \mu > 0$. The PDF of the envelope R is given by:

$$f_R(r) = \frac{\alpha r^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \int_0^\infty t^\mu \exp\left[-\mu t\left(\frac{r}{\hat{r}}\right)^\alpha\right] W(t)\, dt, \qquad r > 0 \tag{20}$$

where $\hat{r} = \sqrt[\alpha]{E(R^\alpha)}$. This is referred to as the $\alpha-\mu$ type fading model.

Proof. Note that $\frac{X_i}{v(t)}|t \sim N(0,1)$ and $\frac{Y_i}{v(t)}|t \sim N(0,1)$, therefore $\sum_{i=1}^{\mu} \frac{X_i^2+Y_i^2}{v(t)^2}|t = \frac{R^\alpha}{v(t)^2}|t \sim \chi^2(2\mu)$ since $\sum_{i=1}^{\mu} \frac{X_i^2}{v(t)^2}|t \sim \chi^2(\mu)$ and $\sum_{i=1}^{\mu} \frac{Y_i^2}{v(t)^2}|t \sim \chi^2(\mu)$, where $v(t)^2 = \frac{\hat{r}^\alpha}{2\mu}$ (see Appendix R1 and R2).
Let $K(t) = \frac{R^\alpha}{v(t)^2}|t$ then

$$f_{K(t)|t}(k) = \frac{1}{2^\mu \Gamma(\mu)} k^{\mu-1} \exp\left(-\frac{k}{2}\right), \qquad k > 0$$

with $\frac{dK(t)}{dR^\alpha} = \frac{1}{v(t)^2}$, thus

$$\begin{aligned} f_{R^\alpha|t}(r^\alpha) &= f_{K(t)|t}(k)\frac{dK(t)}{dR^\alpha} \\ &= f_{K(t)|t}\left(\frac{r^\alpha}{v(t)^2}\right)\frac{1}{v(t)^2} \end{aligned}$$

$$= \frac{1}{2^{\mu}\Gamma(\mu)\, v(t)^{2}} \left(\frac{r^{\alpha}}{v(t)^{2}}\right)^{\mu-1} \exp\left(-\frac{r^{\alpha}}{2v(t)^{2}}\right).$$

To find $f_{R|t}(r)$, note that $\frac{dR^{\alpha}}{dR} = \alpha R^{\alpha-1}$, subsequently

$$\begin{aligned} f_{R|t}(r) &= f_{R^{\alpha}|t}(r^{\alpha}) \frac{dR^{\alpha}}{dR} \\ &= \frac{1}{2^{\mu}\Gamma(\mu)\, v(t)^{2}} \left(\frac{r^{\alpha}}{v(t)^{2}}\right)^{\mu-1} \exp\left(-\frac{r^{\alpha}}{2v(t)^{2}}\right) \alpha r^{\alpha-1} \\ &= \left(\frac{\alpha}{2^{\mu}\Gamma(\mu)\, v(t)^{2\mu}}\right) r^{\alpha\mu-1} \exp\left(-\frac{r^{\alpha}}{2v(t)^{2}}\right). \end{aligned}$$

The unconditional distribution, $f_R(r)$, follows:

$$\begin{aligned} f_R(r) &= \int_0^{\infty} f_{R|t}(r)\, W(t)\, dt \\ &= \int_0^{\infty} \frac{\alpha}{2^{\mu}\Gamma(\mu)\, v(t)^{2\mu}} r^{\alpha\mu-1} \exp\left(-\frac{r^{\alpha}}{2v(t)^{2}}\right) W(t)\, dt \\ &= \int_0^{\infty} \frac{\alpha}{2^{\mu}\Gamma(\mu)\left(\frac{\hat{r}^{\alpha}}{2\mu t}\right)^{\mu}} r^{\alpha\mu-1} \exp\left[-\frac{r^{\alpha}}{2\left(\frac{\hat{r}^{\alpha}}{2\mu t}\right)}\right] W(t)\, dt \\ &= \int_0^{\infty} \frac{\alpha r^{\alpha\mu-1}\mu^{\mu}t^{\mu}}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}} \exp\left[-\mu t\left(\frac{r}{\hat{r}}\right)^{\alpha}\right] W(t)\, dt \\ &= \frac{\alpha r^{\alpha\mu-1}\mu^{\mu}}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}} \int_0^{\infty} t^{\mu} \exp\left[-\mu t\left(\frac{r}{\hat{r}}\right)^{\alpha}\right] W(t)\, dt, \qquad r > 0. \end{aligned}$$

□

4.2 Special cases

The $\alpha - \mu$ type denotes a class with flexibility regarding the underlying distribution which is determined by $W(t)$. Considered in this paper are the special cases where the weighting function is either normal, the dirac delta function (referred to as normal weighting), or the gamma weighting function for a student's t distribution (referred to as t weighting). These two will be compared since the t distribution makes provision for cases where the data has heavier tails and allows us to consider the case where the number of clusters were not sufficient for the assumption of normality. The weighting functions are given in Table 2 (see [4]).

Function name	Weighting function $W(t)$
dirac delta	$\delta(t-1)$
gamma, $\nu > 0$	$\frac{\nu\left(\frac{\nu t}{2}\right)^{\frac{\nu}{2}-1}}{2\Gamma\left(\frac{\nu}{2}\right)\exp(\frac{\nu t}{2})}$

Table 2: Weighting functions

4.2.1 Normal weighting

Consider $W(t) = \delta(t-1)$, the dirac delta function. Then

$$f_R(r) = \frac{\alpha r^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \int_0^\infty t^\mu \exp\left[-\mu t\left(\frac{r}{\hat{r}}\right)^\alpha\right] \delta(t-1)\, dt.$$

Now let $x = t-1$, hence $t = x+1$ and $\frac{dt}{dx} = 1$ then

$$\begin{aligned} f_R(r) &= \frac{\alpha r^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \int_0^\infty (x+1)^\mu \exp\left[-\mu(x+1)\left(\frac{r}{\hat{r}}\right)^\alpha\right] \delta(x)\, dx \\ &= \frac{\alpha r^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \exp\left[-\mu\left(\frac{r}{\hat{r}}\right)^\alpha\right], \qquad r > 0. \end{aligned}$$

This is the same result obtained earlier, (4) and the PDF is illustrated in Figures 3a - 3c.

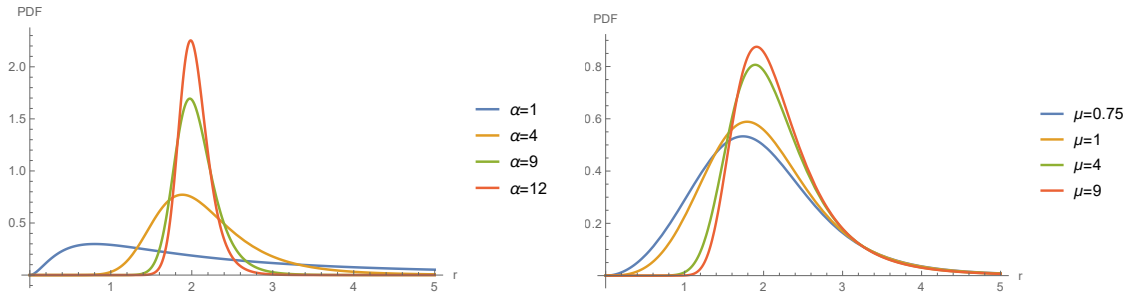
4.2.2 t weighting

Consider $W(t) = \frac{\nu\left(\frac{\nu t}{2}\right)^{\frac{\nu}{2}-1}}{2\Gamma\left(\frac{\nu}{2}\right)\exp(\frac{\nu t}{2})}$ where ν is the degrees of freedom of the t distribution. Then

$$\begin{aligned} f_R(r) &= \frac{\alpha r^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \int_0^\infty t^\mu \exp\left[-\mu t\left(\frac{r}{\hat{r}}\right)^\alpha\right] \frac{\nu\left(\frac{\nu t}{2}\right)^{\frac{\nu}{2}-1}}{2\Gamma\left(\frac{\nu}{2}\right)\exp\left(\frac{\nu t}{2}\right)} dt \\ &= \frac{\alpha r^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \frac{\left(\frac{1}{2}\right)^{\frac{\nu}{2}}\nu^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \int_0^\infty t^{(\mu+\frac{\nu}{2})-1} \exp\left[-t\left(\mu\left(\frac{r}{\hat{r}}\right)^\alpha + \frac{\nu}{2}\right)\right] dt \\ &= \frac{\alpha r^{\alpha\mu-1}\mu^\mu\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}} \frac{\Gamma\left(\mu+\frac{\nu}{2}\right)}{\left(\mu\left(\frac{r}{\hat{r}}\right)^\alpha + \frac{\nu}{2}\right)^{(\mu+\frac{\nu}{2})}}. \end{aligned} \tag{21}$$

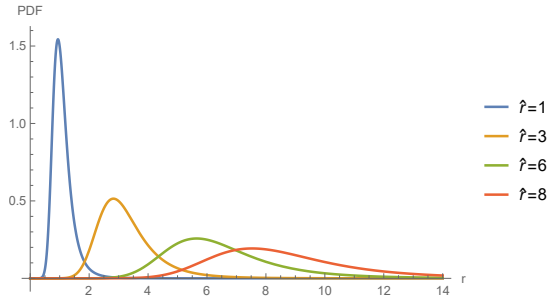
This is true for $\mu + \frac{\nu}{2} > 0$ and $\mu\left(\frac{r}{\hat{r}}\right)^\alpha + \frac{\nu}{2} > 0$ (see Appendix R6).

The PDF is illustrated in Figures 5a - 5d, a comparison with the case where the weighting function is the dirac delta function is made in Figure 6.

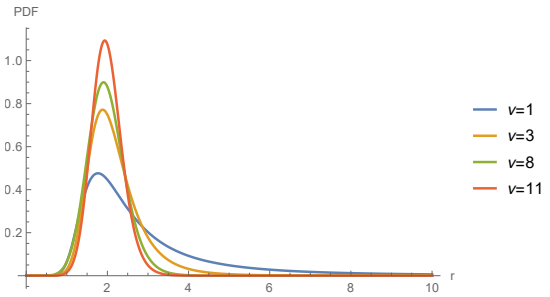


(a) (21) with $\mu = 3$, $\hat{r} = 2$, $\nu = 3$ with α varying.

(b) (21) with $\alpha = 4$, $\hat{r} = 2$, $\nu = 3$ with μ varying.



(c) (21) with $\alpha = 4$, $\mu = 3$, $\nu = 3$ with $\hat{r}$ varying.



(d) (21) with $\alpha = 4$, $\mu = 3$, $\hat{r} = 2$ with ν varying.

Figure 5: PDF of $\alpha - \mu$ type with t weighting.

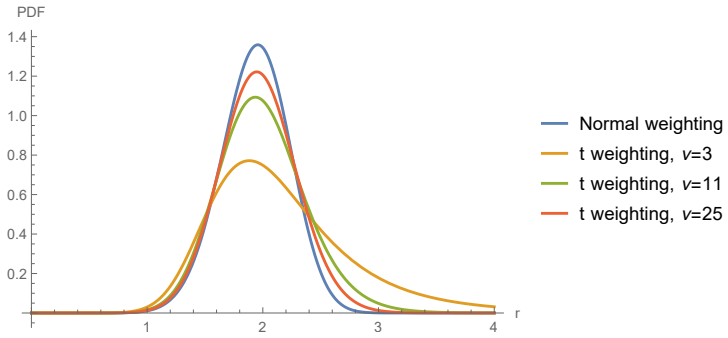


Figure 6: Comparison of (4) and (21) with $\alpha = 4$, $\mu = 3$, $\hat{r} = 2$ with ν varying.

4.3 Statistical characteristics

Some characteristics of the $\alpha - \mu$ type with PDF (20) will now be derived. These characteristics will first be derived in a general form and then special cases will be investigated.

4.3.1 CDF

The general form of the $\alpha - \mu$ type's CDF is

$$F_R(r) = \int_0^r f_R(y)\, dy$$

$$
\begin{aligned}
&= \int_0^r \frac{\alpha y^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}} \int_0^\infty t^\mu \exp\left[-\mu t\left(\frac{y}{\hat{r}}\right)^\alpha\right] W(t)\,dtdy \\
&= \frac{\alpha\mu^\mu}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}} \int_0^r y^{\alpha\mu-1} \int_0^\infty t^\mu \exp\left[-\mu t\left(\frac{y}{\hat{r}}\right)^\alpha\right] W(t)\,dtdy. \qquad (22)
\end{aligned}
$$

Now consider (22) with the normal- and t distribution weighting functions.

Normal weighting

Substituting the dirac delta function, $\delta(t-1)$ into (22)

$$
F_R(r) = \frac{\alpha\mu^\mu}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}} \int_0^r y^{\alpha\mu-1} \int_0^\infty t^\mu \exp\left[-\mu t\left(\frac{y}{\hat{r}}\right)^\alpha\right] \delta(t-1)\,dtdy.
$$

Consider the transformation $x = t-1$, hence $t = x+1$ and $\frac{dt}{dx} = 1$.

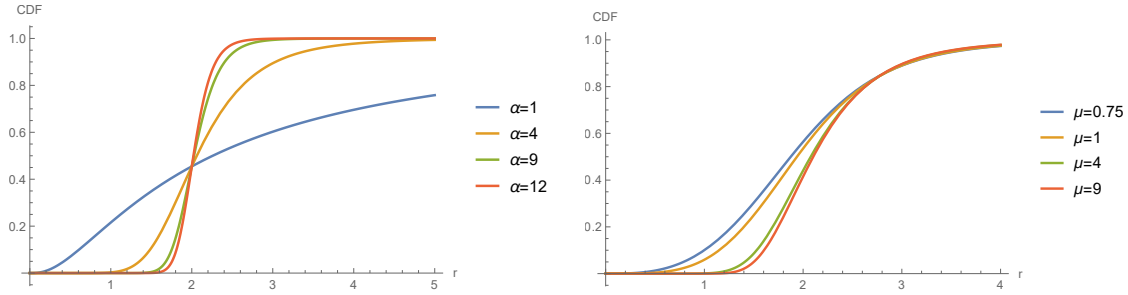
$$
\begin{aligned}
F_R(r) &= \frac{\alpha\mu^\mu}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}} \int_0^r y^{\alpha\mu-1} \int_0^\infty (x+1)^\mu \exp\left[-\mu(x+1)\left(\frac{y}{\hat{r}}\right)^\alpha\right] \delta(x)\,dxdy \\
&= \frac{\alpha\mu^\mu}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}} \int_0^r y^{\alpha\mu-1} \exp\left[-\mu\left(\frac{y}{\hat{r}}\right)^\alpha\right] dy \\
&= \frac{\alpha\mu^\mu}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}} \frac{\gamma\left(\mu, \mu\left(\frac{u}{\hat{r}}\right)^\alpha\right)}{\alpha\left(\frac{\mu}{\hat{r}^\alpha}\right)^\mu} \\
&= \frac{\gamma\left(\mu, \mu\left(\frac{u}{\hat{r}}\right)^\alpha\right)}{\Gamma(\mu)} \\
&\equiv (8).
\end{aligned}
$$

This is true for $u > 0, \mu > 0, \alpha > 0, \hat{r}^\alpha > 0$ (see Appendix R5 and R6). Graphic illustrations of this CDF is given by Figures 4a to 4c.

t weighting

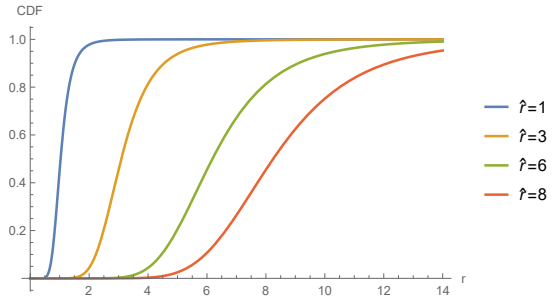
After substituting the relevant weighting function into (22):

$$
\begin{aligned}
F_R(r) &= \frac{\alpha\mu^\mu}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}} \int_0^r y^{\alpha\mu-1} \int_0^\infty t^\mu \exp\left[-\mu t\left(\frac{y}{\hat{r}}\right)^\alpha\right] \frac{\nu\left(\frac{\nu t}{2}\right)^{\frac{\nu}{2}-1}}{2\Gamma\left(\frac{\nu}{2}\right)\exp(\frac{\nu t}{2})} dtdy. \\
&= \frac{\alpha\mu^\mu \nu\nu^{\frac{\nu}{2}-1}}{\Gamma(\mu)\,\hat{r}^{\alpha\mu} 2\Gamma\left(\frac{\nu}{2}\right) 2^{\frac{\nu}{2}-1}} \int_0^r y^{\alpha\mu-1} \int_0^\infty t^{\mu+\frac{\nu}{2}-1} \exp\left(-t\left(\mu\left(\frac{y}{\hat{r}}\right)^\alpha + \frac{\nu}{2}\right)\right) dtdy \\
&= \frac{\alpha\mu^\mu \nu^{\frac{\nu}{2}}}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right) 2^{\frac{\nu}{2}}} \int_0^r y^{\alpha\mu-1} \frac{\Gamma\left(\mu+\frac{\nu}{2}\right)}{\left(\mu\left(\frac{y}{\hat{r}}\right)^\alpha + \frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}} dy \qquad (23) \\
&= \frac{\alpha\mu^\mu \nu^{\frac{\nu}{2}}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\Gamma(\mu)\,\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right) 2^{\frac{\nu}{2}}} \int_0^r y^{\alpha\mu-1} \frac{1}{\left(\mu\left(\frac{y}{\hat{r}}\right)^\alpha + \frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}} dy
\end{aligned}
$$

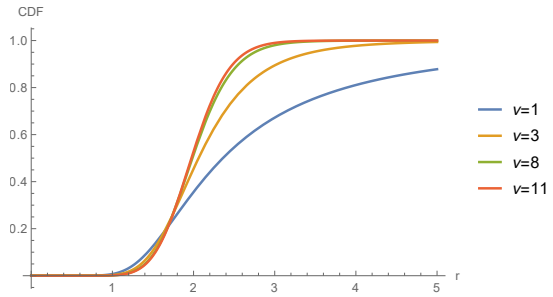


(a) (23) with $\mu = 3$, $\hat{r} = 2$, $\nu = 3$ and α varying.

(b) (23) with $\alpha = 4$, $\hat{r} = 2$, $\nu = 3$ and μ varying.



(c) (23) with $\alpha = 4$, $\mu = 3$, $\nu = 3$ and $\hat{r}$ varying.



(d) (23) with $\alpha = 4$, $\mu = 3$, $\hat{r} = 2$ and ν varying.

Figure 7: CDF of $\alpha - \mu$ type with t weighting.

$$= \frac{\alpha\mu^{\mu}\nu^{\frac{\nu}{2}}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\hat{r}^{\alpha\mu}2^{\frac{\nu}{2}}\Gamma\left(\mu\right)\Gamma\left(\frac{\nu}{2}\right)}\int_0^r y^{\alpha\mu-1}\left(\frac{\nu}{2}\right)^{-\left(\mu+\frac{\nu}{2}\right)}\left(\frac{2\mu}{\nu}\hat{r}^{-\alpha}y^{\alpha}+1\right)^{-\left(\mu+\frac{\nu}{2}\right)}dy$$
$$= \frac{\alpha\mu^{\mu}2^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\hat{r}^{\alpha\mu}\nu^{\mu}\Gamma\left(\mu\right)\Gamma\left(\frac{\nu}{2}\right)}\int_0^r y^{\alpha\mu-1}\left(\frac{2\mu}{\nu}\hat{r}^{-\alpha}y^{\alpha}+1\right)^{-\left(\mu+\frac{\nu}{2}\right)}dy,$$

where $\mu + \frac{\nu}{2} > 0$ and $\left(\mu\left(\frac{y}{\hat{r}}\right)^{\alpha} + \frac{\nu}{2}\right) > 0$ (see Appendix R6). Consider the transformation $z = y^{\alpha}$ then $y = z^{\frac{1}{\alpha}}$ and $dy = \frac{1}{\alpha}z^{\frac{1}{\alpha}-1}dz$. Subsequently:

$$F_R\left(r\right) = \frac{\alpha\mu^{\mu}2^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\hat{r}^{\alpha\mu}\nu^{\mu}\Gamma\left(\mu\right)\Gamma\left(\frac{\nu}{2}\right)}\int_0^{r^{\alpha}}\left(z^{\frac{1}{\alpha}}\right)^{\alpha\mu-1}\left(\frac{2\mu}{\nu}\hat{r}^{-\alpha}z+1\right)^{-\left(\mu+\frac{\nu}{2}\right)}\frac{1}{\alpha}z^{\frac{1}{\alpha}-1}dz$$
$$= \frac{\mu^{\mu}2^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\hat{r}^{\alpha\mu}\nu^{\mu}\Gamma\left(\mu\right)\Gamma\left(\frac{\nu}{2}\right)}\int_0^{r^{\alpha}} z^{\mu-1}\left(\frac{2\mu}{\nu}\hat{r}^{-\alpha}z+1\right)^{-\left(\mu+\frac{\nu}{2}\right)}dz$$
$$= \frac{\mu^{\mu}2^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\hat{r}^{\alpha\mu}\nu^{\mu}\Gamma\left(\mu\right)\Gamma\left(\frac{\nu}{2}\right)}\frac{r^{\alpha\mu}}{\mu}{}_2F_1\left(\mu+\frac{\nu}{2},\mu;1+\mu,-\frac{2}{\nu}\mu\hat{r}^{-\alpha}r^{\alpha}\right)$$
$$= {}_2F_1\left(\mu+\frac{\nu}{2},\mu;1+\mu,-\frac{2}{\nu}\mu\hat{r}^{-\alpha}r^{\alpha}\right)\frac{\mu^{\mu-1}2^{\mu}r^{\alpha\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\hat{r}^{\alpha\mu}\nu^{\mu}\Gamma\left(\mu\right)\Gamma\left(\frac{\nu}{2}\right)},$$

where $\mu > 0$ and ${}_2F_1\left(\cdot\right)$ denotes the Gauss hypergeometric function (see Appendix R6 and R7). This expression of the CDF is plotted and can be seen in Figures 7a, 7b, 7c and 7d with a comparison to the CDF of the $\alpha - \mu$ distribution in Figure 8.

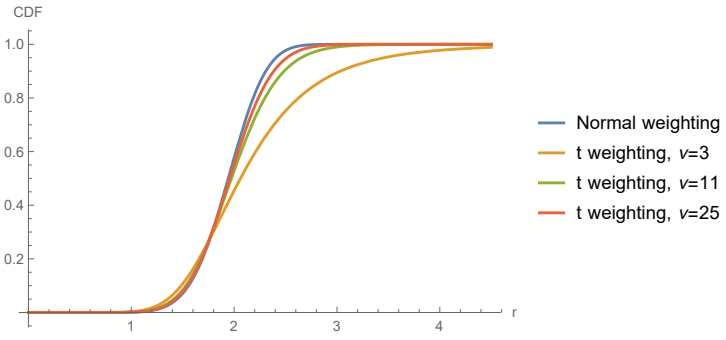


Figure 8: Comparison of CDFs (8) and (23) with $\alpha = 4$, $\mu = 3$, $\hat{r} = 2$ and ν varying.

4.3.2 MGF

To obtain the $\alpha - \mu$ type's MGF in the general case (20) is used:

$$\begin{aligned}
M_R(c) &= \mathrm{E}\left[\exp(cr)\right] \\
&= \int_0^\infty \exp(cr) f_R(r)\, dr \\
&= \int_0^\infty \frac{\alpha r^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \int_0^\infty t^\mu \exp\left[-\mu t\left(\frac{r}{\hat{r}}\right)^\alpha\right] W(t)\, dt \exp(cr)\, dr \\
&= \frac{\alpha\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \int_0^\infty r^{\alpha\mu-1} \exp(cr) \int_0^\infty t^\mu \exp\left[-\mu t\left(\frac{r}{\hat{r}}\right)^\alpha\right] W(t)\, dt dr.
\end{aligned}$$

Normal weighting

Substituting the solution for the PDF of R when considering normal weighting in, (4):

$$\begin{aligned}
M_R(c) &= \int_0^\infty \exp(cr) f_R(r)\, dr \\
&= \int_0^\infty \exp(cr) \frac{\alpha\mu^\mu r^{\alpha\mu-1}}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \exp\left[-\mu\left(\frac{r}{\hat{r}}\right)^\alpha\right] dr \\
&= \frac{\alpha\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \int_0^\infty r^{\alpha\mu-1} \sum_{k=0}^\infty \frac{(cr)^k}{k!} \exp\left[-\mu\left(\frac{r}{\hat{r}}\right)^\alpha\right] dr \\
&= \frac{\alpha\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \sum_{k=0}^\infty \frac{c^k}{k!} \int_0^\infty r^{\alpha\mu+k-1} \exp\left[-\mu\left(\frac{r}{\hat{r}}\right)^\alpha\right] dr \\
&= \frac{\alpha\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \sum_{k=0}^\infty \frac{c^k}{k!} \frac{\Gamma\left(\frac{\alpha\mu+k}{\alpha}\right)}{\alpha\left(\mu\hat{r}^{-\alpha}\right)^{\frac{\alpha\mu+k}{\alpha}}} \\
&= \frac{1}{\Gamma(\mu)} \sum_{k=0}^\infty \frac{c^k}{k!} \frac{\Gamma\left(\mu+\frac{k}{\alpha}\right)}{\left(\mu\hat{r}^{-\alpha}\right)^{\frac{k}{\alpha}}} \\
&\equiv (9).
\end{aligned}$$

This will hold when $\alpha > 0$, $\hat{r}^{\alpha} > 0$ and $\alpha\mu + k > 1$ (see Appendix R6).

t weighting

Substituting the solution for R's density found previously as (21) into the definition of a MGF:

$$\begin{aligned}
M_R(c) &= \int_0^{\infty} \exp(cr) \frac{\alpha r^{\alpha\mu-1}\mu^{\mu}\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}} \frac{\Gamma\left(\mu+\frac{\nu}{2}\right)}{\left(\mu\left(\frac{r}{\hat{r}}\right)^{\alpha}+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}} dr \\
&= \frac{\alpha\mu^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}} \int_0^{\infty} \exp(cr) r^{\alpha\mu-1} \frac{1}{\left(\mu\left(\frac{r}{\hat{r}}\right)^{\alpha}+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}} dr \\
&= \frac{\alpha\mu^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}} \int_0^{\infty} \sum_{k=0}^{\infty} \frac{(cr)^k}{k!} r^{\alpha\mu-1} \frac{1}{\left(\mu\left(\frac{r}{\hat{r}}\right)^{\alpha}+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}} dr \\
&= \frac{\alpha\mu^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}} \sum_{k=0}^{\infty} \frac{c^k}{k!} \int_0^{\infty} r^{(\alpha\mu+k)-1} \frac{1}{\left(\mu\left(\frac{r}{\hat{r}}\right)^{\alpha}+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}} dr \\
&= \frac{\alpha\mu^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}} \sum_{k=0}^{\infty} \frac{c^k}{k!} \int_0^{\infty} r^{(\alpha\mu+k)-1} \frac{1}{\left(\mu\hat{r}^{-\alpha}r^{\alpha}+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}-1\right)+1}} dr \\
&= \frac{\alpha\mu^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)\hat{r}^{\alpha\mu}2^{\frac{\nu}{2}}} \sum_{k=0}^{\infty} \frac{c^k}{k!} \frac{1}{\alpha\left(\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}} \left(\frac{\nu\hat{r}^{\alpha}}{2\mu}\right)^{\left(\mu+\frac{k}{\alpha}\right)} \frac{\Gamma\left(\mu+\frac{k}{\alpha}\right)\Gamma\left(\mu+\frac{\nu}{2}-\left(\mu+\frac{k}{\alpha}\right)\right)}{\Gamma\left(\mu+\frac{\nu}{2}\right)} \\
&= \frac{\alpha\mu^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)\hat{r}^{\alpha\mu}} \frac{1}{\alpha\left(\frac{\nu}{2}\right)^{\mu}\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}\Gamma\left(\mu+\frac{\nu}{2}\right)} \frac{\nu^{\mu}\hat{r}^{\alpha\mu}}{2^{\mu}\mu^{\mu}} \sum_{k=0}^{\infty} \frac{c^k}{k!} \left(\frac{\nu\hat{r}^{\alpha}}{2\mu}\right)^{\frac{k}{\alpha}} \Gamma\left(\mu+\frac{k}{\alpha}\right)\Gamma\left(\frac{\nu}{2}-\frac{k}{\alpha}\right) \\
&= \frac{1}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)} \sum_{k=0}^{\infty} \frac{c^k}{k!} \left(\frac{\nu\hat{r}^{\alpha}}{2\mu}\right)^{\frac{k}{\alpha}} \Gamma\left(\mu+\frac{k}{\alpha}\right)\Gamma\left(\frac{\nu}{2}-\frac{k}{\alpha}\right),
\end{aligned}$$

where $0 < \mu + \frac{k}{\alpha} < \mu + \frac{\nu}{2}$, $\frac{\nu}{2} \neq 0$ and $\mu\hat{r}^{-\alpha} \neq 0$ (see Appendix R6).

4.3.3 j^{th} moment

In a general form the j^{th} moment of the $\alpha - \mu$ type model is given by:

$$\begin{aligned}
m_j &= \mathrm{E}\left(R^j\right) \\
&= \int_0^{\infty} r^j f_R(r) dr \\
&= \int_0^{\infty} \frac{\alpha r^{\alpha\mu-1}\mu^{\mu}}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \int_0^{\infty} t^{\mu} \exp\left[-\mu t\left(\frac{r}{\hat{r}}\right)^{\alpha}\right] W(t)\, dt r^j dr \\
&= \frac{\alpha\mu^{\mu}}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \int_0^{\infty} r^{\alpha\mu+j-1} \int_0^{\infty} t^{\mu} \exp\left[-\mu t\left(\frac{r}{\hat{r}}\right)^{\alpha}\right] W(t)\, dt dr.
\end{aligned} \tag{24}$$

Normal weighting

Substituting (4) into (24)

$$
\begin{aligned}
m_j &= \int_0^\infty r^j \frac{\alpha r^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \exp\left[-\mu\left(\frac{r}{\hat{r}}\right)^\alpha\right] dr \\
&= \frac{\alpha\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \int_0^\infty r^{j+\alpha\mu-1} \exp\left[-\mu\left(\frac{r}{\hat{r}}\right)^\alpha\right] dr \\
&= \frac{\alpha\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \frac{\Gamma\left(\frac{\alpha\mu+j}{\alpha}\right)}{\alpha\left(\mu\hat{r}^{-\alpha}\right)^{\frac{\alpha\mu+j}{\alpha}}} \\
&= \frac{\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \frac{\Gamma\left(\mu+\frac{j}{\alpha}\right)\hat{r}^{\alpha\mu}\hat{r}^j}{\mu^\mu\mu^{\frac{j}{\alpha}}} \\
&= \frac{\hat{r}^j\Gamma\left(\mu+\frac{j}{\alpha}\right)}{\mu^{\frac{j}{\alpha}}\Gamma(\mu)} \\
&\equiv (10).
\end{aligned}
$$

This will hold for $\alpha > 0$, $\hat{r}^\alpha > 0$ and $\alpha\mu + j > 1$ (see Appendix R6).

t weighting

Substituting (21) into (24)

$$
\begin{aligned}
m_j &= \int_0^\infty r^j \frac{\alpha r^{\alpha\mu-1}\mu^\mu\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}} \frac{\Gamma\left(\mu+\frac{\nu}{2}\right)}{\left(\mu\left(\frac{r}{\hat{r}}\right)^\alpha+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}} dr \\
&= \frac{\alpha\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}} \int_0^\infty r^{\alpha\mu+j-1} \frac{1}{\left(\mu\left(\frac{r}{\hat{r}}\right)^\alpha+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}} dr \\
&= \frac{\alpha\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}} \int_0^\infty r^{(\alpha\mu+j)-1} \frac{1}{\left(\mu\hat{r}^{-\alpha}r^\alpha+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}-1\right)+1}} dr \\
&= \frac{\alpha\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma(\mu)\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right)} \frac{1}{\alpha\left(\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}} \left(\frac{\nu\hat{r}^\alpha}{2\mu}\right)^{\left(\mu+\frac{j}{\alpha}\right)} \frac{\Gamma\left(\mu+\frac{j}{\alpha}\right)\Gamma\left(\mu+\frac{\nu}{2}-\left(\mu+\frac{j}{\alpha}\right)\right)}{\Gamma\left(\mu+\frac{\nu}{2}\right)} \\
&= \frac{\mu^\mu\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma(\mu)\hat{r}^{\alpha\mu}\Gamma\left(\frac{\nu}{2}\right)} \frac{1}{\left(\frac{\nu}{2}\right)^\mu\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}} \left(\frac{\nu}{2}\right)^\mu \left(\frac{\nu}{2}\right)^{\frac{j}{\alpha}} \frac{\hat{r}^{\alpha\mu}}{\mu^\mu} \frac{\hat{r}^j}{\mu^{\frac{j}{\alpha}}} \Gamma\left(\mu+\frac{j}{\alpha}\right)\Gamma\left(\frac{\nu}{2}-\frac{j}{\alpha}\right) \\
&= \frac{\Gamma\left(\mu+\frac{j}{\alpha}\right)\Gamma\left(\frac{\nu}{2}-\frac{j}{\alpha}\right)}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)} \left(\frac{\nu\hat{r}^\alpha}{2\mu}\right)^{\frac{j}{\alpha}},
\end{aligned}
$$

where $0 < \mu + \frac{j}{\alpha} < \mu + \frac{\nu}{2}$, $\frac{\nu}{2} \neq 0$ and $\mu\hat{r}^{-\alpha} \neq 0$ (see Appendix R6).

4.4 SNR

To measure the performance of the $\alpha - \mu$ type the instantaneous SNR (see Section 2.2) and outage probability will be derived for the general- and relevant special cases [17, 18].

4.4.1 PDF and Outage probability

The general SNR of the $\alpha - \mu$ type model will be investigated first.

$$\begin{aligned} f_\gamma(\gamma) &= f_R(r)\frac{dr}{d\gamma} \qquad (25)\\ &= \frac{\alpha r^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}}\int_0^\infty t^\mu \exp\left[-\mu t\left(\frac{r}{\hat{r}}\right)^\alpha\right] W(t)\,dt \frac{\hat{r}}{2}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}\frac{1}{\hat{\gamma}}\\ &= \frac{\alpha\mu^\mu r^{\alpha\mu}\hat{r}}{2\Gamma(\mu)\hat{r}^{\alpha\mu}}\left(\hat{r}\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}\frac{1}{\hat{\gamma}}\int_0^\infty t^\mu \exp\left[-\mu t\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^\alpha\right] W(t)\,dt\\ &= \frac{\alpha\mu^\mu}{2\Gamma(\mu)}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha\mu}\gamma^{-1}\int_0^\infty t^\mu \exp\left[-\mu t\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^\alpha\right] W(t)\,dt \qquad (26)\end{aligned}$$

Making use of (26) the outage probability for general $\alpha - \mu$ type model is derived.

$$\begin{aligned} F_\gamma(\gamma_{OP}) &= \int_0^{\gamma_{OP}} f_\gamma(\gamma)\,d\gamma\\ &= \int_0^{\gamma_{OP}} \frac{\alpha\mu^\mu}{2\Gamma(\mu)}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha\mu}\gamma^{-1}\int_0^\infty t^\mu \exp\left[-\mu t\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^\alpha\right] W(t)\,dt d\gamma\\ &= \frac{\alpha\mu^\mu}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^{\gamma_{OP}} \gamma^{\frac{\alpha\mu}{2}-1}\int_0^\infty t^\mu \exp\left[-\mu t\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^\alpha\right] W(t)\,dt d\gamma\end{aligned}$$

The special cases for the normal- and t distribution weighting functions will now be considered.

Normal weighting

Considering (25) and substitute in (4) and find the instantaneous SNR.

$$\begin{aligned} f_\gamma(\gamma) &= \frac{\alpha r^{\alpha\mu-1}\mu^\mu}{\Gamma(\mu)\hat{r}^{\alpha\mu}}\exp\left[-\mu\left(\frac{r}{\hat{r}}\right)^\alpha\right]\frac{\hat{r}}{2}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}\frac{1}{\hat{\gamma}}\\ &= \frac{\alpha\mu^\mu r^{\alpha\mu-1}}{2\Gamma(\mu)\hat{r}^{\alpha\mu-1}}\exp\left[-\mu\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^\alpha\right]\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}\frac{1}{\hat{\gamma}}\\ &= \frac{\alpha\mu^\mu}{2\Gamma(\mu)}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha\mu-1}\exp\left[-\mu\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^\alpha\right]\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}\frac{1}{\hat{\gamma}}\\ &= \frac{\alpha\mu^\mu}{2\Gamma(\mu)}\gamma^{\frac{\alpha\mu}{2}-1}\hat{\gamma}^{-\frac{\alpha\mu}{2}}\exp\left[-\mu\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^\alpha\right]\\ &\equiv (12).\end{aligned}$$

Using (12) as a starting point to find the outage probability:

$$F_\gamma(\gamma_{OP}) = \int_0^{\gamma_{OP}} \frac{\alpha\mu^\mu\gamma^{\frac{\alpha\mu}{2}-1}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\exp\left[-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right]d\gamma$$

$$
\begin{aligned}
&= \frac{\alpha\mu^{\mu}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \int_0^{\gamma_{OP}} \gamma^{\frac{\alpha\mu}{2}-1} \exp\left[-\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right] d\gamma \\
&= \frac{\alpha\mu^{\mu}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \frac{\gamma\left(\mu, \mu\left(\frac{\gamma_{OP}}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right)}{\frac{\alpha}{2}\left(\mu\hat{\gamma}^{-\frac{\alpha}{2}}\right)^{\mu}} \\
&= \frac{\gamma\left(\mu, \mu\left(\frac{\gamma_{OP}}{\hat{\gamma}}\right)^{\frac{\alpha}{2}}\right)}{\Gamma(\mu)} \\
&\equiv (13),
\end{aligned}
$$

when $\mu > 0$, $\hat{\gamma}^{-\frac{\alpha}{2}} > 0$, $\gamma_{OP} > 0$ and $\alpha > 0$ (see Appendix R5 and R6).

t weighting

Substituting (21) into (25) to find the instantaneous SNR:

$$
\begin{aligned}
f_{\gamma}(\gamma) &= \frac{\alpha r^{\alpha\mu-1}\mu^{\mu}}{\Gamma(\mu)\hat{r}^{\alpha\mu}} \frac{\nu^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}} \frac{\Gamma\left(\mu+\frac{\nu}{2}\right)}{\left(\mu\left(\frac{\gamma}{\hat{\gamma}}\right)^{\alpha}+\frac{\nu}{2}\right)^{\mu+\frac{\nu}{2}}} \frac{\hat{r}}{2}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}\frac{1}{\hat{\gamma}} \\
&= \frac{\alpha\mu^{\mu}\nu^{\frac{\nu}{2}}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}}\left(\frac{r}{\hat{r}}\right)^{\alpha\mu-1} \frac{1}{\left(\mu\left(\frac{r}{\hat{r}}\right)^{\alpha}+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}\frac{1}{\hat{\gamma}} \\
&= \frac{\alpha\mu^{\mu}\nu^{\frac{\nu}{2}}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha\mu-1} \frac{1}{\left(\mu\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha}+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{-1}\frac{1}{\hat{\gamma}} \\
&= \frac{\alpha\mu^{\mu}\nu^{\frac{\nu}{2}}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}}\gamma^{\frac{\alpha\mu}{2}-1}\hat{\gamma}^{-\frac{\alpha\mu}{2}} \frac{1}{\left(\mu\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha}+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}}.
\end{aligned} \tag{27}
$$

To derive the outage probability (27) is our departure point.

$$
\begin{aligned}
F_{\gamma}(\gamma_{OP}) &= \int_0^{\gamma_{OP}} \frac{\alpha\mu^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}\left(\mu\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha}+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}}\gamma^{-1}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha\mu} d\gamma \\
&= \frac{\alpha\mu^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}\hat{\gamma}^{\frac{\alpha\mu}{2}}} \int_0^{\gamma_{OP}} \left(\mu\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{\frac{\alpha}{2}}+\frac{\nu}{2}\right)^{-\left(\mu+\frac{\nu}{2}\right)}\gamma^{\frac{\alpha\mu}{2}-1}d\gamma \\
&= \frac{\alpha\mu^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}\hat{\gamma}^{\frac{\alpha\mu}{2}}} \int_0^{\gamma_{OP}} \left(\frac{\nu}{2}\right)^{-\left(\mu+\frac{\alpha}{2}\right)}\left(\mu\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}\frac{2}{\nu}+1\right)^{-\left(\mu+\frac{\nu}{2}\right)}\gamma^{\frac{\alpha\mu}{2}-1}d\gamma.
\end{aligned} \tag{28}
$$

Consider the transformation $z = \gamma^{\frac{\alpha}{2}}$ thus $\gamma = z^{\frac{2}{\alpha}}$ and $d\gamma = dz\frac{2}{\alpha}z^{\frac{2}{\alpha}-1}$. Subsequently:

$$
\begin{aligned}
F_{\gamma}\left(\gamma_{OP}\right) &= \frac{\alpha\mu^{\mu}2^{\mu-1}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\nu^{\mu}\hat{\gamma}^{\frac{\alpha\mu}{2}}\Gamma\left(\mu\right)\Gamma\left(\frac{\nu}{2}\right)}\int_{0}^{\gamma_{OP}^{\frac{\alpha}{2}}}\left(\mu\hat{\gamma}^{-\frac{\alpha}{2}}\frac{2}{\nu}z+1\right)^{-\left(\mu+\frac{\nu}{2}\right)}\left(z^{\frac{2}{\alpha}}\right)^{\frac{\alpha\mu}{2}-1}\frac{2}{\alpha}z^{\frac{2}{\alpha}-1}dz \\
&= \frac{\mu^{\mu}2^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\nu^{\mu}\hat{\gamma}^{\frac{\alpha\mu}{2}}\Gamma\left(\mu\right)\Gamma\left(\frac{\nu}{2}\right)}\int_{0}^{\gamma_{OP}^{\frac{\alpha}{2}}}\left(\mu\hat{\gamma}^{-\frac{\alpha}{2}}\frac{2}{\nu}z+1\right)^{-\left(\mu+\frac{\nu}{2}\right)}z^{\mu-1}dz \\
&= \frac{\mu^{\mu}2^{\mu}\Gamma\left(\mu+\frac{\nu}{2}\right)}{\nu^{\mu}\hat{\gamma}^{\frac{\alpha\mu}{2}}\Gamma\left(\mu\right)\Gamma\left(\frac{\nu}{2}\right)}\frac{\gamma_{OP}^{\frac{\alpha\mu}{2}}}{\mu}{}_2F_1\left(\mu+\frac{\nu}{2},\mu;1+\mu,-\frac{2}{\nu}\mu\hat{\gamma}^{-\frac{\alpha}{2}}\gamma_{OP}^{\frac{\alpha}{2}}\right) \\
&=_2 F_1\left(\mu+\frac{\nu}{2},\mu;1+\mu,-\frac{2}{\nu}\mu\hat{\gamma}^{-\frac{\alpha}{2}}\gamma_{OP}^{\frac{\alpha}{2}}\right)\frac{\Gamma\left(\mu+\frac{\nu}{2}\right)2^{\mu}\mu^{\mu-1}\gamma_{OP}^{\frac{\alpha\mu}{2}}}{\Gamma\left(\mu\right)\Gamma\left(\frac{\nu}{2}\right)\nu^{\mu}\hat{\gamma}^{\frac{\alpha\mu}{2}}},
\end{aligned}
$$

where $\mu > 0$ (see Appendix R6 and R7).

4.4.2 Laplace transform

Using the general expression for the PDF of the SNR (26) the Laplace transform of the SNR can be found in a general form. The methods contained in [9, 16] will again be employed to obtain alternative closed form expressions.

The general expression's derivation follows.

$$
\begin{aligned}
L\left(c\right) &= \mathrm{E}\left[\exp\left(-c\gamma\right)\right] \\
&= \int_{0}^{\infty} f_{\gamma}\left(\gamma\right)\exp\left(-c\gamma\right)d\gamma \\
&= \int_{0}^{\infty}\frac{\alpha\mu^{\mu}}{2\Gamma\left(\mu\right)}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha\mu}\gamma^{-1}\int_{0}^{\infty}t^{\mu}\exp\left[-\mu t\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha}\right]W\left(t\right)dt\exp\left(-c\gamma\right)d\gamma \\
&= \frac{\alpha\mu^{\mu}}{2\Gamma\left(\mu\right)}\int_{0}^{\infty}\exp\left(-c\gamma\right)\gamma^{\frac{\alpha\mu}{2}}\hat{\gamma}^{-\frac{\alpha\mu}{2}}\gamma^{-1}\int_{0}^{\infty}t^{\mu}\exp\left[-\mu t\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}\right]W\left(t\right)dtd\gamma \\
&= \frac{\alpha\mu^{\mu}}{2\Gamma\left(\mu\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_{0}^{\infty}\sum_{k=0}^{\infty}\frac{\left(-c\gamma\right)^{k}}{k!}\gamma^{\frac{\alpha\mu}{2}-1}\int_{0}^{\infty}t^{\mu}\sum_{u=0}^{\infty}\frac{\left(-\mu t\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}\right)^{u}}{u!}W\left(t\right)dtd\gamma \\
&= \frac{\alpha\mu^{\mu}}{2\Gamma\left(\mu\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\sum_{k=0}^{\infty}\sum_{u=0}^{\infty}\frac{\left(-c\right)^{k}}{k!}\frac{\left(-\mu\right)^{u}}{u!\hat{\gamma}^{\frac{\alpha u}{2}}}\int_{0}^{\infty}\gamma^{\frac{\alpha(\mu+u)}{2}+k-1}\int_{0}^{\infty}t^{\mu+u}W\left(t\right)dtd\gamma.
\end{aligned}
$$

Approach 1 - General Case

This approach is similar to that of Magableh [9] (see Appendix R8).

$$
\begin{aligned}
L\left(c\right) &= \mathrm{E}\left[\exp\left(-c\gamma\right)\right] \\
&= \int_{0}^{\infty} f_{\gamma}\left(\gamma\right)\exp\left(-c\gamma\right)d\gamma \\
&= \int_{0}^{\infty}\frac{\alpha\mu^{\mu}}{2\Gamma\left(\mu\right)}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha\mu}\gamma^{-1}\int_{0}^{\infty}t^{\mu}\exp\left[-\mu t\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha}\right]W\left(t\right)dt\exp\left(-c\gamma\right)d\gamma
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\alpha\mu^{\mu}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \int_0^{\infty} \exp(-c\gamma)\,\gamma^{\frac{\alpha\mu}{2}-1} \int_0^{\infty} t^{\mu} \exp\left(-\mu t\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}\right) W(t)\,dtd\gamma \\
&= \frac{\alpha\mu^{\mu}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \int_0^{\infty} (c\gamma)^{-1} \left[(c\gamma)^{1}\exp(-c\gamma)\right]\gamma^{\frac{\alpha\mu}{2}-1}\times \\
&\int_0^{\infty} \left(\mu t\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}\right)^{-1} \left[\left(\mu t\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}\right)^{1}\exp\left(-\mu t\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}\right)\right] W(t)\,t^{\mu}dtd\gamma \\
&= \frac{\alpha\mu^{\mu}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \int_0^{\infty} (c\gamma)^{-1} \left[(c\gamma)^{1} G_{0,1}^{1,0}\left(c\gamma \,\middle|\, \begin{matrix} - \\ 0 \end{matrix}\right)\right]\gamma^{\frac{\alpha\mu}{2}-1}\times \\
&\int_0^{\infty} \left(\mu t\gamma^{\frac{\alpha}{\mu}}\hat{\gamma}^{-\frac{\alpha}{2}}\right)^{-\mu} \left[\left(\mu t\gamma^{\frac{\alpha}{\mu}}\hat{\gamma}^{-\frac{\alpha}{2}}\right)^{\mu} G_{0,1}^{1,0}\left(\mu t\gamma^{\frac{\alpha}{\mu}}\hat{\gamma}^{-\frac{\alpha}{2}} \,\middle|\, \begin{matrix} - \\ 0 \end{matrix}\right)\right] W(t)\,t^{\mu}dtd\gamma \\
&= \frac{\alpha\mu^{\mu}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}} \int_0^{\infty} G_{0,1}^{1,0}\left(c\gamma \,\middle|\, \begin{matrix} - \\ 0 \end{matrix}\right)\gamma^{\frac{\alpha\mu}{2}-1} \int_0^{\infty} G_{0,1}^{1,0}\left(\mu t\gamma^{\frac{\alpha}{\mu}}\hat{\gamma}^{-\frac{\alpha}{2}} \,\middle|\, \begin{matrix} - \\ \mu \end{matrix}\right) W(t)\,t^{\mu}dtd\gamma.
\end{aligned}
$$

Approach 2 - General Case

The Laplace transform will again be considered and rewritten in a form such that the approximation (18) can be implemented.

$$
\begin{aligned}
L(c) &= \mathrm{E}\left[\exp(-c\gamma)\right] \\
&= \int_0^{\infty} f_{\gamma}(\gamma)\exp(-c\gamma)\,d\gamma \\
&= \int_0^{\infty} \frac{\alpha\mu^{\mu}}{2\Gamma(\mu)}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha\mu}\gamma^{-1}\int_0^{\infty} t^{\mu}\exp\left[-\mu t\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha}\right] W(t)\,dt\exp(-c\gamma)\,d\gamma \\
&= \frac{\alpha\mu^{\mu}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^{\infty}\exp(-c\gamma)\,\gamma^{\frac{\alpha\mu}{2}-1}\int_0^{\infty} t^{\mu}\exp\left[-\mu t\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}\right] W(t)\,dtd\gamma.
\end{aligned}
$$

Consider the transformation $\gamma^{\frac{\alpha}{2}} = z$ thus $\gamma = z^{\frac{2}{\alpha}}$ and $\frac{d\gamma}{dz} = \frac{2}{\alpha}z^{\frac{2}{\alpha}-1}$.

$$
L(c) = \frac{\alpha\mu^{\mu}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^{\infty}\exp\left(-cz^{\frac{2}{\alpha}}\right)z^{\mu-\frac{2}{\alpha}}\int_0^{\infty} t^{\mu}\exp\left[-\mu tz\hat{\gamma}^{-\frac{\alpha}{2}}\right] W(t)\,dtdz
$$

Substituting the approximation (18) in:

$$
\begin{aligned}
L(c) &\approx \frac{\alpha\mu^{\mu}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^{\infty}\sum_{i=1}^{4} a_i\exp(-b_icz)\,z^{\mu-\frac{2}{\alpha}}\int_0^{\infty} t^{\mu}\exp\left[-\mu tz\hat{\gamma}^{-\frac{\alpha}{2}}\right] W(t)\,dtdz \\
&= \frac{\alpha\mu^{\mu}}{2\Gamma(\mu)\hat{\gamma}^{\frac{\alpha\mu}{2}}}\sum_{i=1}^{4} a_i\int_0^{\infty}\exp(-b_icz)\,z^{\mu-\frac{2}{\alpha}}\int_0^{\infty} t^{\mu}\exp\left[-\mu tz\hat{\gamma}^{-\frac{\alpha}{2}}\right] W(t)\,dtdz
\end{aligned}
$$

Normal weighting

The PDF of the SNR in the case of the $\alpha-\mu$ type when the normal weighting function is used is the same as the SNR PDF found for the $\alpha-\mu$ distribution, consequently the expression for the Laplace transform is given by (15). Due to the PDF being the same the alternative closed form expressions are given by (17) and (19) respectively.

t weighting

Using (27) the SNR's Laplace transform will now be derived.

$$
\begin{aligned}
L(c) &= \mathrm{E}\left[\exp\left(-c\gamma\right)\right] \\
&= \int_0^\infty f_\gamma(\gamma)\exp\left(-c\gamma\right)d\gamma \\
&= \int_0^\infty \frac{\alpha\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}\left(\mu\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^\alpha+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}}\gamma^{-1}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha\mu}\exp\left(-c\gamma\right)d\gamma \\
&= \frac{\alpha\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty\left(\mu\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}+\frac{\nu}{2}\right)^{-\left(\mu+\frac{\nu}{2}\right)}\gamma^{\frac{\alpha\mu}{2}-1}\exp\left(-c\gamma\right)d\gamma \qquad (29) \\
&= \frac{\alpha\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty\left(\frac{\nu}{2}\right)^{-\left(\mu+\frac{\nu}{2}\right)}\left(\mu\frac{2}{\nu}\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}+1\right)^{-\left(\mu+\frac{\nu}{2}\right)}\gamma^{\frac{\alpha\mu}{2}-1}\exp\left(-c\gamma\right)d\gamma \\
&= \frac{\alpha\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)2^{\mu-1}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)\nu^\mu\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty\sum_{k=0}^\infty\frac{\left(\mu+\frac{\nu}{2}\right)_k}{k!}\left(-\mu\frac{2}{\nu}\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}\right)^k\gamma^{\frac{\alpha\mu}{2}-1}\exp\left(-c\gamma\right)d\gamma \\
&= \frac{\alpha\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)2^{\mu-1}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)\nu^\mu\hat{\gamma}^{\frac{\alpha\mu}{2}}}\sum_{k=0}^\infty\frac{\left(\mu+\frac{\nu}{2}\right)_k}{k!}(-1)^k\left(\frac{2}{\nu}\mu\hat{\gamma}^{-\frac{\alpha}{2}}\right)^k\int_0^\infty\gamma^{\frac{\alpha k}{2}}\gamma^{\frac{\alpha\mu}{2}-1}\exp\left(-c\gamma\right)d\gamma \\
&= \frac{\alpha\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)2^{\mu-1}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)\nu^\mu\hat{\gamma}^{\frac{\alpha\mu}{2}}}\sum_{k=0}^\infty\frac{1}{k!}\frac{\Gamma\left(\mu+\frac{\nu}{2}+k\right)}{\Gamma\left(\mu+\frac{\nu}{2}\right)}(-1)^k\left(\frac{2}{\nu}\mu\hat{\gamma}^{-\frac{\alpha}{2}}\right)^k\int_0^\infty\gamma^{\frac{\alpha k}{2}+\frac{\alpha\mu}{2}-1}\exp\left(-c\gamma\right)d\gamma \\
&= \frac{\alpha\mu^\mu 2^{\mu-1}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)\hat{\gamma}^{\frac{\alpha\mu}{2}}\nu^\mu}\sum_{k=0}^\infty\frac{(-1)^k}{c^{\frac{\alpha(\mu+k)}{2}}}\frac{\Gamma\left(\mu+\frac{\nu}{2}+k\right)\Gamma\left(\frac{\alpha(\mu+k)}{2}\right)}{k!}\left(\frac{2\mu}{\hat{\gamma}^{\frac{\alpha}{2}}\nu}\right)^k, \qquad (30)
\end{aligned}
$$

where $c>0$ and $\frac{\alpha(\mu+k)}{2}>0$ (see Appendix R6).

Approach 1

Working with the Laplace transform and substituting (27) in:

$$
\begin{aligned}
L(c) &= \mathrm{E}\left[\exp\left(-c\gamma\right)\right] \\
&= \int_0^\infty f_\gamma(\gamma)\exp\left(-c\gamma\right)d\gamma
\end{aligned}
$$

$$
\begin{aligned}
&= \int_0^\infty \frac{\alpha\mu^\mu \Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}\left(\mu\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^\alpha+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}}\gamma^{-1}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha\mu}\exp(-c\gamma)\,d\gamma \\
&= \frac{\alpha\mu^\mu \Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty \gamma^{\frac{\alpha\mu}{2}-1}\left(\mu\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}+\frac{\nu}{2}\right)^{-\left(\mu+\frac{\nu}{2}\right)}\exp(-c\gamma)\,d\gamma \\
&= \frac{\alpha\mu^\mu \Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty \gamma^{\frac{\alpha\mu}{2}-1}\left(\frac{\nu}{2}\right)^{-\left(\mu+\frac{\nu}{2}\right)}\left(\mu\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}\frac{\nu}{2}+1\right)^{-\left(\mu+\frac{\nu}{2}\right)}(c\gamma)^{-1}\left[(c\gamma)\,G_{0,1}^{1,0}\left(c\gamma\,\middle|\,\begin{matrix}-\\0\end{matrix}\right)\right]d\gamma \\
&= \frac{\alpha\mu^\mu \Gamma\left(\mu+\frac{\nu}{2}\right)2^{\mu-1}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)\nu^\mu\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty \gamma^{\frac{\alpha\mu}{2}-1}\frac{1}{\Gamma\left(\mu+\frac{\nu}{2}\right)}G_{1,1}^{1,1}\left(\mu\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}\frac{\nu}{2}\,\middle|\,\begin{matrix}1-\mu-\frac{\nu}{2}\\0\end{matrix}\right)G_{0,1}^{1,0}\left(c\gamma\,\middle|\,\begin{matrix}-\\0\end{matrix}\right)d\gamma \\
&= \frac{\alpha\mu^\mu 2^{\mu-1}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)\nu^\mu\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty \gamma^{\frac{\alpha\mu}{2}-1}G_{0,1}^{1,0}\left(c\gamma\,\middle|\,\begin{matrix}-\\0\end{matrix}\right)G_{1,1}^{1,1}\left(\mu\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}\frac{\nu}{2}\,\middle|\,\begin{matrix}1-\mu-\frac{\nu}{2}\\0\end{matrix}\right)d\gamma \\
&= \frac{\alpha\mu^\mu 2^{\mu-1}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)\nu^\mu\hat{\gamma}^{\frac{\alpha\mu}{2}}}c^{-\frac{\alpha\mu}{2}}(2\pi)^{\frac{1}{2}(1-\alpha)-1}\alpha^{-\frac{1}{2}+\frac{\alpha\mu}{2}}2^{\mu+\frac{\nu}{2}}\times \\
&G_{2+\alpha,2}^{2,2+\alpha}\left(\left(\frac{2\mu}{\hat{\gamma}^{\frac{\alpha}{2}}\nu}\right)^2\left(\frac{\alpha}{c}\right)^\alpha\,\middle|\,\begin{matrix}\Delta\left(2,1-\mu-\frac{\nu}{2}\right),\Delta\left(\alpha,1-\frac{\alpha\mu}{2}\right)\\ \Delta(2,0)\end{matrix}\right),
\end{aligned}
$$

(see Appendix R8) where $\Delta(x,y)$ is a sequence with x parameters given by:

$$\frac{y}{x},\frac{y+1}{x},\ldots,\frac{y+x-1}{x}.$$

Thus $\Delta\left(2,1-\mu-\frac{\nu}{2}\right)=\frac{1-\mu-\frac{\nu}{2}}{2},\frac{2-\mu-\frac{\nu}{2}}{2}$, $\Delta\left(\alpha,1-\frac{\alpha\mu}{2}\right)=\frac{1-\frac{\alpha\mu}{2}}{\alpha},\frac{2-\frac{\alpha\mu}{2}}{\alpha},\ldots,\frac{\alpha-\frac{\alpha\mu}{2}}{\alpha}$ and $\Delta(2,0)=0,\frac{1}{2}$ consequently:

$$
\begin{aligned}
L(c) = {} & \frac{\alpha^{\frac{1}{2}(\alpha\mu+1)}\mu^\mu 2^{2\mu+\frac{\nu}{2}-\frac{1}{2}\alpha-\frac{3}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)\nu^\mu\hat{\gamma}^{\frac{\alpha\mu}{2}}}c^{-\frac{\alpha\mu}{2}}(\pi)^{-\frac{1}{2}(1+\alpha)}\times \\
& G_{2+\alpha,2}^{2,2+\alpha}\left(\left(\frac{2\mu}{\hat{\gamma}^{\frac{\alpha}{2}}\nu}\right)^2\left(\frac{\alpha}{c}\right)^\alpha\,\middle|\,\begin{matrix}\frac{1-\mu-\frac{\nu}{2}}{2},\frac{2-\mu-\frac{\nu}{2}}{2},\frac{1-\frac{\alpha\mu}{2}}{\alpha},\frac{2-\frac{\alpha\mu}{2}}{\alpha},\ldots,\frac{\alpha-\frac{\alpha\mu}{2}}{\alpha}\\ 0,\frac{1}{2}\end{matrix}\right).
\end{aligned}
\tag{31}
$$

Approach 2

Working with the Laplace transform once more and as with the previous approach substituting (27) in an approximate expression will be obtained.

$$
\begin{aligned}
L(c) &= \mathrm{E}\left[\exp(-c\gamma)\right] \\
&= \int_0^\infty f_\gamma(\gamma)\exp(-c\gamma)\,d\gamma
\end{aligned}
$$

$$= \int_0^\infty \frac{\alpha\mu^\mu \Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}\left(\mu\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^\alpha+\frac{\nu}{2}\right)^{\left(\mu+\frac{\nu}{2}\right)}}\gamma^{-1}\left(\sqrt{\frac{\gamma}{\hat{\gamma}}}\right)^{\alpha\mu}\exp(-c\gamma)\,d\gamma$$
$$= \frac{\alpha\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty \gamma^{\frac{\alpha\mu}{2}-1}\left(\mu\gamma^{\frac{\alpha}{2}}\hat{\gamma}^{-\frac{\alpha}{2}}+\frac{\nu}{2}\right)^{-\left(\mu+\frac{\nu}{2}\right)}\exp(-c\gamma)\,d\gamma$$

Consider the transformation $\gamma^{\frac{\alpha}{2}} = z$ thus $\gamma = z^{\frac{2}{\alpha}}$ and, $\frac{d\gamma}{dz} = \frac{2}{\alpha}z^{\frac{2}{\alpha}-1}$.

$$L(c) = \frac{\alpha\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}+1}\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty z^{\mu-\frac{2}{\alpha}}\left(\mu z\hat{\gamma}^{-\frac{\alpha}{2}}+\frac{\nu}{2}\right)^{-\left(\mu+\frac{\nu}{2}\right)}\exp\left(-cz^{\frac{2}{\alpha}}\right)\frac{2}{\alpha}z^{\frac{2}{\alpha}-1}dz$$
$$= \frac{\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty z^{\mu-1}\left(\mu z\hat{\gamma}^{-\frac{\alpha}{2}}+\frac{\nu}{2}\right)^{-\left(\mu+\frac{\nu}{2}\right)}\exp\left(-cz^{\frac{2}{\alpha}}\right)dz.$$

Substituting in the approximation (18):

$$L(c) \approx \frac{\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}\hat{\gamma}^{\frac{\alpha\mu}{2}}}\int_0^\infty z^{\mu-1}\left(\mu z\hat{\gamma}^{-\frac{\alpha}{2}}+\frac{\nu}{2}\right)^{-\left(\mu+\frac{\nu}{2}\right)}\sum_{i=1}^4 a_i\exp(-b_icz)\,dz$$
$$= \frac{\mu^\mu\Gamma\left(\mu+\frac{\nu}{2}\right)\nu^{\frac{\nu}{2}}}{\Gamma(\mu)\Gamma\left(\frac{\nu}{2}\right)2^{\frac{\nu}{2}}\hat{\gamma}^{\frac{\alpha\mu}{2}}}\sum_{i=1}^4 a_i\int_0^\infty z^{\mu-1}\left(\mu z\hat{\gamma}^{-\frac{\alpha}{2}}+\frac{\nu}{2}\right)^{-\left(\mu+\frac{\nu}{2}\right)}\exp(-b_icz)\,dz. \quad (32)$$

Fitted values of the a_i's and b_i's for equations (19) and (32) for a specific case are given in Table 3.

	a_1	a_2	a_3	a_4	b_1	b_2	b_3	b_4
Normal	−0.181812	0.790359	−0.274838	1.463507	1.563427	1.647754	1.694966	1.741016
t	1.147886	1.018561	0.674093	−0.725686	2.018094	1.900272	1.327419	1.313918

Table 3: Fitted values for a_i and b_i when $\alpha = 1$, $\mu = 3$, $\hat{\gamma} = 2$, $c = 0.1$ and in t case $\nu = 3$.

Comparison of Laplace transforms

Specific values are evaluated for the Laplace transform for the normal- and t cases using the derived expressions, see Table 4.

Normal				
Equation	(14)	(15)	(16)	(19)
$L(0.1)$	0.797194	0.812525	0.797194	1.42148
t				
Equation	(29)	(30)	(31)	(32)
$L(0.1)$	0.624743	$1.87485 * 10^{21}$	0.624743	1.37588

Table 4: Values of Laplace transforms based on Table 3's results.

Infinite sums were truncated to 16 for computational ease, when larger sums are considered the value $L(0.1)$ for equation (15) diverges similar to what is seen with (30). This together with (30)'s outcome (see Table 4) indicates that the series expansions for $L(c)$ does not yield satisfactory results, with the approximation of the exponential function proposed by [16] also not performing well.

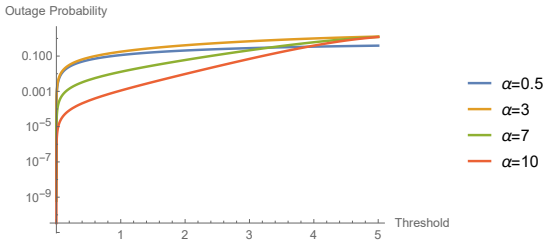
5 Performance analysis

The performance of the $\alpha - \mu$ type and distribution will be compared by considering their outage probabilities and ABER.

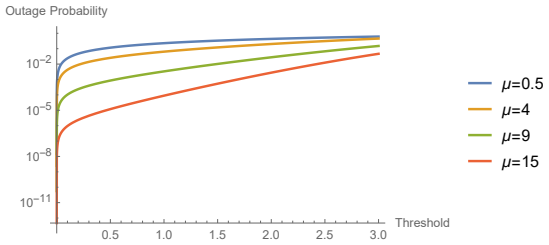
5.1 Outage Probability

The outage of the two special cases, the substitution of the normal and t distribution weighting functions, of the $\alpha - \mu$ type model will be evaluated by first considering the effect that the parameter values will have on each of the outages and then by direct comparison.

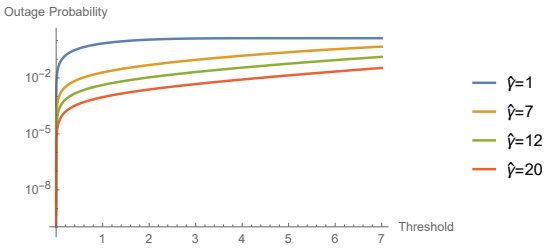
The effect of the parameters on the outage probability of the $\alpha - \mu$ distribution, which is also the $\alpha - \mu$ type model with the dirac delta function as weighting, can be seen in Figures 9a, 9b and 9c. . From the figures it is clear that a smaller α becomes preferable as the threshold value increases, while a larger number of multipath clusters, μ, will result in a lower outage probability. Similarly larger values of $\hat{\gamma}$ lead to a lower probability of an outage occurring.



(a) (13) with $\mu = 2$, $\hat{\gamma} = 1$ and α varying.



(b) (13) with $\alpha = 3$, $\hat{\gamma} = 1$ and μ varying.

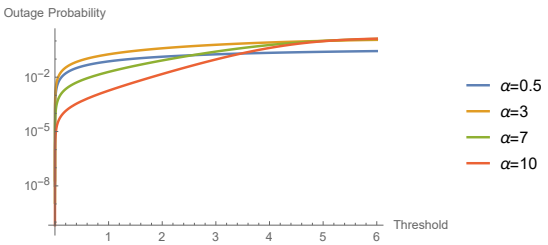


(c) (13) with $\alpha = 3$, $\mu = 2$ and $\hat{\gamma}$ varying.

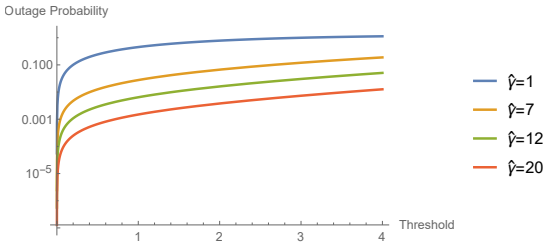
Figure 9: Outage probabilities of the $\alpha - \mu$ type model with normal weighting.

Similarly the effect of a change in parameters on the $\alpha - \mu$ type with the t distribution case is considered in Figures 10a through 10d. Two plots are included in cases where the outage probability's behaviour changes as the threshold increases. The behaviour of the outage probability under the $\alpha - \mu$ type is similar when parameters α and $\hat{\gamma}$ are being considered, the difference comes when considering the multipath clusters. When the threshold values are small a large μ performs better but as the threshold increases a small number of multipath clusters becomes preferred. The last parameter to consider is ν, which is the degrees of freedom of the t distribution, the outage probability is lower when a small degree

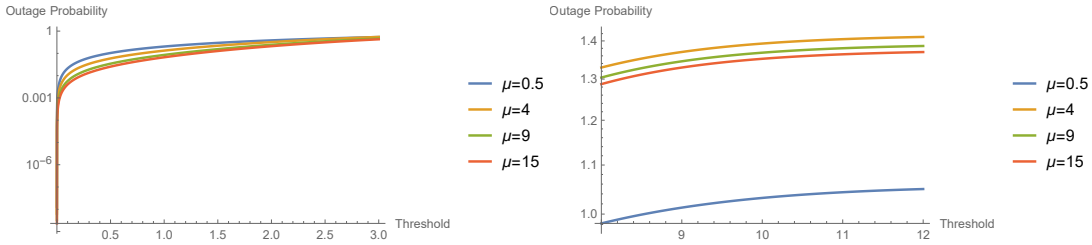
of freedom is considered. Thus the heavier the tails of the t distribution the lower the outage probability.



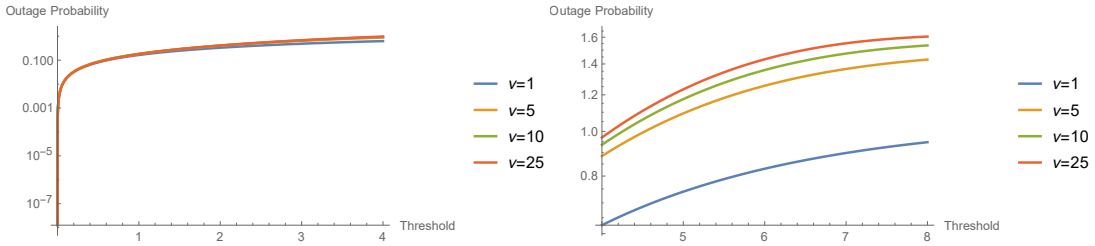
(a) (28) with $\mu = 1$, $\hat{\gamma} = 3$, $\nu = 3$ and α varying.



(b) (28) with $\alpha = 2$, $\mu = 1$, $\nu = 3$ and $\hat{\gamma}$ varying.



(c) (28) with $\alpha = 2$, $\hat{\gamma} = 3$, $\nu = 3$ and μ varying.



(d) (28) with $\alpha = 2$, $\mu = 1$, $\hat{\gamma} = 3$ and ν varying.

Figure 10: Outage probabilities of the $\alpha - \mu$ type model with t weighting.

Comparing the outage probability of the $\alpha - \mu$ type with the normal and t distribution weighting on the same graph as is done in Figure 11, it can be seen that the t distribution's weighting function results in a lower outage probability than the $\alpha - \mu$ distribution even for large degrees of freedom. As expected the $\alpha - \mu$ type approaches the $\alpha - \mu$ distribution as the degrees of freedom of the t distribution increases, since a larger degree of freedom results in the tails being lighter and closer to what is seen with the normal distribution.

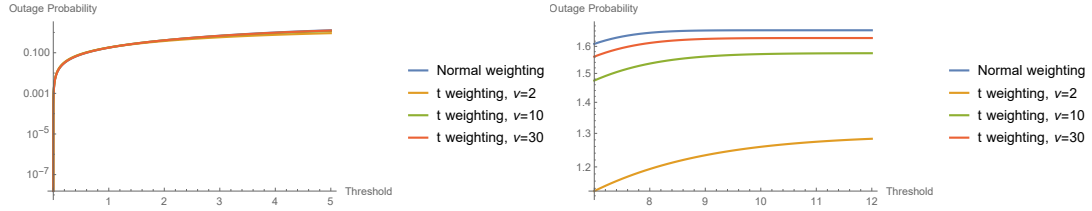


Figure 11: (13) and (28) with $\alpha = 2$, $\mu = 1$, $\hat{\gamma} = 3$ and ν varying.

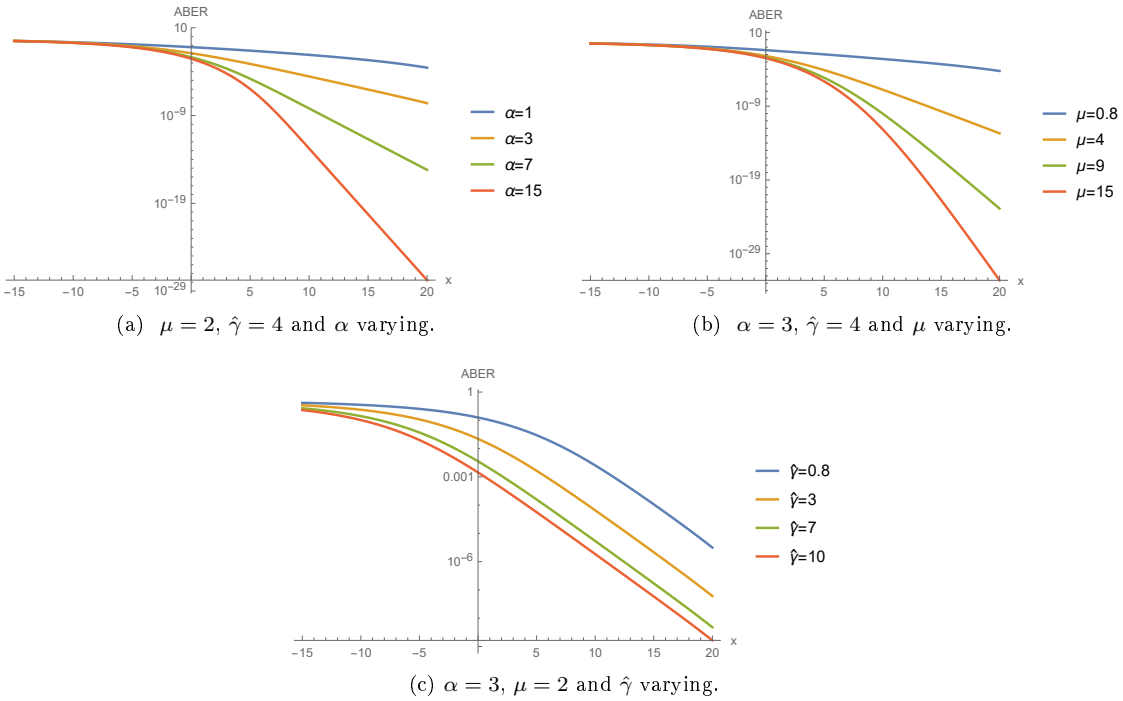
Remark 4. Figures 9a-11 are plotted on log scale resulting in the outage probability, which statistically is a cumulative distribution function (CDF), being greater than 1 for large threshold values. It can however

be shown that the power's CDFs does integrate to 1.

5.2 ABER

Similar to what was done for the outage probability, the special cases will first be considered separately to determine the effect of the parameters after which there will be a direct comparison. These comparisons are made using (1) with coherent detection of BPSK as derived by Ermolova [6] (see Section 2.4) and the integral expressions of the Laplace transforms (14) and (29) respectively.

Figures 12a, 12b and 12c illustrate the parameter's effects on the ABER in the case of the weighting function being the dirac delta function, while the t distribution's weighting is considered in Figures 13a to 13d. A lower ABER will be preferred and so in the case of the $\alpha-\mu$ distribution a large μ,$\hat{\gamma}$ will result in a better ABER.



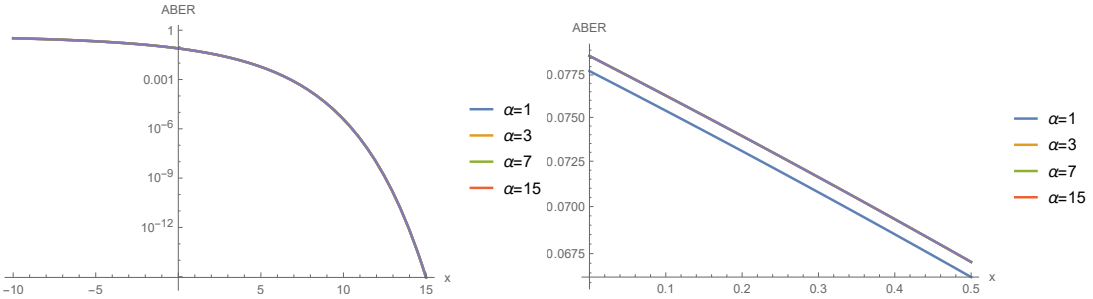
(a) $\mu = 2$, $\hat{\gamma} = 4$ and α varying.

(b) $\alpha = 3$, $\hat{\gamma} = 4$ and μ varying.

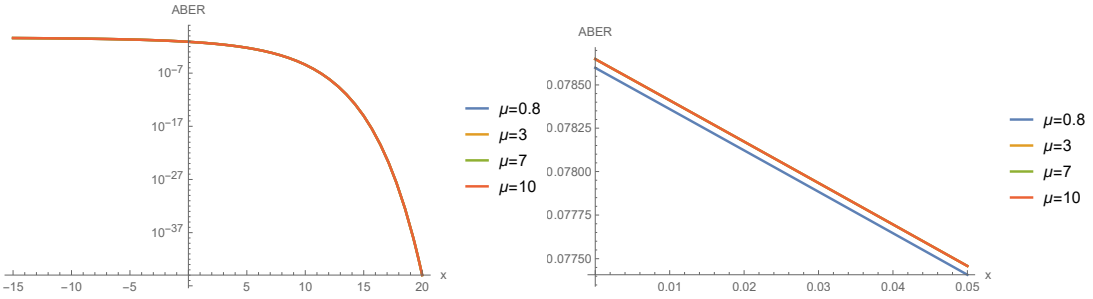
(c) $\alpha = 3$, $\mu = 2$ and $\hat{\gamma}$ varying.

Figure 12: ABER of $\alpha-\mu$ type model with normal weighting.

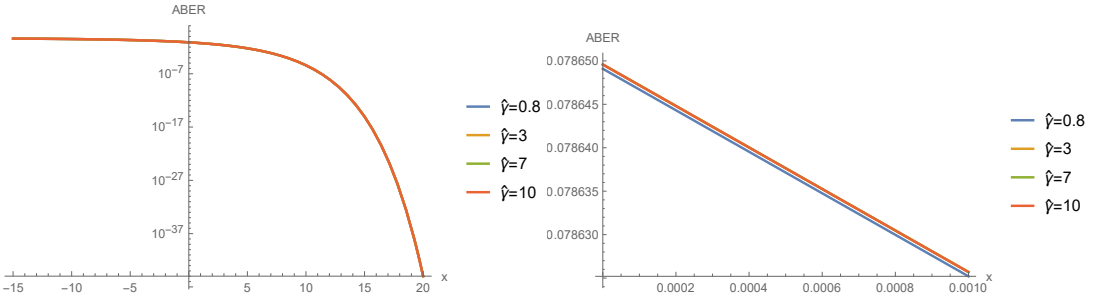
The figures for the $\alpha-\mu$ type model are not as revealing as those from the $\alpha-\mu$ distribution seeing as the ABER is very similar for all the parameter variations that were plotted here. Thus it is unclear which parameter values perform better when large ranges are considered for the ABER. Due to this a second plot over a small range, effectively magnifying the plot, is considered here and it can be seen that a smaller parameter value performs better regardless of the parameter being considered, see Figure 13.



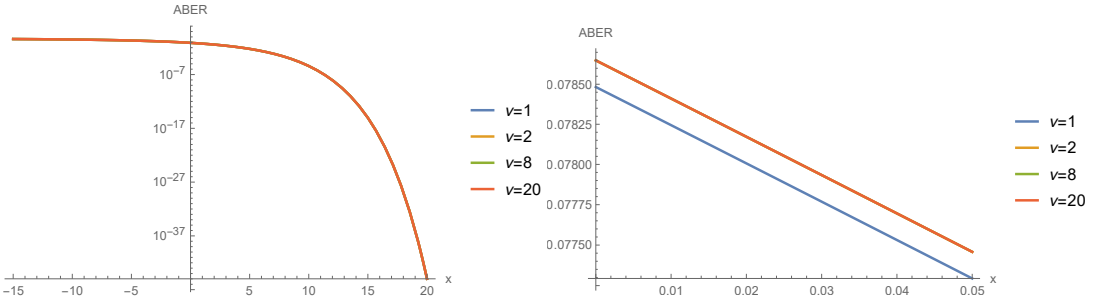
(a) $\mu = 2$, $\hat{\gamma} = 4$, $\nu = 3$ and α varying.



(b) $\alpha = 3$, $\hat{\gamma} = 4$, $\nu = 3$ and μ varying.



(c) $\alpha = 3$, $\mu = 2$, $\nu = 3$ and $\hat{\gamma}$ varying.



(d) $\alpha = 3$, $\mu = 2$,$\hat{\gamma} = 4$ and ν varying.

Figure 13: ABER of $\alpha - \mu$ type model with t weighting.

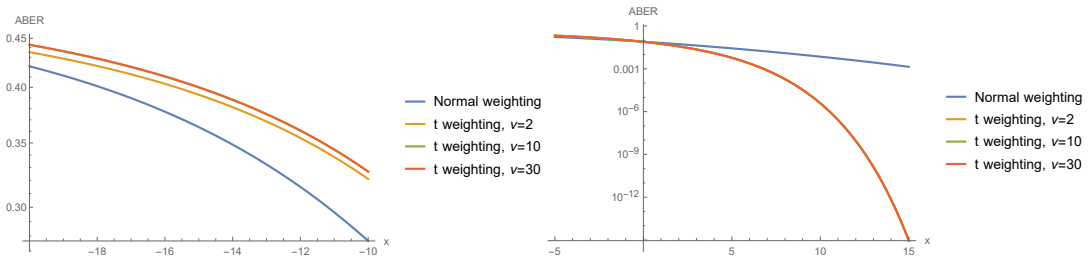


Figure 14: ABER comparison of $\alpha - \mu$ type with normal- and t weighting.

Comparing the ABER of the $\alpha - \mu$ type model with the normal and t distribution weighting on the same graph as is done in Figure 14, it becomes clear that there are cases when the $\alpha - \mu$ distribution will be preferred. When considering small values of x, which is the c in the Laplace transforms, the normal weighting outperforms the $\alpha - \mu$ type while large values result in the performance being reversed making the underlying t more effective. As x becomes larger the difference in the performance becomes more prominent.

6 Conclusion

6.1 Summary

Based on the performance evaluation for the two cases of the elliptical assumption, it is clear that there are many cases in which the ABER and outage probability of the $\alpha - \mu$ distribution can be improved through the use of the newly derived $\alpha - \mu$ type with underlying t distribution.

6.2 Future work

The poor performance of the exponential approximation as proposed in [16] could be investigated and an optimal fitting method found. This would be especially valuable for the $\alpha - \mu$ type model with underlying t distribution when considering the ABER since the Meijer's G function approach (31) will require a large number of input parameters when α becomes large.

References

[1] L. Amoroso. Ricerche intorno alla curva dei redditi. *Annali di Matematica Pura ed Applicata*, 2(1):123–159, 1925.

[2] M. Arashi, A. Bekker, M. T. Loots, and J. J. J. Roux. Integral representation of quaternion elliptical density and its applications. *Communications in Statistics - Theory and Methods*, 44(4):778–789, 2015.

[3] L. J. Bain and M. Engelhardt. *Introduction to Probability and Mathematical Statistics*. Brooks/Cole, 1987.

[4] K. C. Chu. Estimation and decision for linear systems with elliptical random processes. *IEEE Transactions on Automatic Control*, 18(5):499–505, Oct 1973.

[5] A. J. Coulson, A. G. Williamson, and R. G. Vaughan. A statistical basis for lognormal shadowing effects in multipath fading channels. *IEEE Transactions on Communications*, 46(4):494–502, Apr 1998.

[6] N. Y. Ermolova. Moment generating functions of the generalized $\eta - \mu$ and $\kappa - \mu$ distributions and their applications to performance evaluations of communication systems. *IEEE Communications Letters*, 12(7):502–504, July 2008.

[7] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products*. Academic press, 2014.

[8] E. J. Leonardo and M. D. Yacoub. The product of two $\alpha - \mu$ variates and the composite $\alpha - \mu$ multipath-shadowing model. *IEEE Transactions on Vehicular Technology*, 64(6):2720–2725, June 2015.

[9] A. M. Magableh and M. M. Matalgah. Moment generating function of the generalized $\alpha - \mu$ distribution with applications. *IEEE Communications Letters*, 13(6):411–413, June 2009.

[10] A. M. Mathai. *A Handbook of Generalized Special Functions for Statistical and Physical Sciences*. Oxford University Press, USA, 1993.

[11] A. M. Mathai and R. K. Saxena. *Generalized Hypergeometric Functions with Applications in Statistics and Physical Sciences*, volume 348. Springer, 1973.

[12] E. Ollila, J. Eriksson, and V. Koivunen. Complex elliptically symmetric random variables - generation, characterization, and circularity tests. *IEEE Transactions on Signal Processing*, 59(1):58–69, Jan 2011.

[13] J. Owen and R. Rabinovitch. On the class of elliptical distributions and their applications to the theory of portfolio choice. *Journal of Finance*, 38(3):745 – 752, 1983.

[14] J. F. Paris. Advances in the statistical characterization of fading: from 2005 to present. *International Journal of Antennas and Propagation*, 2014, 2014.

[15] A. P. Prudnikov, Y. A. Brychkov, and O. I. Marichev. *Integrals and Series*, volume 3 of *More Special Functions*. Oxford University Press US, 1986.

[16] E. Salahat, A. Hakam, N. Ali, and A. Kulaib. Moment generating functions of generalized wireless fading channels and applications in wireless communication theory. 2015.

[17] P. M. Shankar. *Fading and Shadowing in Wireless Systems*. Springer Science & Business Media, 2011.

[18] M. K. Simon and M. S. Alouini. *Digital Communication over Fading Channels*, volume 95. John Wiley & Sons, 2005.

[19] E. W. Stacy. A generalization of the gamma distribution. *The Annals of Mathematical Statistics*, 33(3):1187–1192, 1962.

[20] M. D. Yacoub. The $\alpha-\mu$ distribution: a general fading distribution. In *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pages 629–633 vol.2, Sept 2002.

[21] M. D. Yacoub. The $\alpha-\mu$ distribution: a physical fading model for the Stacy distribution. *IEEE Transactions on Vehicular Technology*, 56(1):27–34, Jan 2007.

Appendix

Results

R1. Standard normal to χ^2 [3, page 271]

If $Z \sim N(0,1)$ then $Z^2 \sim \chi^2(1)$.

R2. Sum of χ^2 random variables [3, page 270]

If $C_i \sim \chi^2(\nu_i)$ are independent chi-squared random variables for $i = 1, \ldots, n$, then $\sum_{i=1}^{n} C_i^2 \sim \chi^2\left(\sum_{i=1}^{n} \nu_i\right)$

.

R3. Gamma Function [3, page 111]

The gamma function denoted by $\Gamma(\mu)$ for all $\mu > 0$ is given by $\Gamma(\mu) = \int_0^\infty t^{\mu-1} e^{-t} dt$.

R4. PDF of a gamma random variable [3, page 111] For a random variable $X \sim GAM(\theta, \kappa)$ where $\theta > 0$ and $\kappa > 0$, X has PDF :

$$f_X(x) = \frac{1}{\Gamma(\kappa)\theta^\kappa} x^{\kappa-1} \exp\left(-\frac{x}{\theta}\right) \quad x > 0,$$

where $\Gamma(\cdot)$ is the gamma function as defined above.

R5. Incomplete gamma function

The incomplete gamma function has two cases:

1. $\gamma(\cdot,\cdot)$ is the lower incomplete gamma function and is defined as $\gamma(\alpha, x) = \int_0^x \exp(-t)\, t^{\alpha-1} dt$ for $\alpha > 0$ [7, eq 8.350-1].
2. $\Gamma(\cdot,\cdot)$ is the upper incomplete gamma function and is defined as $\Gamma(\alpha, x) = \int_x^\infty \exp(-t)\, t^{\alpha-1} dt$ [7, eq 8.350-2].

R6. Integral results

1. $\int_0^u \frac{x^{\mu-1}}{(1+\beta x)^\nu} dx = \frac{u^\mu}{\mu} {}_2F_1(\nu, \mu; 1+\mu; -\beta u)$ when $\mu > 0$ and ${}_2F_1(\cdot)$ denotes the Gauss hypergeometric function [7, eq 3.194-1].
2. $\int_0^\infty x^{\mu-1}(p + qx^\nu)^{-(n+1)} dx = \frac{1}{\nu p^{n+1}} \left(\frac{p}{q}\right)^{\frac{\mu}{\nu}} \frac{\Gamma\left(\frac{\mu}{\nu}\right)\Gamma\left(n+1-\frac{\mu}{\nu}\right)}{\Gamma(n+1)}$ when $0 < \frac{\mu}{\nu} < n+1$, $p \neq 0$ and $q \neq 0$ [7, eq 3.241-4].

3. $\int_0^\infty x^m \exp(-\beta x^n)\, dx = \frac{\Gamma(\gamma)}{n\beta^\gamma}$ where $\gamma = \frac{m+1}{n}$ when the requirements $\beta > 0$, $m > 0$ and $n > 0$ are met [7, eq 3.326-2].

4. $\int_0^\infty x^{\nu-1} \exp(-\mu x)\, dx = \frac{\Gamma(\nu)}{\mu^\nu}$ when requirements $\mu > 0$ and $\nu > 0$ are met [7, eq 3.381-4].

5. $\int_0^u x^m \exp(-\beta x^n)\, dx = \frac{\gamma(\nu, \beta u^n)}{n\beta^\nu}$ where $\nu = \frac{m+1}{n}$ when the requirements $\nu > 0$, $\beta > 0$ and $n > 0$ are met and $u > 0$ [7, eq 3.381-8].

R7.Generalized hypergeometric series

${}_pF_q(\alpha_1, \alpha_2, \ldots, \alpha_p; \quad \beta_1, \beta_2, \ldots, \beta_q; z) = \sum_{k=0}^\infty \frac{(\alpha_1)_k (\alpha_2)_k \ldots (\alpha_p)_k}{(\beta_1)_k (\beta_2)_k \ldots (\beta_p)_k} \frac{z^k}{k!}$ where $(a)_k = \frac{\Gamma(a+k)}{\Gamma(a)}$ [7, eq 9.14-1].

R8. Meijer's G function

1. $(1-z)^{-a} = \frac{1}{\Gamma(a)} G_{1,1}^{1,1}\left(-z \left| \begin{matrix} 1-a \\ 0 \end{matrix} \right. \right)$ [11, p54].

2. $z^\alpha \exp(-z) = z^\alpha G_{0,1}^{1,0}\left(z \left| \begin{matrix} - \\ 0 \end{matrix} \right. \right)$ [10, p69].

Code

SAS:

```
/*Underlying Normal fitting parameters*/
proc iml; start Func_Diff(x) global (i);
alpha = 1;
mu = 3;
ghat = 2;
c = 0.1;
*Theoretical constant;
tconst = (alpha*(mu**mu))/(2*(ghat**(alpha*mu/2))*Gamma(mu));
dt = 0.001;
do g = 0 to 50 by dt;
* Calculating function values for theoretical MGF;
ITpoint=(g**(alpha*mu/2-1))*exp(-mu*(g**(alpha/2))*(ghat**(-alpha/2))+c*g);
/*Function values at all gamma values included in loop*/
IT = IT//ITpoint;
end;
sum = sum(IT)*dt;
```

```
*Riemann sum;
MGF = tconst*sum;
*Theoretical;
print MGF;
asum = x[1]*(mu*ghat##(-alpha/2)+x[5]*c)##(-mu)+
x[2]*(mu*ghat##(-alpha/2)+x[6]*c)##(-mu)+
x[3]*(mu*ghat##(-alpha/2)+x[7]*c)##(-mu)+
x[4]*(mu*ghat##(-alpha/2)+x[8]*c)##(-mu);
aMGF = ((mu##mu)/(ghat##(alpha*mu/2)))*asum;
print aMGF;
Diff = abs(MGF - aMGF);
print diff;
return(Diff);
finish Func_Diff;
start Cons(x) global(i);
*mu >0 and:;
con = J(4, 1,.);
c = 0.1;
*points of c where we are fitting;
con[1] = (mu*ghat**(-alpha/2)+x[5]*c);*[i]);
con[2] = (mu*ghat**(-alpha/2)+x[6]*c);*[i]);
con[3] = (mu*ghat**(-alpha/2)+x[7]*c);*[i]);
con[4] = (mu*ghat**(-alpha/2)+x[8]*c);*[i]);
return(con);
finish Cons;
x = J(1,8,1);
opt = {0,2};
do i = 1 to 3;
call nlpnms(rc, xres, "Func_Diff", x, opt,,,,,"Cons");
end;
quit;

/*Underlying t fit of parameters*/
proc iml;
```

```
start Min_func(x) global(j);
alpha = 1;
mu = 3;
ghat = 2;
c = 0.1;
nu = 3;
dt=0.001;
/*Step size for numerical integration*/
tconst = (alpha*(mu**mu)*(nu**(nu/2))*
Gamma(mu+nu/2))/((2**(nu/2+1))*
(ghat**(alpha*mu/2))*Gamma(mu)*Gamma(nu/2));
*theoretical constant;
do g = 0 to 50 by dt;
* Calculating function values for theoretical MGF;
ITpoint=((mu*(g**(alpha/2))*(ghat**(-alpha/2))+nu/2)**(-mu-nu/2))
*(g**(alpha*mu/2))*exp(-c*g);
/*Function values at all gamma values included in loop*/
IT = IT//ITpoint;
end;
sum = sum(IT)*dt;
*Riemann sum;
MGF = tconst*sum;
*Theoretical;
print MGF;
aconst = (1/Gamma(mu))*(1/Gamma(nu/2))*
(((mu**mu)*(nu**(nu/2))*Gamma(mu+nu/2))/((2**(nu/2))*
(ghat**(alpha*mu/2))));
*approximation constant;
asum =0;
do j = 1 to 4;
d=0.001;
/*Step size for numerical integration*/
do z = 0 to 50 by d;
Ipoint=(z**(mu-1))*
```

```
((mu*z*ghat**(-alpha/2)+nu/2)**(-mu-nu/2))
*exp(-c*z*x[j+4]);
/*Function values at all z values*/
I = I//Ipoint;
end;
Int=sum(I)*d;
/*Estimated integral value*/
asum = asum + Int*x[j];
end;
aMGF = asum*aconst;
print aMGF;
diff = abs(MGF-aMGF);
return (diff);
finish Min_func;
x=J(1,8,1);
*x1 to x4 will be a1 to a4 and x5 to x8 b1 to b4 respectively;
opt = {0,2};
call nlpnms(rc, xest, "Min_func",x,opt);
quit;
```

Mathematica:

```
(*Define the PDF of the alpha mu distribution*)
FunctionAMPDF[alpha_, mu_, rhat_] :=
( alpha*(mu^mu)*(r^(alpha*mu - 1)))/((rhat^(alpha*mu))*Gamma[mu])*
Exp[-mu*(r/rhat)^alpha]

(*Plotting the PDF of the alpha mu distribution with varying values \ of alpha.*)
Plot[{FunctionAMPDF[1, 3, 2], FunctionAMPDF[4, 3, 2], FunctionAMPDF[9, 3, 2],
FunctionAMPDF[12, 3, 2]},{r, 0, 3.5}, AxesLabel -> {"r", "\[Alpha]-\[Mu] PDF"},
PlotRange -> All, PlotStyle -> Thick, PlotLegends -> {"\[Alpha]=1", "\[Alpha]=4",
"\[Alpha]=9", "\[Alpha]=12" }, LabelStyle -> {FontSize -> 13}]

(*Plotting the PDF of the alpha mu distribution with varying values \ of mu.*)
Plot[{FunctionAMPDF[4, 0.75, 2], FunctionAMPDF[4, 1, 2], FunctionAMPDF[4, 4, 2],
```

```
FunctionAMPDF[4, 9, 2]}, {r, 0, 4}, AxesLabel -> {"r", "\[Alpha]-\[Mu] PDF"},
PlotRange -> All, PlotStyle -> Thick, PlotLegends -> {"\[Mu]=0.75", "\[Mu]=1",
"\[Mu]=4", "\[Mu]=9" }, LabelStyle -> {FontSize -> 13}]

(*Plotting the PDF of the alpha mu distribution with varying values \ of r_hat.*)
Plot[{FunctionAMPDF[4, 3, 1], FunctionAMPDF[4, 3, 3], FunctionAMPDF[4, 3, 6],
FunctionAMPDF[4, 3, 8]}, {r, 0, 12}, AxesLabel -> {"r", "\[Alpha]-\[Mu] PDF"},
PlotRange -> All, PlotStyle -> Thick, PlotLegends -> {"\!\(\*OverscriptBox[\(r\),
\(^\)]\)=1", "\!\(\*OverscriptBox[\(r\), \(^\)]\)=3", "\!\(\*OverscriptBox[\(r\), \(^\)]\)=6",
"\!\(\*OverscriptBox[\(r\), \(^\)]\)=8" }, LabelStyle -> {FontSize -> 13}]

(*Define the CDF of the alpha mu distribution*)
FunctionAMCDF[alpha_, mu_, rhat_] :=
alpha*((mu^mu)/(Gamma[mu]*rhat^(alpha*mu)))*
NIntegrate[Exp[-(y^alpha*mu)/rhat^alpha]*y^(alpha*mu - 1), {y, 0, r}]

(*Plotting the CDF of the alpha mu distribution with varying values \ of alpha.*)
Plot[{FunctionAMCDF[1, 3, 2], FunctionAMCDF[4, 3, 2], FunctionAMCDF[9, 3, 2],
FunctionAMCDF[12, 3, 2]}, {r, 0, 4}, AxesLabel -> {"r", "\[Alpha]-\[Mu] CDF"},
PlotRange -> All, PlotStyle -> Thick, PlotLegends -> {"\[Alpha]=1", "\[Alpha]=4",
"\[Alpha]=9", "\[Alpha]=12"}, LabelStyle -> {FontSize -> 13}]

(*Plotting the CDF of the alpha mu distribution with varying values \ of mu.*)
Plot[{FunctionAMCDF[4, 0.75, 2], FunctionAMCDF[4, 1, 2], FunctionAMCDF[4, 4, 2],
FunctionAMCDF[4, 9, 2]}, {r, 0, 3.5}, AxesLabel -> {"r", "\[Alpha]-\[Mu] CDF"},
PlotRange -> All, PlotStyle -> Thick, PlotLegends -> {"\[Mu]=0.75", "\[Mu]=1", "\[Mu]=4",
"\[Mu]=9" }, LabelStyle -> {FontSize -> 13}]

(*Plotting the CDF of the alpha mu distribution with varying values \ of r_hat.*)
Plot[{FunctionAMCDF[4, 3, 1], FunctionAMCDF[4, 3, 3], FunctionAMCDF[4, 3, 6],
FunctionAMCDF[4, 3, 8]}, {r, 0, 12}, AxesLabel -> {"r", "\[Alpha]-\[Mu] CDF"},
PlotRange -> All, PlotStyle -> Thick, PlotLegends -> {"\!\(\*OverscriptBox[\(r\),
\(^\)]\)=1", "\!\(\*OverscriptBox[\(r\), \(^\)]\)=3", "\!\
(\*OverscriptBox[\(r\), \(^\)]\)=6","\!\(\*OverscriptBox[\(r\), \(^\)]\)=8" },
```

```
LabelStyle -> {FontSize -> 13}]

(*Define the PDF of the alpha mu type with t distribution weighting*)
FunctionTPDF[alpha_, mu_, rhat_, nu_] :=
(alpha*(mu^mu)*(r^(alpha*mu - 1))*(nu^(nu/2))*Gamma[mu + nu/2])/
((rhat^(alpha*mu))*(2^(nu/ 2))*(mu*((r/rhat)^alpha) + nu/2)^(mu + nu/2)
*Gamma[mu]*Gamma[nu/2])

(*Plot the PDF with varying values of alpha*)
Plot[{FunctionTPDF[1, 3, 2, 3], FunctionTPDF[4, 3, 2, 3], FunctionTPDF[9, 3, 2, 3],
FunctionTPDF[12, 3, 2, 3]}, {r, 0, 5}, PlotRange -> All, PlotStyle -> Thick,
PlotLegends -> {"\[Alpha]=1", "\[Alpha]=4", "\[Alpha]=9", "\[Alpha]=12"},
AxesLabel -> {"r", "PDF"}]

(*Plot the PDF with varying values of mu*)
Plot[{FunctionTPDF[4, 0.75, 2, 3], FunctionTPDF[4, 1, 2, 3], FunctionTPDF[4, 4, 2, 3],
FunctionTPDF[4, 9, 2, 3]}, {r, 0, 5}, PlotRange -> All, PlotStyle -> Thick,
PlotLegends -> {"\[Mu]=0.75", "\[Mu]=1", "\[Mu]=4", "\[Mu]=9"},
AxesLabel -> {"r", "PDF"}]

(*Plot the PDF with varying values of r hat*)
Plot[{FunctionTPDF[4, 3, 1, 3], FunctionTPDF[4, 3, 3, 3], FunctionTPDF[4, 3, 6, 3],
FunctionTPDF[4, 3, 8, 3]}, {r, 0, 14}, PlotRange -> All, PlotStyle -> Thick,
PlotLegends -> {"\!\(\*OverscriptBox[\(r\), \(^\)]\)=1", "\!\
(\*OverscriptBox[\(r\), \(^\)]\)=3", "\!\(\*OverscriptBox[\(r\), \(^\)]\)=6",
"\!\(\*OverscriptBox[\(r\), \(^\)]\)=8"}, AxesLabel -> {"r", "PDF"}]

(*Plot the PDF with varying values of nu*)
Plot[{FunctionTPDF[4, 3, 2, 1], FunctionTPDF[4, 3, 2, 3], FunctionTPDF[4, 2, 2, 8],
FunctionTPDF[4, 3, 2, 11]}, {r, 0, 10}, PlotRange -> All, PlotStyle -> Thick,
PlotLegends -> {"\[Nu]=1", "\[Nu]=3", "\[Nu]=8", "\[Nu]=11"}, AxesLabel -> {"r", "PDF"}]

(*Define the alpha mu PDF (Normal weighting) again for comaprison*)
FunctionAMPDF[alpha_, mu_, rhat_] :=
```

```
( alpha*(mu^mu)*(r^(alpha*mu - 1)))/((rhat^(alpha*mu))*Gamma[mu])*
Exp[-mu*(r/rhat)^alpha]

(*Plot for comparison of PDF's with normal weighting and t \
distribution weighting and varying values of nu*)
Plot[{FunctionAMPDF[4, 3, 2], FunctionTPDF[4, 3, 2, 3], FunctionTPDF[4, 3, 2, 11],
FunctionTPDF[4, 3, 2, 25]}, {r, 0, 4}, PlotRange -> All, PlotStyle -> Thick,
PlotLegends -> {"Normal weighting", "t weighting, \[Nu]=3", "t weighting,
\[Nu]=11", "t weighting, \[Nu]=25"}, AxesLabel -> {"r", "PDF"}]

(*Define the CDF of the alpha mu type with t distribution weighting*)
FunctionTCDF[alpha_, mu_, rhat_, nu_] :=
((alpha*(mu^mu)*(nu^(nu/2)))*Gamma[mu + nu/2])/
( rhat^(alpha*mu)*2^(nu/2)*Gamma[mu]*Gamma[nu/2])*
NIntegrate[(y^(alpha*mu - 1))/((mu*(y/rhat)^alpha) + nu/2)^( mu + nu/2), {y, 0, r}]

(*Plot the CDF with varying values of alpha*)
Plot[{FunctionTCDF[1, 3, 2, 3], FunctionTCDF[4, 3, 2, 3], FunctionTCDF[9, 3, 2, 3],
FunctionTCDF[12, 3, 2, 3]}, {r, 0, 5}, PlotRange -> All, PlotStyle -> Thick,
PlotLegends -> {"\[Alpha]=1", "\[Alpha]=4", "\[Alpha]=9", "\[Alpha]=12"},
AxesLabel -> {"r", "CDF"}]

(*Plot the CDF with varying values of mu*)
Plot[{FunctionTCDF[4, 0.75, 2, 3], FunctionTCDF[4, 1, 2, 3], FunctionTCDF[4, 4, 2, 3],
FunctionTCDF[4, 9, 2, 3]}, {r, 0, 4}, PlotRange -> All, PlotStyle -> Thick,
PlotLegends -> {"\[Mu]=0.75", "\[Mu]=1", "\[Mu]=4", "\[Mu]=9"},
AxesLabel -> {"r", "CDF"}]

(*Plot the CDF with varying values of rhat*)
Plot[{FunctionTCDF[4, 3, 1, 3], FunctionTCDF[4, 3, 3, 3], FunctionTCDF[4, 3, 6, 3],
FunctionTCDF[4, 3, 8, 3]}, {r, 0, 14}, PlotRange -> All, PlotStyle -> Thick,
PlotLegends -> {"\!\(\*OverscriptBox[\(r\), \(^\)]\)=1", "\!\(\*OverscriptBox[\(r\),
\(^\)]\)=3", "\!\(\*OverscriptBox[\(r\), \(^\)]\)=6", "\!\(\*OverscriptBox[\(r\),
\(^\)]\)=8"}, AxesLabel -> {"r", "CDF"}]
```

```
(*Plot the CDF with varying values of nu*)
Plot[{FunctionTCDF[4, 3, 2, 1], FunctionTCDF[4, 3, 2, 3], FunctionTCDF[4, 3, 2, 8],
FunctionTCDF[4, 3, 2, 11]}, {r, 0, 5}, PlotRange -> All, PlotStyle -> Thick,
PlotLegends -> {"\[Nu]=1", "\[Nu]=3", "\[Nu]=8", "\[Nu]=11"},
AxesLabel -> {"r", "CDF"}]

(*Define the alpha mu CDF (Normal weighting) again for comaprison*)
FunctionAMCDF[alpha_, mu_, rhat_] :=
alpha*((mu^mu)/(Gamma[mu]*rhat^(alpha*mu)))*
NIntegrate[Exp[-(y^alpha*mu)/rhat^alpha]*y^(alpha*mu - 1), {y, 0, r}]

(*Plotting the CDF of the alpha mu distribution and the alpha mu type
\ with t distribution weighting function with varying values of nu.*)
Plot[{FunctionAMCDF[4, 3, 2], FunctionTCDF[4, 3, 2, 3], FunctionTCDF[4, 3, 2, 11],
FunctionTCDF[4, 3, 2, 25]}, {r, 0, 4.5}, PlotRange -> All, PlotStyle -> Thick,
PlotLegends -> {"Normal weighting", "t weighting, \[Nu]=3", "t weighting, \[Nu]=11",
"t weighting, \[Nu]=25"}, AxesLabel -> {"r", "CDF"}]

(*Define the outage probability of the alpha mu distribution*)
FunctionAMOut[alpha_, mu_, ghat_] :=
((alpha*(mu^mu))/(2*Gamma[mu]*(ghat^(alpha*mu/2))))*
NIntegrate[((10^(g/10))^(alpha*mu/2 - 1))* Exp[-mu*(((10^(g/10))/ghat)^(alpha/2))],
{g, 0, out}]

(*Plot the outage for varying alpha with other parameters constant*)
LogPlot[{FunctionAMOut[0.5, 2, 3], FunctionAMOut[3, 2, 3], FunctionAMOut[7, 2, 3],
FunctionAMOut[10, 2, 3]}, {out, 0, 5}, PlotStyle -> Thick,
AxesLabel -> {"Threshold", "Outage Probability"}, PlotRange -> All,
PlotLegends -> {"\[Alpha]=0.5", "\[Alpha]=3", "\[Alpha]=7", "\[Alpha]=10"}]

(*Plot the outage for varying mu with other parameters constant*)
LogPlot[{FunctionAMOut[3, 0.5, 3], FunctionAMOut[3, 4, 3], FunctionAMOut[3, 9, 3],
FunctionAMOut[3, 15, 3]}, {out, 0, 3}, PlotStyle -> Thick,
```

```
AxesLabel -> {"Threshold", "Outage Probability"}, PlotRange -> All,
PlotLegends -> {"\[Mu]=0.5", "\[Mu]=4", "\[Mu]=9", "\[Mu]=15"}]

(*Plot the outage for varying gamma hat with other parameters \ constant*)
LogPlot[{FunctionAMOut[3, 2, 1], FunctionAMOut[3, 2, 7], FunctionAMOut[3, 2, 12],
FunctionAMOut[3, 2, 20]}, {out, 0, 7}, PlotStyle -> Thick,
AxesLabel -> {"Threshold", "Outage Probability"}, PlotRange -> All,
PlotLegends -> {"\!\(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=1", "\!\
(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=7", "\!\(\*OverscriptBox[\(\[Gamma]\),
\(^\)]\)=12", "\!\(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=20"}]

(*Define the outage probability of the alpha mu type with the t
\ distribution weighting function*)
FunctionAMTOut[alpha_, mu_, ghat_, nu_] :=
(alpha*(mu^mu)*Gamma[mu + nu/2]*(nu^(nu/2)))/
(Gamma[mu]* Gamma[nu/2]*(2^(nu/2 + 1))*(ghat^(alpha*mu/2)))*
NIntegrate[(10^(g/ 10))^(alpha*mu/2 - 1)/((mu*((Sqrt[10^(g/10)/ghat])^alpha)
+ nu/2)^(mu + nu/2)), {g, 0, out}]

(*Plot the outage for alpha varying and all other parameters constant*)
LogPlot[{FunctionAMTOut[0.5, 2, 3, 3], FunctionAMTOut[3, 2, 3, 3],
FunctionAMTOut[7, 2, 3, 3], FunctionAMTOut[10, 2, 3, 3]}, {out, 0, 6}, PlotStyle -> Thick,
PlotRange -> All, AxesLabel -> {"Threshold", "Outage Probability"},
PlotLegends -> {"\[Alpha]=0.5", "\[Alpha]=3", "\[Alpha]=7", "\[Alpha]=10"}]

(*Plot the outage for mu varying and all other parameters constant*)
LogPlot[{FunctionAMTOut[3, 0.5, 3, 3], FunctionAMTOut[3, 4, 3, 3],
FunctionAMTOut[3, 9, 3, 3], FunctionAMTOut[3, 15, 3, 3]}, {out, 0, 3}, PlotStyle -> Thick,
PlotRange -> All, AxesLabel -> {"Threshold", "Outage Probability"},
PlotLegends -> {"\[Mu]=0.5", "\[Mu]=4", "\[Mu]=9", "\[Mu]=15"}]
LogPlot[{FunctionAMTOut[3, 0.5, 3, 3], FunctionAMTOut[3, 4, 3, 3],
FunctionAMTOut[3, 9, 3, 3], FunctionAMTOut[3, 15, 3, 3]}, {out, 8, 12}, PlotStyle -> Thick,
PlotRange -> All, AxesLabel -> {"Threshold", "Outage Probability"},
PlotLegends -> {"\[Mu]=0.5", "\[Mu]=4", "\[Mu]=9", "\[Mu]=15"}]
```

```
(*Plot the outage for gamma hat varying and all other parameters \ constant*)
LogPlot[{FunctionAMTOut[3, 2, 1, 3], FunctionAMTOut[3, 2, 7, 3], FunctionAMTOut[3, 2, 12, 3],
FunctionAMTOut[3, 2, 20, 3]}, {out, -12, 4}, PlotStyle -> Thick,
PlotRange -> All, AxesLabel -> {"Threshold", "Outage Probability"},
PlotLegends -> {"\!\(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=1", "\!\
(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=7", "\!\(\*OverscriptBox
[\(\[Gamma]\), \(^\)]\)=12","\!\(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=20"}]

(*Plot the outage for nu varying and all other parameters constant*)
LogPlot[{FunctionAMTOut[3, 2, 3, 1], FunctionAMTOut[3, 2, 3, 5], FunctionAMTOut[3, 2, 3, 10],
FunctionAMTOut[3, 2, 3, 25]}, {out, 0, 4}, PlotStyle -> Thick,
PlotRange -> All, AxesLabel -> {"Threshold", "Outage Probability"},
PlotLegends -> {"\[Nu]=1", "\[Nu]=5", "\[Nu]=10", "\[Nu]=25"}]
LogPlot[{FunctionAMTOut[3, 2, 3, 1], FunctionAMTOut[3, 2, 3, 5], FunctionAMTOut[3, 2, 3, 10],
FunctionAMTOut[3, 2, 3, 25]}, {out, 4, 8}, PlotStyle -> Thick,
PlotRange -> All, AxesLabel -> {"Threshold", "Outage Probability"},
PlotLegends -> {"\[Nu]=1", "\[Nu]=5", "\[Nu]=10", "\[Nu]=25"}]

(*Plot the normal and t outage on one set of axes with varying nu \ values*)
LogPlot[{FunctionAMOut[3, 2, 3], FunctionAMTOut[3, 2, 3, 2], FunctionAMTOut[3, 2, 3, 10],
FunctionAMTOut[3, 2, 3, 30]}, {out, 0, 5}, PlotStyle -> Thick,
AxesLabel -> {"Threshold", "Outage Probability"}, PlotRange -> All,
PlotLegends -> {"Normal weighting", "t weighting, \[Nu]=2", "t weighting,
\[Nu]=10", "t weighting, \[Nu]=30"}]
LogPlot[{FunctionAMOut[3, 2, 3], FunctionAMTOut[3, 2, 3, 2], FunctionAMTOut[3, 2, 3, 10],
FunctionAMTOut[3, 2, 3, 30]}, {out, 7, 12}, PlotStyle -> Thick,
AxesLabel -> {"Threshold", "Outage Probability"}, PlotRange -> All,
PlotLegends -> {"Normal weighting", "t weighting, \[Nu]=2", "t weighting,
\[Nu]=10", "t weighting, \[Nu]=30"}]

(*Define the Laplace transform of alpha mu distribution in 3 parts,
\ part one is the constant, part two the inner integral and the last
\ part is the combination of the previous two with the definition of the ABER*)
```

```
FunctionConst[a_, m_, ghat_] :=
(a*m^m)/(2*ghat^((a*m)/2)*Gamma[m])
FunctionA[a_, m_, ghat_] :=
NIntegrate[ g^((a*m)/2 - 1)*
Exp[-10^((x/10))/(Sin[theta])^2*g - m*g^(a/2)*ghat^(-a/2)], {g, 0.01, 100}]
FunctionAB[a_, m_, ghat_] :=
1/Pi*FunctionConst[a, m, ghat]* NIntegrate[FunctionA[a, m, ghat], {theta, 0.01, Pi/2}]

(*Plotting the alpha mu distribtion with parameter alpha varying*)
LogPlot[{FunctionAB[1, 2, 4], FunctionAB[3, 2, 4], FunctionAB[7, 2, 4],
FunctionAB[15, 2, 4]}, {x, -15, 20}, PlotStyle -> Thick,
PlotRange -> All, PlotLegends -> {"\[Alpha]=1", "\[Alpha]=3", "\[Alpha]=7",
"\[Alpha]=15"}, AxesLabel -> {"x", "ABER"}]

(*Plotting the alpha mu distribtion with parameter mu varying*)
LogPlot[{FunctionAB[3, 0.8, 4], FunctionAB[3, 4, 4], FunctionAB[3, 9, 4],
FunctionAB[3, 15, 4]}, {x, -15, 20}, PlotStyle -> Thick, PlotRange -> All,
PlotLegends -> {"\[Mu]=0.8", "\[Mu]=4", "\[Mu]=9", "\[Mu]=15"},
AxesLabel -> {"x", "ABER"}]

(*Plotting the alpha mu distribtion with parameter gamma hat varying*)
LogPlot[{FunctionAB[3, 2, 0.8], FunctionAB[3, 2, 3], FunctionAB[3, 2, 7], FunctionAB[3, 2, 10]},
{x, -15, 20}, PlotStyle -> Thick, PlotRange -> All,
PlotLegends -> {"\!\(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=0.8", "\!\
(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=3", "\!\(\*OverscriptBox[\(\[Gamma]\),
\(^\)]\)=7", "\!\(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=10"}, AxesLabel -> {"x", "ABER"}]

(*Define the ABER of the alpha-mu type in 3 parts, the constant term, \ the inner integral,
and the outer integral over theta*)
FunctionConstant[a_, m_, ghat_, v_] :=
(a*m^m*v^(v/2)*Gamma[m + v/2])/( 2^(v/2 + 1)*ghat^((a*m)/2)*Gamma[m]*Gamma[v/2])
FunctionInnerInt[a_, m_, ghat_, v_] :=
NIntegrate[(m*g^(a/2)*ghat^(-a/2) + v/2)^-(m + v/2)*g^((a*m)/2 - 1)*
Exp[-10^((x/10))/(Sin[theta])^2], {g, 0.01, 10000}]
```

```
FunctionOuterInt[a_, m_, ghat_, v_] :=
FunctionConstant[a, m, ghat, v]*1/Pi*
NIntegrate[FunctionInnerInt[a, m, ghat, v], {theta, 0.001, Pi/2}]

(*Plot the ABER for alpha varying and other parameters constant*)
LogPlot[{FunctionOuterInt[1, 2, 4, 3], FunctionOuterInt[3, 2, 4, 3],
FunctionOuterInt[7, 2, 4, 3], FunctionOuterInt[15, 2, 4, 3]},
{x, -10, 15}, PlotStyle -> Thick, PlotRange -> All, AxesLabel ->
{"x", "ABER"}, PlotLegends -> {"\[Alpha]=1", "\[Alpha]=3", "\[Alpha]=7",
"\[Alpha]=15"}]
LogPlot[{FunctionOuterInt[1, 2, 4, 3], FunctionOuterInt[3, 2, 4, 3],
FunctionOuterInt[7, 2, 4, 3], FunctionOuterInt[15, 2, 4, 3]}, {x, 0, 0.5},
PlotStyle -> Thick, PlotRange -> All, AxesLabel -> {"x", "ABER"},
PlotLegends -> {"\[Alpha]=1", "\[Alpha]=3", "\[Alpha]=7", "\[Alpha]=15"}]

(*Plot the ABER for mu varying and other parameters constant*)
LogPlot[{FunctionOuterInt[3, 0.8, 4, 3], FunctionOuterInt[3, 4, 4, 3],
FunctionOuterInt[3, 9, 4, 3], FunctionOuterInt[3, 15, 4, 3]}, {x, -15, 20},
PlotStyle -> Thick, PlotRange -> All, AxesLabel -> {"x", "ABER"},
PlotLegends -> {"\[Mu]=0.8", "\[Mu]=3", "\[Mu]=7", "\[Mu]=10"}]
LogPlot[{FunctionOuterInt[3, 0.8, 4, 3], FunctionOuterInt[3, 4, 4, 3],
FunctionOuterInt[3, 9, 4, 3], FunctionOuterInt[3, 15, 4, 3]}, {x, 0, 0.05},
PlotStyle -> Thick, PlotRange -> All, AxesLabel -> {"x", "ABER"},
PlotLegends -> {"\[Mu]=0.8", "\[Mu]=3", "\[Mu]=7", "\[Mu]=10"}]

(*Plot the ABER for gamma hat varying and other parameters constant*)
LogPlot[{FunctionOuterInt[3, 2, 0.8, 3], FunctionOuterInt[3, 2, 3, 3],
FunctionOuterInt[3, 2, 7, 3], FunctionOuterInt[3, 2, 10, 3]}, {x, -15, 20},
PlotStyle -> Thick, PlotRange -> All, AxesLabel -> {"x", "ABER"},
PlotLegends -> {"\!\(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=0.8",
"\!\(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=3", "\!\(\*OverscriptBox[\(\[Gamma]\),
\(^\)]\)=7", "\!\(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=10"}]
LogPlot[{FunctionOuterInt[3, 2, 0.8, 3], FunctionOuterInt[3, 2, 3, 3],
FunctionOuterInt[3, 2, 7, 3], FunctionOuterInt[3, 2, 10, 3]}, {x, 0, 0.001},
```

```
PlotStyle -> Thick, PlotRange -> All, AxesLabel -> {"x", "ABER"},
PlotLegends -> {"\!\(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=0.8",
"\!\(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=3", "\!\(\*OverscriptBox[\(\[Gamma]\),
\(^\)]\)=7", "\!\(\*OverscriptBox[\(\[Gamma]\), \(^\)]\)=10"}]

(*Plot the ABER for nu varying and other parameters constant*)
LogPlot[{FunctionOuterInt[3, 2, 4, 1], FunctionOuterInt[3, 2, 4, 2],
FunctionOuterInt[3, 2, 4, 8], FunctionOuterInt[3, 2, 4, 20]}, {x, -15, 20},
PlotStyle -> Thick, PlotRange -> All, AxesLabel -> {"x", "ABER"},
PlotLegends -> {"\[Nu]=1", "\[Nu]=2", "\[Nu]=8", "\[Nu]=20"}]
LogPlot[{FunctionOuterInt[3, 2, 4, 1], FunctionOuterInt[3, 2, 4, 2],
FunctionOuterInt[3, 2, 4, 8], FunctionOuterInt[3, 2, 4, 20]}, {x, 0, 0.05},
PlotStyle -> Thick, PlotRange -> All, AxesLabel -> {"x", "ABER"},
PlotLegends -> {"\[Nu]=1", "\[Nu]=2", "\[Nu]=8", "\[Nu]=20"}]

(*Comparison plot between alpha mu type with normal and gamma \
weighting functions with varying nu values*)
LogPlot[{FunctionAB[1, 3, 2], FunctionOuterInt[1, 3, 2, 2],
FunctionOuterInt[1, 3, 2, 10], FunctionOuterInt[1, 3, 2, 30]}, {x, -20, -10},
PlotStyle -> Thick, AxesLabel -> {"x", "ABER"}, PlotRange -> All,
PlotLegends -> {"Normal weighting", "t weighting, \[Nu]=2", "t weighting,
\[Nu]=10", "t weighting, \[Nu]=30"}]
LogPlot[{FunctionAB[1, 3, 2], FunctionOuterInt[1, 3, 2, 2],
FunctionOuterInt[1, 3, 2, 10], FunctionOuterInt[1, 3, 2, 30]}, {x, -5, 15},
PlotStyle -> Thick, AxesLabel -> {"x", "ABER"}, PlotRange -> All,
PlotLegends -> {"Normal weighting", "t weighting, \[Nu]=2", "t weighting,
\[Nu]=10", "t weighting, \[Nu]=30"}]

(*Give Parameter values underlying normal*)
a = 1
m = 3
gh = 2
c = 0.1
v = 3
```

```
a1 = -0.181812

a2 = 0.790359

a3 = -0.274838

a4 = 1.463507

b1 = 1.563427

b2 = 1.647754

b3 = 1.694966

b4 = 1.741016


(*Laplace for theoretical*)

(a*m^m)/(2*Gamma[m]*gh^((a*m)/2))* NIntegrate[ g^((a*m)/2 - 1)*Exp[-c*g]

*Exp[-m*g^(a/2)*gh^(-(a/2))], {g, 0, 2000}]


(*Sum*)

1/Gamma[m]*\!\( \*UnderoverscriptBox[\(\[Sum]\), \(k = 0\), \(16\)]\

( \*FractionBox[ SuperscriptBox[\((\(-c\))\), \(k\)], \(Factorial[k]\)]*

\*FractionBox[\(Gamma[m +  \*FractionBox[\(2*k\), \(a\)]]\),

SuperscriptBox[\(m\),  FractionBox[\(2*k\), \(a\)]]]

\*SuperscriptBox[\(gh\), \(k\)]\)\)


(*Expression with G*)

(a*m^m)/(2*Gamma[m]*gh^((a*m)/2)) NIntegrate[ g^((a*m)/2 - 1)*

MeijerG[{{}, {}}, {{0}, {}}, m/gh^(a/2) g^(a/2)]*

MeijerG[{{}, {}}, {{0}, {}}, c*g],{g, 0, 2000}]


(*Expression with approximation*)

(m^m*gh^(-((a*m)/2)))*(a1*(m*gh^(-a/2) + b1*c)^-m +

a2*(m*gh^(-a/2) + b2*c)^-m +

a3*(m*gh^(-a/2) + b3*c)^-m +

a4*(m*gh^(-a/2) + b4*c)^-m)


(*Set parameter values underlying t*)

a11 = 1.147886

a12 = 1.018561
```

```
a13 = 0.674093

a14 = -0.725686

b11 = 2.018094

b12 = 1.900272

b13 = 1.327419

b14 = 1.313918


(*Integral*)

(a*m^m*Gamma[m + v/2]*v^(v/2))/( Gamma[m]*Gamma[v/2]*2^(v/2 + 1)*gh^((a*m)/2))*

NIntegrate[(m*g^(a/2)*gh^(-a/2) + v/2)^-(m + v/2)*g^((a*m)/2 - 1)* Exp[-c*g], {g, 0, 2000}]


(*Sum*)

(a*m^m*2^(m - 1))/(Gamma[m]*Gamma[v/2]*gh^((a*m)/2)*v^m)*(\!\( \*UnderoverscriptBox[\(\[Sum]\),

\(k = 0\), \(16\)]\( \*FractionBox[ SuperscriptBox[\((\(-1\))\), \(k\)],  SuperscriptBox[\(c\),

 FractionBox[\(a*\((m + k)\)\), \(2\)]]]* \*FractionBox[\(Gamma[m +  \*FractionBox[\(v\), \(2\)]

+ k]*Gamma[ \*FractionBox[\(a*\((m + k)\)\), \(2\)]]\), \(Factorial[k]\)]*

\*SuperscriptBox[\(( \*FractionBox[\(2*m\), \( \*SuperscriptBox[\(gh\),

FractionBox[\(a\), \(2\)]]*v\)]\), \(k\)]\)\))


(*Expression with G*)

(a^(1/2*(a*m + 1))*m^m*2^(2*m + v/2 - 3/2 - a/2))/( Gamma[m]*Gamma[v/2]*v^m*(c*gh)^((a*m)/2))

*Pi^(-1/2 (a + 1))* MeijerG[{{(1 - m - v/2)/2, (2 - m - v/2)/2, (1 - (a*m)/2)/ a}, {}},

{{0, 1/2}, {}}, ((2*m)/(gh^(a/2)*v))^2*(a/c)^a]


(*Expression with approximation*)

(m^m*Gamma[m + v/2]*v^(v/2))/( Gamma[m]*Gamma[v/2]*2^(v/2)*gh^((a*m)/ 2))*

(a11* NIntegrate[ z^(m - 1)*(m*z*gh^(-a/2) + v/2)^-(m + v/2)*Exp[-b11*c*z], {z, 0, 2000}] +

a12*NIntegrate[ z^(m - 1)*(m*z*gh^(-a/2) + v/2)^-(m + v/2)*Exp[-b12*c*z], {z, 0, 2000}]

+ a13*NIntegrate[ z^(m - 1)*(m*z*gh^(-a/2) + v/2)^-(m + v/2)*Exp[-b13*c*z], {z, 0, 2000}]

+ a14*NIntegrate[ z^(m - 1)*(m*z*gh^(-a/2) + v/2)^-(m + v/2)*Exp[-b14*c*z], {z, 0, 2000}])
```

Modeling ordered categorical data

Belinda Lemmer 14008808

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr. N. Strydom

Department of Statistics, University of Pretoria

30 October 2017

Abstract

Models for ordinal categorical data will be investigated. Loglinear models and logit models for two-way tables are considered, specically. It is important to investigate how the models are adjusted to take account of the orderings. Scores are assigned to quantify the order of the ordinal variable. These models for ordinal data has the advantage of fewer parameters which makes the interpretation and calculations easier. An ordinal approach will detect a positive association between variables X and Y, more successfully.

Declaration

I, *Belinda,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____
Belinda Lemmer

_____
Dr. N. Strydom

_____
Date

Acknowledgments

1. Thanks to the Lord, for my talents.
2. Thanks to my family for their love and support.
3. Thanks to Dr. N. Strydom for her time, advice and guidance with this research report.
4. I acknowledge the financial support from the Center for AI Reseacrh, Meraka Institute, CSIR.
5. Thanks to the Department of Statistics of the University of Pretoria for the opportunity.

Contents

List of Figures

List of Tables

1 Introduction

If data is measured by sets of categories, it is referred to as a categorical variable. A categorical variable with no natural ordering is a nominal variable, such as gender, race or nationality. Ordinal variables are classified as categorical variables with ordered levels [1]. The possible values of an ordinal variable has a natural ordering, but the distances between the values are undefined [12]. An example of an ordinal variable is social class with ordered levels of "lower", "middle" and "upper".

In this research report, models for ordinal categorical data are investigated. Loglinear models and logit models are specifically considered. These models for categorical data do not always take into account the orderings of ordinal categorical data. When the orderings of ordinal variables are considered, it results in different and in some cases more accurate results, which may be easier to interpret. Therefore it is important to investigate how the models are adjusted to take account of the orderings [1].

Scores are being used to quantify the order of the ordinal variable. The most common scores used are integer scores where, for example, r levels are scored $i = 1, 2, ...r$. It is noticed that the distance between the levels of the ordinal variable, are assumed then to be of unit length.

Ordinal data are of particular interest since ordinal data models have fewer parameters to estimate. Stronger inferences and better tests for goodness of fit can then be made [1]. When an ordinal approach is used, it will detect a positive association between variables X and Y, more successfully. If ordered categories are inspected, the odds for adjacent response categories and how these odds depend on one or more of the explanatory variables can be quantified more accurately [6].

Karl Pearson and Yule wrote many articles about the association between categorical variables, which are described by the loglinear model [1, 3]. The response (dependent) and explanatory (independent) variables are not differentiated in loglinear models. Loglinear models also describe the cell expected frequencies through interaction parameters. The loglinear models can be useful where the response categories are ordered and the explanatory variable's are ordered or unordered [6]. If the explanatory variable are not ordinal, the parameterization of the additive effects, as in the two-way ANOVA, can be used. If the explanatory variable is ordinal, a set of parameters that take the ordering of the variables into consideration will be used [6].

Haberman [8] and Goodman [5] developed more complex loglinear models for situations where there is some association and at least one of the variables are ordinal.

If there are too many cell frequencies equal to zero, it may be impossible to fit certain models. A structural zero is when a zero value occurs in a cell where it is theoretically possible to have no observations. A sampling zero is when a zero value occurs in a cell for which the expected frequency is greater than zero but the observed value is equal to zero due to the sample size being small. Sampling zeros give rise to difficulties which can be eliminated by adding a small constant in each cell of the table [1].

Logit models describe the effect of the explanatory variables on the response variables. Logit models such as the cumulative logit model (proportional odds model), as well as the adjacent-categories logit model and the continuation ratio logit model will be discussed in this report [2]. Logit models are special cases of loglinear models, since both models contain the same structure for association between the response and explanatory variables.

In the case of ordinal data, the loglinear and logit models can be categorized into the so called uniform association model and the row effects model which will be discussed further in Section 3.2 and 3.3. The uniform association model is considered when an ordinal-ordinal table is used. The row effects model fixes the column scores and treat the row scores as the parameters [2]. The uniform association model and the row effects model are special cases of the linear-by-linear association model [2]. For example, a uniform association model in terms of local odds is a linear-by-linear association model with equally spaced scores.

In Section 2, an overview is given of the basic theory which underlies loglinear and logit models. Concepts are illustrated using a practical application. The theory of the loglinear and logit models for ordinal data are then discussed and are also illustrated with the same application, in Section 3.

Marginal formulas for X	Marginal formulas for Y
$n_{i+} = \sum_j n_{ij}$	$n_{+j} = \sum_i n_{ij}$
$p_{i+} = \sum_j p_{ij} = \frac{n_{i+}}{n}$	$p_{+j} = \sum_i p_{ij} = \frac{n_{+j}}{n}$

Table 1: Marginal formulas

2 Background Theory

2.1 Cross-classification tables

A cross-classification table (contingency table) gives count of the number of observations with a certain combination of characteristics [1]. The term contingency were formulated by Karl Pearson as any measure of the total deviation from independent probabilities [11]. Yule then viewed the categories as fixed and considered the relationship structure between or among variables by using various functions of the cross-product ratio, which is discussed in more detail in Section 2.2 [13]. Bartlett used Yule's cross-product ratio to define and develop a test for the presence of an interaction term in a 2x2x2 table [4].

Suppose that X is the row variable with r categories and Y is the column variable with c categories. Let n_{ij} be the number of observations in the cell located in the i^{th} row and j^{th} column of the table. Then the total sample size will be defined as $n = \sum_i \sum_j n_{ij}$. Let $p_{ij} = \frac{n_{ij}}{n}$ be the probability of an observation being in the i^{th} row and j^{th} column of the table. The probabilities in the joint distribution for a sample, will be denoted as $\{p_{ij}\}$ where $\sum_i \sum_j p_{ij} = 1$.

$$\sum_i p_{i+} = \sum_j p_{+j} = 1$$

The estimated expected frequency is then defined as $\widehat{m}_{ij} = \frac{n_{+j} n_{i+}}{n}$.

The population joint probabilities is defined as $\{\pi_{ij}\}$. The joint distribution function is $F_{ij} = \sum_{a \leq i} \sum_{b \leq j} (\pi_{ab})$ where $i = 1, ...r$ and $j = 1, ...c$ and gives the probability that a response is classified into the first i rows and the first j columns.

The conditional distribution is a collection of response proportions at a certain level of the explanatory variable. Given row i, the proportion of observations in the j^{th} category of Y is the sample conditional distribution denoted as $\{p_{j(i)}\}$. The population conditional probabilities is defined as $\{\pi_{j(i)}\}$. The conditional distribution function is $F_{j(i)} = \sum_{b \leq j} (\pi_{b(i)})$, thus it yields the probability that a response is classified into one of first j columns given that it is classified into row i.

$p_{j(i)} = \frac{n_{ij}}{n_{i+}}$ $\qquad\qquad$ $\sum_j p_{j(i)} = 1$

The marginal distribution function of X is $F_i^X = F_{ic} = \sum_{a \leq i} (\pi_{a+})$ and the marginal distribution function of Y is $F_j^Y = F_{rj} = \sum_{b \leq j} (\pi_{+b})$.

General notation for a two way contingency table, are illustrated in the next example.

Dumping Severity Example

In Table 2, notation is illustrated for a contingency table with ordinal variables. The data of four different operations of treating duodenal ulcer patients are given. The operations corresponds to the removal of various amounts of the stomach. Operation A represents drainage and vagotomy, while operation B is 25% resection and vagotomy. A 50% resection and vagotomy is operation C and operation D is a 75% resection. The levels of operations are naturally ordered, with A being the least severe operation. The extent of undesirable side effects of the operation is described by dumping severity and also have a natural ordering form none to moderate. The dumping severity example will be used to illustrate all the concepts researched.

	Dumping Severity			
Operation	None	Slight	Moderate	Total
A	$n_{11} = 61$	$n_{12} = 28$	$n_{13} = 7$	$n_{1+} = 96$
B	$n_{21} = 68$	$n_{22} = 23$	$n_{23} = 13$	$n_{2+} = 104$
C	$n_{31} = 58$	$n_{32} = 40$	$n_{33} = 12$	$n_{3+} = 110$
D	$n_{41} = 53$	$n_{42} = 38$	$n_{43} = 16$	$n_{4+} = 107$
Total	$n_{+1} = 240$	$n_{+2} = 129$	$n_{+3} = 48$	$n = 417$

Table 2: Frequency of Dumping Severity Data

	Dumping Severity			
Operation	None	Slight	Moderate	Total
Probability	$p_{11} = 0,1463$	$p_{12} = 0,0671$	$p_{13} = 0,0168$	$p_{1+} = 0,2302$
Conditional Probability	$p_{1(1)} = 0,6354$	$p_{2(1)} = 0,2917$	$p_{3(1)} = 0,0729$	
Probability	$p_{21} = 0,1631$	$p_{22} = 0,0552$	$p_{23} = 0,0312$	$p_{2+} = 0,2494$
Conditional Probability	$p_{1(2)} = 0,6538$	$p_{2(2)} = 0,2212$	$p_{3(2)} = 0,1250$	
Probability	$p_{31} = 0,1391$	$p_{32} = 0,0959$	$p_{33} = 0,0288$	$p_{3+} = 0,2638$
Conditional Probability	$p_{1(3)} = 0,5273$	$p_{2(3)} = 0,3636$	$p_{3(3)} = 0,1091$	
Probability	$p_{41} = 0,1271$	$p_{42} = 0,0911$	$p_{43} = 0,0384$	$p_{4+} = 0,2566$
Conditional Probability	$p_{1(4)} = 0,4953$	$p_{2(4)} = 0,3551$	$p_{3(4)} = 0,1495$	
Total	$p_{+1} = 0,5755$	$p_{+2} = 0,3094$	$p_{+3} = 0,1151$	1

Table 3: Probability Table of Dumping Severity Data

In Table 3 the probabilities (p_{ij}) and the conditional probabilities $\left(p_{(i)j}\right)$ are given for the data.

2.2 Odds ratios for 2 x 2 tables

Odds ratios are a measure used to describe the degree of association in a 2x2 table. The odds that the response is in column 1 instead of in column 2, for row i are defined as $\Omega_i = \frac{\pi_{i2}}{\pi_{i1}}$ with each $\Omega_i > 0$. If $\Omega_i > 1$ it implies that the odds of the response being in column 1 is less likely than the odds of being in column 2. The odds ratio (cross product ratio) is then defined as

$$\theta = \frac{\Omega_2}{\Omega_1} = \frac{\pi_{11}\pi_{22}}{\pi_{12}\pi_{21}}$$

The log odds ratio is $\log(\theta) = \log(\pi_{11}) - \log(\pi_{12}) - \log(\pi_{21}) + \log(\pi_{22})$. The log odds ratio is symmetric about 0 such that if rows and columns are switched, it would result in a change of sign. Hence a degree of association can be represented by values of $\log(\theta)$ that have the same absolute value but differ due to their sign [1].

Variables are independent if and only if $\Omega_1 = \Omega_2$ which implies that $\theta = 1$. If $\theta > 1$ it implies that $\pi_{2(2)} > \pi_{2(1)}$, which means that individuals in row 2 are more likely to make a second response than individuals in row 1. If $0 \leq \theta < 1$ it implies that $\pi_{2(2)} < \pi_{2(1)}$.

For sample frequencies $\{n_{ij}\}$, an estimate of θ is $\widehat{\theta} = \frac{n_{11}n_{22}}{n_{12}n_{21}}$ which is equal to 0 or ∞ if any $n_{ij} = 0$. This is not a desirable estimator and therefore Gart and Zweifull [9] showed that $\widetilde{\theta} = \frac{(n_{11}+0,5)(n_{22}+0,5)}{(n_{12}+0,5)(n_{21}+0,5)}$ is a better estimator since it has smaller bias and a smaller mean square error.

Under standard random sampling, $\widehat{\theta}$ and $\log(\widehat{\theta})$ are asymptotically $(n \to \infty)$ normally distributed around population values. The $\widehat{\theta}$ distribution is highly skewed for small n. An important note is that $\log(\widehat{\theta})$ converges quicker to its asymptotic distribution than $\widehat{\theta}$ does [1]. The calculation of odds ratios are illustrated with the next example.

	None to Moderate	Slight to Moderate
A	8,71	4
B	5,23	1,77
C	4,83	3,33
D	3,31	2,38
Total	5	2,69

Table 4: Odds Ratios for Dumping Severity Variable

	None	Slight	Moderate	Total
A to D	1,15	0,74	0,41	0,90
B to D	1,28	0,61	0,81	0,97
C to D	1,09	1,05	0,75	1,02

Table 5: Odds Ratios for Operation Variable

Dumping Severity Example

In Section 3.1, the adjacent odds ratios are calculated for the dumping severity example. For now, odds ratios with respect to a specified reference category is considered. Firstly, the odds ratios for the dumping severity variable is calculated with the reference level 'moderate'. The columns of Table 4 represents the odds of dumping being none or slight to moderate. For example, the odds of dumping being none for operation B is 5,23 times higher than the odds of dumping being moderate for operation B. The odds of dumping being none for all the operations is 5 times higher than the odds of dumping being moderate for all the operations.

Odds ratios for the operation variable is calculated with the reference level operation D. The rows of Table 5 represents the odds of operation A,B and C to operation D. For example, the odds of operation D for no dumping is 1,15 times higher than the odds of operation A for no dumping. The odds of operation D for all types of dumping is 1,02 times higher for operation C for all types of dumping.

2.3 Measures of goodness of fit

2.3.1 Chi-square statistic

Pearson developed his χ^2 test for comparing the expected frequencies, under the hypothesis of independence, and the observed frequencies [10]. The null hypothesis state that the expected frequencies do not differ significantly from the observed frequencies. The test statistic used is calculated as

$$\chi^2 = \sum_i \sum_j \frac{(n_{ij} - \widehat{m}_{ij})^2}{\widehat{m}_{ij}}$$

and has a χ^2 distribution with $(r-1)(c-1)$ degrees of freedom. The cell χ^2 value which is calculated as $\frac{(n_{ij}-\widehat{m}_{ij})^2}{\widehat{m}_{ij}}$ represent the difference between the estimated frequency and the observed frequency in each cell of the table. If the cell χ^2 value exceeds the value of 3,84, it indicates a significant difference between the observed and the expected frequency on a 5% level of significance. Hence if the null hypothesis is rejected to conclude that the expected frequency and the observed frequency do differ significantly, the cell χ^2 values can be used to see in exactly which cell of the table, the significant difference occur.

2.3.2 Likelihood Ratio Test

To test the hypothesis of independence, the Pearson statistic or the likelihood ratio statistic are being used. The likelihood ratio statistic is defined as

$$G^2 = 2\sum_i \sum_j n_{ij} log(\frac{n_{ij}}{\widehat{m}_{ij}})$$

Under the alternative hypothesis the constraint $\sum_i \sum_j (\pi_{ij}) = 1$ yields $rc - 1$ linearly independent parameters.

2.4 Measures of Association

Goodman and Kruskal [7] formed confidence intervals for the association measures by obtaining approximate standard errors. Goodman and Kruskal [7] also showed that most measures of association have an asymptotic normal distribution for multinomial sampling. These statistics will therefore detect associations, as $n \to \infty$, where the true value of the measures are not zero. Measure of association estimates or parameter estimates of the model, can be used to test hypotheses of independence, conditional independence or higher-order interactions [1].

2.4.1 Conditional independence test

To test the statistical significance of the association between X and Y, the hypothesis considered is $H_0 : \beta = 0$ or $H_0 : \tau_1 = ... = \tau_r = 0$, based on the specified model[1]. Hence the test is the conditional independence test under the assumption that the specified model (M) holds. The test statistic that will be considered is $G^2(I|M) = G^2(I) - G^2(M)$ with $df = 1$. It is very important to note that an ordinal test based on $G^2(I|M)$ is asymptotically more powerful for detecting changes from the independence model, than the test based on $G^2(I)$, when the specified model holds and for small degrees of freedom. The statistic $G^2(I)$ is however better than $G^2(I|M)$ when detecting non-monotonic dependencies for which β is zero or close to zero.

2.5 Loglinear models

A saturated model is one with no degrees of freedom and therefore fits the data perfectly. The number of parameters for the model is the same as the number of cells in the table. A saturated model has the form

$$\log(m_{ij}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_{ij}^{XY} \tag{1}$$

where m_{ij} is the expected frequencies, μ is the overall effect, λ_i^X and λ_j^Y are the effects of category i of variable X and the effects of category j of variable Y and $\lambda_{ij}^{XY}$is the interaction effect between category i of variable X and category j of variable Y [1].

The independence model has the form

$$\log(m_{ij}) = \mu + \lambda_i^X + \lambda_j^Y \tag{2}$$

If the variable X and Y are independent, it implies that the interaction effect is absent.

Dumping Severity Example

Let D denote dumping severity variable and O the operation variable. If the levels of D is denoted by i, then $i = N, S, M$ represent the levels "none", "slight" and "moderate". The levels of O is denoted by j therefore A, B, C and D represent the level of operation. The saturated model for the dumping severity data is given by:

$$\log(m_{ij}) = \mu + \lambda_i^D + \lambda_j^O + \lambda_{ij}^{DO} \tag{3}$$

Dumping Severity Estimates	Operation Estimates	Interaction Estimates		
$\lambda_1^D = 0,7633$	$\lambda_1^O = -0,1973$	$\lambda_{11}^{OD} = 0,2179$	$\lambda_{12}^{OD} = 0,0810$	$\lambda_{13}^{OD} = -0,2989$
$\lambda_2^D = 0,1216$	$\lambda_2^O = -0,0203$	$\lambda_{21}^{OD} = 0,1495$	$\lambda_{22}^{OD} = -0,2927$	$\lambda_{23}^{OD} = 0,1432$
$\lambda_3^D = -(\lambda_1^D + \lambda_2^D) = -0,8849$	$\lambda_3^O = 0,0844$	$\lambda_{31}^{OD} = -0,1143$	$\lambda_{32}^{OD} = 0,1231$	$\lambda_{33}^{OD} = -0,0416$
	$\lambda_4^O = -(\lambda_1^O + \lambda_2^O + \lambda_3^O) = 0,1332$	$\lambda_{41}^{OD} = -0,2531$	$\lambda_{42}^{OD} = 0,0886$	$\lambda_{43}^{OD} = 0,1973$

Table 6: Parameter Estimates for Saturated Loglinear Model Applied to Dumping Severity Data

Dumping Severity Estimates	Operation Estimates
$\lambda_1^D = 0,7434$	$\lambda_1^O = -0,0812$
$\lambda_2^D = 0,1226$	$\lambda_2^O = -0,00112$
$\lambda_3^D = -(\lambda_1^D + \lambda_2^D) = -0,866$	$\lambda_3^O = 0,055$
	$\lambda_4^O = -(\lambda_1^O + \lambda_2^O + \lambda_3^O) = 0,02732$

Table 7: Parameter Estimates for Independence Loglinear Model Applied to Dumping Severity Data

where λ_i^D and λ_j^O are the effects of category i of dumping severity and the effects of category j of the operations, $\lambda_{ij}^{DO}$is the interaction effect between dumping severity and the operations. The overall effect μ is equal to 3,3269. The estimated effect parameters are given in Table 6.

The independence model for the dumping severity data is given by:

$$\log(m_{ij}) = \mu + \lambda_i^D + \lambda_j^O \tag{4}$$

The overall effect μ for the independence model is equal to 3,35. The estimated effect parameters are given in Table 7.

The χ^2-statistic is 10,5419 based on 6 degrees of freedom. The p-value is 0,1036 which implies that the null hypothesis is not rejected. Therefore the conclusion reached, is that overall the estimated frequencies do not differ significantly from the observed frequencies. In Table 8, the cell chi-square values are given. It is observed that none of the values exceed the critical value of 3,84, hence the estimated frequency values do not significantly differ from the observed values, in each cell.

The likelihood ratio for testing the independence of the variables are $G^2 = 10,878$ based on 6 degrees of freedom. The p-value is calculated as 0,0922 which means that the dumping severity variable and the operation variable are independent at a 5% significance level.

	Dumping Severity		
Operation	None	Slight	Moderate
A	0,598	0,0971	1,68
B	1,108	2,6152	0,0884
C	0,4453	1,0478	0,0346
D	1,1962	0,7251	1,1016

Table 8: Cell Chi-Square Values for Dumping Severity Data

		Dumping Severity	
		None	Some
Operation	A	61	35
	B	68	36
	C	58	52
	D	53	54

Table 9: Dumping Severity Adjusted Frequency Table

Operation Estimates
$\tau_1^O = 0,235$
$\tau_2^O = 0,316$
$\tau_3^O = -0,211$
$\tau_4^O = -(\tau_1^O + \tau_2^O + \tau_3^O) = 0,02732$

Table 10: Parameter Estimates for Logit Model Applied to Adjusted Dumping Severity Data

2.6 Logit models

Suppose variables X and Y are categorical and that the Y variable is a dichotomous response variable. The conditional probability of a response j for variable Y, at level i of X, is $\pi_{j(i)} = \frac{\pi_{ij}}{\pi_{i+}}$.

The logit for Y at i is $\log\left[\frac{\pi_{2(i)}}{1-\pi_{2(i)}}\right] = \log\left(\frac{m_{i2}}{m_{i1}}\right)$, hence it is the log of the odds of level 1 of variable Y relative to level 2 of variable Y.

Logit models for the cross-classification can be fitted by using weighted least squares or by using maximum likelihood estimation if the explanatory variables are continuous.

Consider X as a nominal variable [1]. A model that includes the effect of X is

$$\log\left(\frac{m_{i2}}{m_{i1}}\right) \quad = \quad \alpha + \tau_i^X \tag{5}$$

where $\sum \tau_i^X = 0$.

Dumping Severity Example

To illustrate the logit model, the dumping severity data are adjusted so that dumping severity is a dichotomous variable. Collapsing the second and third column of Table 2, yields the adjusted data in Table 9. The logit model for the dumping severity data is given by:

$$\log(m_{ij}) = \alpha + \tau_i^O \tag{6}$$

The α for the model is equal to 0,3205. The estimated effect parameters are given in Table 10.

The χ^2-statistic is 7,8865 based on 3 degrees of freedom. The p-value is 0,0484 which implies that the null hypothesis is rejected. Therefore the conclusion reached, is that overall the estimated frequencies do differ significantly from the observed frequencies. In Table 11, the cell chi-square values are given. It is observed that none of the values exceed the critical value of 3,84, hence the estimated frequency values do not significantly differ from the observed values, in each cell.

The likelihood ratio for testing the independence of the variables are $G^2 = 7,9204$ based on 3 degrees of freedom. The p-value is calculated as 0,0477 which means that the dumping severity variable and the operation variable are dependent at a 5% significance level.

	Dumping Severity	
Operation	None	Some
A	0,598	0,8109
B	1,108	1,5024
C	0,4453	0,6037
D	1,1962	1,6219

Table 11: Cell Chi-Square Values for Dumping Severity Data

2.7 Estimation

Loglinear and logit models can be fitted by using weighted least squares or maximum likelihood methods [1]. In this report, only the maximum likelihood method is considered. The maximum likelihood estimates $\{\widehat{m}_{ij}\}$ are used to test the hypothesis that the population cell proportions satisfy an assumed model.

The maximum likelihood estimates $\{\widehat{m}_{ij}\}$ depend on the cell counts through sufficient statistics [3]. Suppose that variables X and Y are assumed to be independent Poisson random variables. Then the joint Poisson probability mass function of $\{n_{ij}\}$ is

$$\prod_i \prod_j \frac{\exp\left(-m_{ij}\right) m_{ij}^{n_{ij}}}{n_{ij}!}$$

where $\prod_i \prod_j$ gives the product over all the cells in the table. The log likelihood can then be expressed as $L\left(m\right) = \sum_i \sum_j n_{ij} \log\left(m_{ij}\right) - \sum_i \sum_j m_{ij}$. Now consider the loglinear model $\log(m_{ij}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_{ij}^{XY}$. Then the log likelihood becomes

$$L\left(m\right) = n\mu + \sum_i n_{i+}\lambda_i^X + \sum_j n_{+j}\lambda_j^Y + \sum_i \sum_j n_{ij}\lambda_{ij}^{XY} - \sum_i \sum_j \exp\left(\mu + \lambda_i^X + \lambda_j^Y + \lambda_{ij}^{XY}\right) \qquad (7)$$

It is known that the Poisson distribution is in the exponential family, and therefore the coefficients of parameters in equation (7) are sufficient statistics. This implies that $n_{i+}$and n_{+j} are sufficient statistics. The likelihood equations are then obtained by differentiating $L\left(m\right)$ with respect to a parameter and setting the result equal to zero [3]. The likelihood equations obtained for this example, will then be $\widehat{m}_{++} = n$, $\widehat{m}_{i+} = n_{i+}$, $\widehat{m}_{+j} = n_{+j}$ and $\widehat{m}_{ij} = n_{ij}$.

Birch showed that there is a unique solution $\{\widehat{m}_{ij}\}$ that satisfy the model and match the sample data in their minimal sufficient statistics. Therefore, if a solution is found, the solution must be the ML solution. The fitted values $\left\{\widehat{m}_{ij} = \frac{n_{i+}n_{+j}}{n}\right\}$ satisfies the model, as well as the likelihood equations. Hence it implies that they are the ML estimates.

For the two-way loglinear model, there are an explicit formula for $\widehat{m}_{ij}$ and the estimates are called direct. Unfortunately, many loglinear models do not have direct estimates and an iterative method, such as the Newton Rhapson approach, is required to get the ML estimates. These iterative methods can also be used for models with direct estimates, but are redundant [3].

3 Categorical Analysis of Ordinal Data

3.1 Odds ratios for ordinal variables

Literature [1] differentiates between basic sets of odds ratios namely: a local set, a local-global set and a global set.

A local set of $(r-1)(c-1)$ odds ratios, also called the adjacent odds ratios, is defined as

$$\theta_{ij} = \frac{\pi_{ij}\pi_{i+1,j+1}}{\pi_{i,j+1}\pi_{i+1,j+1}} \qquad (8)$$

Cross Products		
	None to Slight	Slight to Moderate
A to B	0,7369	2,2609
B to C	2,0390	0,5308
C to D	1,0396	1,4035

Table 12: Cross Product Ratios of Dumping Severity Data

This is a more natural method for cross-classifications of ordinal variables. Each θ_{ij} describe sample association in a certain part of the table. The local odds treat the row and column variables the same and is formed by using cells in adjacent rows and adjacent columns. Goodman suggested loglinear models to analyze $\{\theta_{ij}\}$ [5].

The **local-global set** of odds ratios does make a distinction between rows and column variables. The value of the local-global odds ratios indicate if rows are stochastically ordered on the response variable. The response (row) variable is regarded as local and the column variable as global, since all the categories of the column variable will be considered [1].

$$\theta_{ij}^{'} = \frac{(\sum_{b\leq j}\pi_{ib})(\sum_{b>j}\pi_{i+1,b})}{(\sum_{b>j}\pi_{ib})(\sum_{b\leq j}\pi_{i+1,b})} = \frac{F_{j(i)}(1-F_{j(i+1)})}{F_{j(i+1)}(1-F_{j(i)})} \tag{9}$$

For rows i and $i+1$, $\log(\theta_{ij}^{'}) \geq 0$ for $j = 1, ...(c-1)$, is equivalent to $F_{j(i)} \geq F_{j(i+1)}$ for $j = 1, ...c$. If the equivalence holds, the probability in row $i+1$ is greater for higher ordered variables. Hence the distribution in row $i+1$ is stochastically higher than the distribution in row i.

The **global set** is similar to regular odds ratios for 2x2 tables with $(r-1)(c-1)$ ways of collapsing the category levels of the variables into dichotomies. It treats the response and explanatory variables the same and describe the global association in the variables.

$$\theta_{ij}^{''} = \frac{(\sum_{a\leq i}\sum_{b\leq j}\pi_{ab})(\sum_{a>i}\sum_{b>j}\pi_{ab})}{(\sum_{a\leq i}\sum_{b>j}\pi_{ab})(\sum_{a>i}\sum_{b\leq j}\pi_{ab})} = \frac{F_{ij}(1+F_{ij}-F_{i+}-F_{+j})}{(F_{i+}-F_{ij})(F_{+j}-F_{ij})} \tag{10}$$

If all $\log(\theta_{ij}) \geq 0$ then all $\log(\theta_{ij}^{'}) \geq 0$ and if all $\log(\theta_{ij}^{'}) \geq 0$ then all $\log(\theta_{ij}^{''}) \geq 0$. The converse of these rules are not true.

Dumping Severity Example

For the given data, the cross product ratios (adjacent odds ratios) are calculated by equation (8) and tabulated in Table 12. The estimated odds that dumping is slight instead of none is 2,0390 times higher for operation C than for operation B.

In Table 13 the odds ratios are given where $\widehat{\theta}_{ij}^{'}$ is calculated using equation (9) and $\widehat{\theta}_{ij}^{''}$ is calculated using equation (10). From Table 13, it is observed that the estimated odds that dumping is moderate instead of none or slight is 1,82 times higher for operation B than for operation A. The estimated odds that dumping is moderate instead of none or slight is 1,86 times higher than for operation B, C, D than for operation A. Operation D is stochastically higher than operations A, B and C. Operation C is stochastically higher than operation A.

3.2 Loglinear models for ordinal variables

Standard loglinear models have the limitation that it treat variables as nominal which are invariant to the orderings of ordered categorical data. Hence loglinear models must be adjusted to be useful

	i \ j	N : S, M	M : N, S
$\hat{\theta}'_{ij}$	A : B	0,92	1,82
	B : C	1,69	0,86
	C : D	1,14	1,44
$\hat{\theta}''_{ij}$	A : B, C, D	1,38	1,86
	A, B : C, D	1,74	1,33
	A, B, C : D	1,55	1,53

Table 13: Values of Ordinal Odds Ratios for Dumping Severity Data.

when analyzing ordinal data. These models give a more structured form of association and interaction terms. The adjusted model association parameters describe certain types of trends and have greater power to detect important types of alternatives to the null hypothesis.

To obtain the maximum likelihood estimates of the parameters, iterative methods are needed [5]. Estimates of the loglinear models may therefore be obtained by using the iterative Newton-Rhapson method. When the Newton-Rhapson method is used, it produces a covariance matrix of the parameter estimates.

For large samples, the estimate of β has an asymptotically normal distribution.

3.2.1 Loglinear models for ordinal-ordinal tables

Suppose a two dimensional table with ordinal variables, X and Y. Variable X has r levels and variable Y has c levels. Scores $\{u_i\}$ and $\{v_j\}$ are assigned to the ordinal variables, where $u_1 < ... < u_r$ and $v_1 < ... < v_c$. The scores assigned are usually chosen as a measure of the distances between categories. If scores are equally spread then it will simplify the interpretation of the model. In practice, integer scores where scores assigned are $\{u_i = i\}$ and $\{v_j = j\}$ are most common.

The uniform association model is defined as:

$$\log(m_{ij}) \quad = \quad \mu + \lambda_i^X + \lambda_j^Y + \beta(u_i - \bar{u})(v_j - \bar{v}) \tag{11}$$

where $\sum \lambda_i^X = \sum \lambda_j^Y = 0$ and take the orderings into account explicitly through the scores assigned.

In equation (11) the β parameter describe the association between variables X and Y. Hence if $\beta = 0$ the model simplifies to the independence model (2). The association term reflects the deviation of the model (11) from the independence model (2).

The model given by (11) has degrees of freedom equal to $rc - r - c$. This model does not need to add association parameters as the number of categories increase [1].

A model in which all local odds ratios (8) are equal is a uniform association model. If integer scores $\{u_i = i\}$ and $\{v_j = j\}$ are used then it implies that all $\theta_{ij} = exp(\beta)$.

For a specific row i in the uniform association model (11), the deviation is a linear function of Y through scores $\{v_j\}$ with a slope of $\beta(u_i - \bar{u})$. Therefore it is known as a linear-by-linear association model.

The relative size of expected frequencies in rows a and b is measured by $\log(\frac{m_{bj}}{m_{aj}}) = (\lambda_b^X - \lambda_a^X) + \beta(u_b - u_a)(v_j - \bar{v})$. If $\beta > 0$ for any rows $a < b$, then there are relatively more large observations on Y in row b than in row a. Hence the conditional distribution of Y in row b is stochastically higher than conditional distribution of Y in row a [1]. For any pair of rows $a < b$ and columns $c < d$, $\log(\frac{m_{ac}m_{bd}}{m_{ad}m_{bc}}) = \beta(u_b - u_a)(v_d - v_c)$. This shows that the log odds ratio formed by a rectangular pattern of cells is proportional directly to the product of the distance between the rows and the distance between the columns. The log odds ratio equals β when the rows and the columns are one unit apart [1].

There is no closed form expression for the MLE of m_{ij} in the uniform association model (11). Under the sampling assumptions, the estimates satisfy the following likelihood equations:

$\widehat{m}_{i+} = n_{i+}$	$\widehat{m}_{+j} = n_{+j}$	$\sum_i \sum_j u_i v_j \widehat{m}_{ij} = \sum_i \sum_j u_i v_j n_{ij}$

Let $p_{ij} = \frac{n_{ij}}{n}$ and $\hat{\pi}_{ij} = \frac{\hat{m}_{ij}}{n}$ denote the estimates of the probability π_{ij} for the observed data. Then the third equation implies $\sum_i \sum_j u_i v_j \hat{\pi}_{ij} = \sum_i \sum_j u_i v_j p_{ij}$.

The conditional independence test under the assumption that the uniform association model holds, consider the test statistic $G^2(I|U) = G^2(I) - G^2(U)$ with $df = 1$.

3.2.2 Loglinear models for ordinal-nominal tables

Suppose that the categories of X are nominal and the categories of Y are ordinal. Then scores $v_1 < ... < v_c$ are used to reflect the orderings of the columns.

The row effects model is defined as a loglinear model that use the ordinal nature of Y through scores $\{v_j\}$ and has the form:

$$\log(m_{ij}) = \mu + \lambda_i^X + \lambda_j^Y + \tau_i(v_j - \bar{v}) \tag{12}$$

where $\sum \lambda_i^X = \sum \lambda_j^Y = 0$, $\{v_j\}$ are known constants and $\{\tau_i\}$ are the row effect parameters. The row effects model (12) has degrees of freedom $(r-1)(c-2)$. If all $\tau_i = 0$ it implies the independence model. The uniform association model is obtained when $\tau_i = \beta(u_i - \bar{u})$. The deviation from the independence model (2) is reflected by $\tau_i(v_j - \bar{v})$. The deviation of the row effects model (12) from the independence model in a specific row, is a linear function of Y with slope τ_i. Hence the row effects model is a linear-by-linear model. If $\tau_i > 0$ it implies that in row i the probability of classification above $\bar{v}$ on Y is higher than expected if the variables are independent. The row effect parameters are a measure used to compare rows in terms of how the responses on the ordinal variable tend to be distributed [1].

For fixed rows a and b, the relative size of expected frequencies is $\log(\frac{m_{bj}}{m_{aj}}) = (\lambda_b^X - \lambda_a^X) + (\tau_b - \tau_a)(v_j - \bar{v})$. If $\tau_b > \tau_a$ then the conditional distribution of Y at level b of X is stochastically higher than the conditional distribution of Y at level a of X.

For any pair of rows $a < b$ and columns $c < d$, the log odds ratio is defined as $\log(\frac{m_{ac}m_{bd}}{m_{ad}m_{bc}}) = (\tau_b - \tau_a)(v_d - v_c)$ for the row effects model. This log odds ratio is proportional to the distance between the columns. If $\tau_b > \tau_a$ the log odds ratio is positive. When integer scores $\{v_j = j\}$ are applied, the log odds ratio will be a constant [1].

The row effects model (12) can also be applied to the ordinal-ordinal tables in cases where the departure from the independence model is not linear across rows or when the study is mainly concerned with comparing levels of the row variable with respect to the conditional distribution on the column variable.

Suppose the column variable Y is nominal and that scores $\{u_i\}$ are assigned to the ordinal row variable X. Then the loglinear column effects model is defined as

$$\log(m_{ij}) = \mu + \lambda_i^X + \lambda_j^Y + \rho_j(u_i - \bar{u}) \tag{13}$$

where $\sum \lambda_i^X = \sum \lambda_j^Y = 0$, $\{u_i\}$ are known constants and $\{\rho_i\}$ are the column effect parameters.

There is no closed form expression for the MLE of m_{ij} in the row effects model (12). The estimates can be obtained by using the Newton-Rhapson method. Under the sampling assumptions, the estimates satisfy the following likelihood equations:

$\widehat{m}_{i+} = n_{i+}$	$\widehat{m}_{+j} = n_{+j}$	$\sum_i \sum_j v_j \widehat{m}_{ij} = \sum_i \sum_j v_j n_{ij}$

Equation 3 implies that the mean of the conditional distribution across the columns, are the same as when the conditional distribution is based on the observed data or the estimated expected frequency.

Parameter	Estimate
$\widehat{\mu}$	3,3319
$\hat{\lambda}_1^O$	-0,2010
$\hat{\lambda}_2^O$	-0,0330
$\hat{\lambda}_3^O$	0,0993
$\hat{\lambda}_4^O$	0,1347
$\hat{\lambda}_1^D$	0,7605
$\hat{\lambda}_2^D$	0,1332
$\hat{\lambda}_3^D$	-0,8937
$\widehat{\beta}$	0,0407

Table 14: Parameter estimates of uniform association model for dumping severity data

The conditional independence test hypothesis is $H_0 : \tau_1 = ... = \tau_r = 0$. The test is based on $G^2(I|R) = G^2(I) - G^2(R)$ with $df = 1$. The statistic $G^2(I|R)$ focus on where the ordinal scale is used through the linear departure of $\log m_{ij}$ from the independence model. The statistic $G^2(I)$ ignores the ordinal nature of Y, if the specified model is the row effects model.

3.2.3 Dumping Severity Example

The independence model for the dumping severity data is given by:

$$\log(m_{ij}) = \mu + \lambda_i^O + \lambda_j^D \tag{14}$$

The scores assigned for the following examples is $u_i = 3,\ 1,\ -1,\ -3$ and $v_j = 2,\ 0,\ -2$.
The uniform association model for the dumping severity data is given by:

$$\log(m_{ij}) = \mu + \lambda_i^O + \lambda_j^D + \beta(u_i - \bar{u})(v_j - \bar{v})$$

where $i = A, B, C, D$ and $j = N, S, M$. The parameter estimates are calculated and tabulated in table 14.

The row effects model for the dumping severity data is given by:

$$\log(m_{ij}) = \mu + \lambda_i^O + \lambda_j^D + \tau_i(v_j - \bar{v})$$

where $\{\tau_i\}$ are the row effect parameters. The parameter estimates are calculated and tabulated in table 15.

The column effects model for the dumping severity data is given by:

$$\log(m_{ij}) = \mu + \lambda_i^O + \lambda_j^D + \rho_j(u_i - \bar{u}))$$

where $\{\rho_j\}$ are the column effect parameters. The parameter estimates are calculated and tabulated in table 16.

Parameter	Estimate
$\widehat{\mu}$	3,3317
$\hat{\lambda}_1^O$	-0,1838
$\hat{\lambda}_2^O$	-0,0598
$\hat{\lambda}_3^O$	0,1090
$\hat{\lambda}_4^O$	0,1346
$\hat{\lambda}_1^D$	0,8644
$\hat{\lambda}_2^D$	0,1335
$\hat{\lambda}_3^D$	-0,9979
$\widehat{\tau}_1$	0,0549
$\widehat{\tau}_2$	0,0147
$\widehat{\tau}_3$	0,1040
$\widehat{\tau}_4$	-0,1736

Table 15: Parameter estimates of uniform association model for dumping severity data

Parameter	Estimate
$\widehat{\mu}$	3,3348
$\hat{\lambda}_1^O$	-0,1894
$\hat{\lambda}_2^O$	-0,0285
$\hat{\lambda}_3^O$	0,0956
$\hat{\lambda}_4^O$	0,1223
$\hat{\lambda}_1^D$	0,7569
$\hat{\lambda}_2^D$	0,1244
$\hat{\lambda}_3^D$	-0,8813
$\widehat{\rho}_1$	0,0822
$\widehat{\rho}_2$	-0,0223
$\widehat{\rho}_3$	0,0599

Table 16: Parameter estimates of uniform association model for dumping severity data

	G^2	df
Independence model	10,878	6
Uniform association model	4,4773	5
Row effects model	4,3991	3
Col effects model	4,1205	4

Table 17: Goodness of fit for different loglinear models for Dumping Severity Example

	G^2	df
Independence | Uniform	6,4007	1
Independence | Row	6,4789	3
Independence | Col	6,7575	2

Table 18: Measure of association for different loglinear models for Dumping Severity Example

Goodness of fit

It is observed from the table, that the independence model is an inadequate fit for this data. The uniform association model, fits the data the best. It is observed that the row effects model as well as the column effects model also fits the data well.

When the conditional test of independence is done, it is observed that there is strong evidence of an association between the different types of operations and the dumping severity, for all the models.

3.3 Logits for ordinal variables

Suppose there is $c \geq 2$ response categories. For a variable with response probabilities $(\pi_1, ...\pi_c)$ at certain combinations of levels of the explanatory variable, the conditional logit is defined as

$$\log(\frac{\pi_j}{\pi_k}) = \log(\frac{\pi_j/(\pi_j+\pi_k)}{\pi_k/(\pi_j+\pi_k)})$$

This gives the value of the log odds that a response is classified into category j instead of category k, given that the observation falls in one of the categories. Suppose $L_j = \log(\frac{\pi_j}{\pi_c})$ for $j = 1, ...(c-1)$ then $\log(\frac{\pi_j}{\pi_k}) = L_j - L_k$ for $1 \leq j \leq k \leq c-1$.

Special cases that will be considered is cumulative logit (proportional odds logit), continuation logit and adjacent-category logit [1].

3.3.1 Cumulative logit model

The cumulative logits are defined as follows for $j = 1, ...(c-1)$

$$L_j = \log\left[\frac{\pi_{j+1} + ... + \pi_c}{\pi_1 + ... + \pi_j}\right]$$

When calculating the cumulative logit all c categories are used.

Suppose the response variable is an ordinal variable and that j is a fixed cut point selected. Then the j^{th} cumulative logit in row i where $i = 1...r$ is:

$$L_{j(i)} = \log\left[\frac{m_{i,j+1} + ... + m_{ic}}{m_{i1} + ... + m_{ij}}\right] \tag{15}$$

If the row variable is ordinal then scores $\{u_i\}$ can be assigned to its levels. The cumulative logit model for the j^{th} cumulative logit values is $L_{j(i)} = \alpha_j + \beta_j(u_i - \bar{u})$ where $L_{j(i)}$ is given in equation (15). This model has $df = r - 2$. If the cumulative logit model holds and $\beta_j = 0$, then it implies that the two variables are independent. The difference between logits in adjacent rows is $L_{j(i+1)} - L_{j(i)} = \beta_j(u_{i+1} - u_i)$ which simplifies to $L_{j(i+1)} - L_{j(i)} = \beta_j$ when integer scores $\{u_i = i\}$ is used.

Dumping Severity Example

Then, for example, the cumulative log odds, for $i = 1$, with cut point $j = 1$, is $L_{1(1)} = \log\left(\frac{28+7}{61}\right) = -0,2413$ and cumulative log odds with cut point $j = 2$, is $L_{2(1)} = \log\left(\frac{7}{61+28}\right) = -1,1043$.

3.3.2 Continuation logit model

The continuation logits is defined as following for $j = 1, ...(c-1)$

$$L_{j(i)} = \log\left[\frac{\pi_{i,j+1}}{\pi_{i,1} + ... + \pi_{i,j}}\right]$$

Dumping Severity Example

Then, for example, the continuation log odds of dumping severity is $L_{1(1)} = \log\left(\frac{28}{61}\right) = -0,3382$ for $j = 1$ and $L_{2(1)} = \log\left(\frac{7}{61+28}\right) = -1,1043$ for $j = 2$.

3.3.3 Adjacent-category logit model

The adjacent-cumulative logits is defined as following for $j = 1, ...(c-1)$

$$L_{j(i)} = \log\left[\frac{\pi_{i,j+1}}{\pi_{i,j}}\right]$$

Dumping Severity Example

Then, for example, the adjacent log odds of dumping severity slight relative to dumping severity moderate, is $L_{2(1)} = \log\left(\frac{7}{28}\right) = -0,6021$ and the adjacent log odds of dumping severity none relative to dumping severity slight, is $L_{1(1)} = \log\left(\frac{28}{61}\right) = -0,3382$.

3.4 Logit models for ordinal data

Goodman [6] presented models using the logits discussed in Subsection 3.3.

3.4.1 Logit model for ordinal-ordinal tables

Consider two ordinal variables X and Y in a two-way table. Suppose Y is the response (column) variable and that $\{u_i\}$ scores are assigned to X, the ordinal row variable. Although the response variable is ordinal, it is not necessary to assign scores to it.

For a two-way table with fixed j, the ordinal logit regression model is defined as

$$L_{j(i)} = \alpha_j + \beta_j(u_i - \bar{u}) \tag{16}$$

for $i = 1...r$. If it is assumed that $\beta_1 = ... = \beta_{r-1}$ then (16) simplifies to the cumulative logit model

$$L_{j(i)} = \alpha_j + \beta(u_i - \bar{u}) \tag{17}$$

where $L_{j(i)}$ is given in equation 15) with $1 \leq i \leq r$ and $1 \leq j \leq c-1$. In (17) there is a single association parameter β and $c-1$ $\{\alpha_j\}$ parameters. The cumulative logit model has $df = rc - r - c$ and a simple interpretation of parameters. For each i, $L_{1(i)} \geq ... \geq L_{c-1(i)}$ such that $\{\alpha_j\}$are monotone decreasing. If the model holds and $\beta = 0$, then for every j, the j^{th} logit is the same in each row. This implies independence between X and Y. If $\beta > 0$, the logit increases as X increases. Thus conditional Y distributions are stochastically higher at high values of X [1].

For (17) it can be showed that $L_{j(b)} - L_{j(a)} = \beta\,(u_b - u_a)$ which implies that the log odds ratio is proportional to the distance between the rows a and b. If integer scores are applied to (17), then $L_{j(b)} - L_{j(a)} = \beta$.

Given that the uniform association model (11) holds, the test of independence ($H_0 : \beta = 0$) is based on the statistic $G^2(I|U) = G^2(I) - G^2(U)$.

3.4.2 Logit model for ordinal-nominal tables

Consider the explanatory variable X, with nominal levels. For each of the $c-1$ ways of forming the cumulative logits, the model is considered having row effects [1].

The row effects indicate the nature of the association. If the row effects are added for the levels of the nominal variable X, the model becomes

$$L_{j(i)} \quad = \quad \alpha_j + \tau_i^X \tag{18}$$

where $\sum \tau_i = 0$ for $1 \leq i \leq r$ and $1 \leq j \leq c-1$. The α_j parameter represents the average over the r rows of the j^{th} cumulative logit. The model (18) has $df = (r-1)(c-2)$. If rows a and b are considered, the difference in the logits are $L_{j(b)} - L_{j(a)} = \tau_b - \tau_a$. If $\tau_a < \tau_b$ it implies that the conditional Y distribution is stochastically higher in row b than in row a [1].

The deviation from the mean caused by the location of the cell in row i, is given by τ_i^X. If the explanatory variable is ordinal and it has a linear effect, then β has a slope interpretation. Hence β then represents the change in the logit and e^β represents the multiplicative change in the odds as the ordinal variable change with one unit.

The independence model in terms of cumulative logits are defined as $L_{j(i)} = \alpha_j$ for $1 \leq i \leq r$ and $1 \leq j \leq c-1$. This implies that all the row effects must be equal to zero to get the independence model.

The results of fitting models for separate logits are independent. Hence the G^2 statistics and the degrees of freedom values can be added together to obtain an over-all goodness-of-fit test.

If model (18) holds, the conditional test of independence is based on $G^2(I|R) = G^2(I) - G^2(R)$, where the G^2 statistic has an asymptotically chi-squared distribution under the null hypothesis with $df = r - 1$ [1].

3.4.3 Dumping Severity Example

The cumulative model for the dumping severity example is given as

$$L_{j(i)} = \alpha_j + \beta(u_i - \bar{u})$$

where $i = A, B, C, D$ and $\{u_i\}$ are the scores assigned to the operation variable. The estimate of the linear effect of operations on the logit of dumping severity is $\widehat{\beta} = 0,225$. Since $\widehat{\beta}$ is positive, it implies that the logit increase as the operation increases. The estimates for the logit model is $\widehat{\alpha}_1 = -0,320$ and $\widehat{\alpha}_2 = -2,074$. The goodness of fit for this model is $G^2 = 4,27$ based on $df = 5$. Hence the model is a very good fit.

Goodness of fit

It is observed from Table 19, that the basic logit model, which does not take the orderings into account, is an inadequate fit. The logit uniform model, which take account of the orderings, fits very well.

	G^2	df
Logit model (cumulative)	7,9204	3
Logit uniform model	4,27	5

Table 19: Goodness of fit for different logit models for Dumping Severity Example

4 Conclusion

It was concluded that when working with ordinal data, the loglinear and logit models, adjusted to take orderings into account, fitted the data much better.

References

[1] A. Agresti. *Analysis of Ordinal Categorical Data*. John Wiley & Sons, 1984.

[2] A. Agresti. Modelling ordered categorical data: Recent advances and future challenges. *Statistics in Medicine*, 18:2191–2207, 1999.

[3] A. Agresti. *Categorical Data Analysis*. John Wiley & Sons, 2002.

[4] M.S. Bartlett. Contingency table interactions. *Supplement to the Journal of the Royal Statistical Society*, 2(2):248–252, 1935.

[5] L.A. Goodman. Simple models for the analysis of association in cross-classifications having ordered categories. *Journal of the American Statistical Association*, 74(367):537–552, 1979.

[6] L.A. Goodman. The analysis of dependence in cross-classifications having ordered categories, using log-linear models for frequencies and log-linear models for odds. *Biometrics*, pages 149–160, 1983.

[7] Leo A Goodman and William H Kruskal. Measures of association for cross classifications. *Journal of the American Statistical Association*, 49(268):732–764, 1954.

[8] S.J. Haberman. Log-linear models for frequency tables with ordered classifications. *Biometrics*, 30(4):589–600, 1974.

[9] J.R. Zweiful J.J. Gart. On the bias of various estimators of logit and its variance with application to quantal bioassay. *Biometrika*, 54:181–187, 1967.

[10] K. Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from randon sampling. *Philosophical Magazine*, 50(302):157–175, 1900.

[11] K. Pearson. *Drapers' Company Research Memoirs: Biometric Series*, volume 1. Cambridge University Press, 1904.

[12] N. Balakrishnan S. Kotz, editor. *Encyclopedia of Statistical Sciences*, volume 9. John Wiley & Sons, 2 edition, 1986.

[13] G.U. Yule. On the association of attributes in statistics: with illustrations from the material of the childhood society. *Philosophical Transactions of the Royal Society of London.*, 194:257–319, 1900.

Appendix

Dumping Severity Example

Dataset

```
options nodate linesize = 64 pagesize = 250;
title1 'Dumping Severity';
proc format;
value aa 1 = 'A'
                2 = 'B'
                3 = 'C'
                4 = 'D';
value bb 1 = 'NONE'
                2 = 'SLIGHT'
                3 = 'MODERATE';
data DSE;
input op ds num @@;
label op = 'Operation';
label ds = 'Dumping Severity';
cards;
1 1 61  1 2 28  1 3 7
2 1 68  2 2 23  2 3 13
3 1 58  3 2 40  3 3 12
4 1 53  4 2 38  4 3 16
;
```

Frequency procedure

Output used in Section 2.1 and 2.5.

```
proc freq data = DSE;
weight num;
tables op*ds/ chisq expected cellchisq;
format op aa. ds bb.;
title3 'Chisquare test for independence';
run;
```

Dumping Severity

Chisquare test for independence

The FREQ Procedure

Frequency Expected Cell Chi-Square Percent Row Pct Col Pct

Table of op by ds

op(Operation)	ds(Dumping Severity) NONE	SLIGHT	MODERATE	Total
A	61 55.252 0.598 14.63 63.54 25.42	28 29.698 0.0971 6.71 29.17 21.71	7 11.05 1.4846 1.68 7.29 14.58	96 23.02
B	68 59.856 1.108 16.31 65.38 28.33	23 32.173 2.6152 5.52 22.12 17.83	13 11.971 0.0884 3.12 12.50 27.08	104 24.94
C	58 63.309 0.4453 13.91 52.73 24.17	40 34.029 1.0478 9.59 36.36 31.01	12 12.662 0.0346 2.88 10.91 25.00	110 26.38
D	53 61.583 1.1962 12.71 49.53 22.08	38 33.101 0.7251 9.11 35.51 29.46	16 12.317 1.1016 3.84 14.95 33.33	107 25.66
Total	240 57.55	129 30.94	48 11.51	417 100.00

Statistics for Table of op by ds

Statistic	DF	Value	Prob
Chi-Square	6	10.5419	0.1036
Likelihood Ratio Chi-Square	6	10.8782	0.0922
Mantel-Haenszel Chi-Square	1	6.2170	0.0127
Phi Coefficient		0.1590	
Contingency Coefficient		0.1570	
Cramer's V		0.1124	

Sample Size = 417

Odds ratios

Output used in Section 2.2.

```
proc iml;
X = {61 28 7, 68 23 13, 58 40 12, 53 38 16};
print X;
title1 'Odds ratios for dumping severity example';
n = nrow(X);
m = n-1;
k = ncol(X);
p = k-1;

/* Totals added*/
X1 = X[,+];
X2 = X||X1;
X3 = X2[+,];
Y = X2//X3;
print Y;
w = nrow(Y);
z = ncol(Y);

/* Fix moderate*/
/* Odds ratios for dumping severity*/
print 'Odds ratios for dumping severity';
A = J(w,p,0);
/* A is a 2*5 matrix
        Column one is ratio of none to moderate
        Column two is ratio of slight to moderate*/
do i = 1 to w;
        do j = 1 to p;
                A[i,1] = Y[i,1]/Y[i,j+1];
                A[i,2] = Y[i,j]/Y[i,j+1];
        end;
end;
print A;

/* Fix Operation D*/
/* Odds ratios for operation*/
print 'Odds ratios for operation';
B = J(m,z,0);
/* B is a 3*4 matrix
        Row 1 is ratio of A to D
        Row 2 is ratio of B to D
        Row 3 is ratio of C to D*/
do i = 1 to m;
        do j = 1 to z;
                B[1,j] = Y[1,j]/Y[i+1,j];
                B[2,j] = Y[2,j]/Y[i+1,j];
                B[3,j] = Y[i,j]/Y[i+1,j];
        end;
end;
print B;
```

Odds ratios for dumping severity example

X		
61	28	7
68	23	13
58	40	12
53	38	16

Odds ratios for dumping severity

A	
8.7142857	4
5.2307692	1.7692308
4.8333333	3.3333333
3.3125	2.375
5	2.6875

Odds ratios for operation

B			
1.1509434	0.7368421	0.4375	0.8971963
1.2830189	0.6052632	0.8125	0.9719626
1.0943396	1.0526316	0.75	1.0280374

Loglinear model fitted

Output used in Section 2.5.

```
proc catmod data = DSE;
weight num;
model op*ds = _response_ / ml nogls pred = freq noprofile;
loglin op | ds;
format op aa. ds bb.;
title3 'Loglinear model: Saturated model';
run;

proc catmod data = DSE;
weight num;
model op*ds = _response_ / ml nogls pred = freq noprofile;
loglin op ds;
format op aa. ds bb.;
title3 'Loglinear model: Independence';
run;
```

Dumping Severity

Loglinear model: Saturated model

The CATMOD Procedure

Data Summary			
Response	op*ds	**Response Levels**	12
Weight Variable	num	**Populations**	1
Data Set	DSE	**Total Frequency**	417
Frequency Missing	0	**Observations**	12

Maximum Likelihood Analysis
Maximum likelihood computations converged.

Maximum Likelihood Analysis of Variance			
Source	**DF**	**Chi-Square**	**Pr > ChiSq**
op	3	3.65	0.3019
ds	2	113.14	<.0001
op*ds	6	10.37	0.1099
Likelihood Ratio	0	.	.

Analysis of Maximum Likelihood Estimates					
Parameter		**Estimate**	**Standard Error**	**Chi-Square**	**Pr > ChiSq**
op	**A**	-0.1973	0.1214	2.64	0.1042
	B	-0.0203	0.1069	0.04	0.8492
	C	0.0844	0.1043	0.65	0.4184
ds	**NONE**	0.7633	0.0729	109.68	<.0001
	SLIGHT	0.1216	0.0814	2.23	0.1354
op*ds	**A NONE**	0.2179	0.1374	2.52	0.1127
	A SLIGHT	0.0810	0.1530	0.28	0.5967
	B NONE	0.1495	0.1236	1.46	0.2262
	B SLIGHT	-0.2927	0.1462	4.01	0.0453
	C NONE	-0.1143	0.1231	0.86	0.3534
	C SLIGHT	0.1559	0.1333	1.37	0.2423

Loglinear model: Saturated model

The CATMOD Procedure

Maximum Likelihood Predicted Values for Response Functions					
	Observed		Predicted		
Function Number	Function	Standard Error	Function	Standard Error	Residual
1	1.338285	0.28088	1.338285	0.28088	0
2	0.559616	0.313392	0.559616	0.313392	0
3	-0.82668	0.453163	-0.82668	0.453154	-1.76E-9
4	1.446919	0.277859	1.446919	0.27786	0
5	0.362905	0.325543	0.362905	0.325543	0
6	-0.20764	0.373394	-0.20764	0.373394	0
7	1.287854	0.282385	1.287854	0.282385	0
8	0.916291	0.295804	0.916291	0.295804	0
9	-0.28768	0.381881	-0.28768	0.381882	0
10	1.197703	0.285251	1.197703	0.285251	0
11	0.864997	0.29802	0.864997	0.29802	0

Maximum Likelihood Predicted Values for Frequencies						
		Observed		Predicted		
op	ds	Frequency	Standard Error	Frequency	Standard Error	Residual
A	NONE	61	7.216421	61	7.216424	1.806E-9
A	SLIGHT	28	5.110764	28	5.110766	8.29E-10
A	MODERATE	7	2.623451	7	2.623374	-1.21E-8
B	NONE	68	7.543956	68	7.543959	2.014E-9
B	SLIGHT	23	4.661697	23	4.661699	6.81E-10
B	MODERATE	13	3.548905	13	3.548906	3.85E-10
C	NONE	58	7.066318	58	7.066321	1.717E-9
C	SLIGHT	40	6.013574	40	6.013576	1.184E-9
C	MODERATE	12	3.413895	12	3.413896	3.55E-10
D	NONE	53	6.801749	53	6.801752	1.569E-9
D	SLIGHT	38	5.876833	38	5.876836	1.125E-9
D	MODERATE	16	3.922511	16	3.922513	4.74E-10

Loglinear model: Independence

The CATMOD Procedure

Data Summary			
Response	op*ds	**Response Levels**	12
Weight Variable	num	**Populations**	1
Data Set	DSE	**Total Frequency**	417
Frequency Missing	0	**Observations**	12

Maximum Likelihood Analysis
Maximum likelihood computations converged.

Maximum Likelihood Analysis of Variance			
Source	**DF**	**Chi-Square**	**Pr > ChiSq**
op	3	1.04	0.7911
ds	2	114.70	<.0001
Likelihood Ratio	6	10.88	0.0922

Analysis of Maximum Likelihood Estimates					
Parameter		**Estimate**	**Standard Error**	**Chi-Square**	**Pr > ChiSq**
op	**A**	-0.0812	0.0873	0.87	0.3522
	B	-0.00112	0.0849	0.00	0.9895
	C	0.0550	0.0834	0.43	0.5097
ds	**NONE**	0.7434	0.0709	109.93	<.0001
	SLIGHT	0.1226	0.0789	2.42	0.1202

Maximum Likelihood Predicted Values for Response Functions					
	Observed		**Predicted**		
Function Number	**Function**	**Standard Error**	**Function**	**Standard Error**	**Residual**
1	1.338285	0.28088	1.500957	0.211563	-0.16267
2	0.559616	0.313392	0.880131	0.219873	-0.32051
3	-0.82668	0.453163	-0.10848	0.140579	-0.7182
4	1.446919	0.277859	1.581	0.209661	-0.13408
5	0.362905	0.325543	0.960173	0.218044	-0.59727
6	-0.20764	0.373394	-0.02844	0.1377	-0.1792
7	1.287854	0.282385	1.637089	0.208406	-0.34924
8	0.916291	0.295804	1.016263	0.216838	-0.09997
9	-0.28768	0.381881	0.027652	0.135782	-0.31533

Loglinear model: Independence

The CATMOD Procedure

Maximum Likelihood Predicted Values for Response Functions					
	Observed		Predicted		
Function Number	Function	Standard Error	Function	Standard Error	Residual
10	1.197703	0.285251	1.609438	0.158102	-0.41173
11	0.864997	0.29802	0.988611	0.169062	-0.12361

Maximum Likelihood Predicted Values for Frequencies						
		Observed		Predicted		
op	ds	Frequency	Standard Error	Frequency	Standard Error	Residual
A	NONE	61	7.216421	55.2518	5.466064	5.748202
A	SLIGHT	28	5.110764	29.69784	3.434248	-1.69784
A	MODERATE	7	2.623451	11.05036	1.797204	-4.05036
B	NONE	68	7.543956	59.85611	5.673992	8.143885
B	SLIGHT	23	4.661697	32.17266	3.60725	-9.17266
B	MODERATE	13	3.548905	11.97122	1.917255	1.028777
C	NONE	58	7.066318	63.30935	5.823566	-5.30935
C	SLIGHT	40	6.013574	34.02878	3.734924	5.971223
C	MODERATE	12	3.413895	12.66187	2.007026	-0.66187
D	NONE	53	6.801749	61.58273	5.749429	-8.58273
D	SLIGHT	38	5.876833	33.10072	3.671295	4.899281
D	MODERATE	16	3.922511	12.31655	1.962166	3.683453

Dataset For Cumulative Logit

```
proc format;
value aa 1 = 'A'
                2 = 'B'
                3 = 'C'
                4 = 'D';
value bb 1 = 'NONE'
                2 = 'SOME';
data DSEadj;
input op ds num @@;
label op = 'Operation';
label ds = 'Dumping Severity';
cards;
1 1 61  1 2 35
2 1 68  2 2 36
3 1 58  3 2 52
4 1 53  4 2 54
;
```

Logit model fitted

Output used in Section 2.6.

```
proc freq data = DSEadj;
weight num;
tables op*ds/ chisq expected cellchisq;
format op aa. ds bb.;
title3 'Chisquare test for independence';
run;

proc catmod data = DSEadj;
weight num;
model ds = op /ml;
format op aa. ds bb.;
title3 'Logit model';
run;
```

Dumping Severity

Chisquare test for independence

The FREQ Procedure

Frequency Expected Cell Chi-Square Percent Row Pct Col Pct

Table of op by ds			
	ds(Dumping Severity)		
op(Operation)	NONE	SOME	Total
A	61 55.252 0.598 14.63 63.54 25.42	35 40.748 0.8109 8.39 36.46 19.77	96 23.02
B	68 59.856 1.108 16.31 65.38 28.33	36 44.144 1.5024 8.63 34.62 20.34	104 24.94
C	58 63.309 0.4453 13.91 52.73 24.17	52 46.691 0.6037 12.47 47.27 29.38	110 26.38
D	53 61.583 1.1962 12.71 49.53 22.08	54 45.417 1.6219 12.95 50.47 30.51	107 25.66
Total	240 57.55	177 42.45	417 100.00

Statistics for Table of op by ds

Statistic	DF	Value	Prob
Chi-Square	3	7.8865	0.0484
Likelihood Ratio Chi-Square	3	7.9204	0.0477
Mantel-Haenszel Chi-Square	1	6.3862	0.0115
Phi Coefficient		0.1375	
Contingency Coefficient		0.1362	
Cramer's V		0.1375	

Sample Size = 417

Dumping Severity

Logit model

The CATMOD Procedure

Data Summary			
Response	ds	**Response Levels**	2
Weight Variable	num	**Populations**	4
Data Set	DSE	**Total Frequency**	417
Frequency Missing	0	**Observations**	8

Population Profiles		
Sample	**op**	**Sample Size**
1	A	96
2	B	104
3	C	110
4	D	107

Response Profiles	
Response	**ds**
1	NONE
2	SOME

Maximum Likelihood Analysis
Maximum likelihood computations converged.

Maximum Likelihood Analysis of Variance			
Source	**DF**	**Chi-Square**	**Pr > ChiSq**
Intercept	1	10.19	0.0014
op	3	7.82	0.0498
Likelihood Ratio	0	.	.

Analysis of Maximum Likelihood Estimates					
Parameter		**Estimate**	**Standard Error**	**Chi-Square**	**Pr > ChiSq**
Intercept		0.3205	0.1004	10.19	0.0014
op	**A**	0.2350	0.1805	1.70	0.1928
	B	0.3155	0.1770	3.18	0.0747
	C	-0.2113	0.1683	1.58	0.2092

Ordinal odds ratios

Output used in Section 3.1.

```
proc iml;
X = {61 28 7, 68 23 13, 58 40 12, 53 38 16};
print X;
title1 'Odds ratios for dumping severity example';
/* Getting cross product ratio */
n = nrow(X);
m = n-1;
k = ncol(X);
p = k-1;
Cross_Product_Ratio = J(m,p,0);
do i = 1 to m;
        do j = 1 to p;
        Cross_Product_Ratio[i,j] = (X[i,j]*X[i+1,j+1])/(X[i+1,j]*X[i,j+1]);
        end;
end;
print Cross_Product_Ratio;
/* Totals added*/
X1 = X[,+];
X2 = X||X1;
X3 = X2[+,];
Y = X2//X3;
print Y;
w = nrow(Y);
z = ncol(Y);

T1 = J(m,p,0);
do i = 1 to m;
        T1[i,1]=(X[i,1]*(X[i+1,2]+X[i+1,3]))/(X[i+1,1]*(X[i,2]+X[i,3]));
        T1[i,2]=(X[i+1,3]*(X[i,1]+X[i,2]))/(X[i,3]*(X[i+1,1]+X[i+1,2]));
end;
print T1;

T2 = J(m,p,0);
do i = 1 to m;
        do j = 1 to p;
                C = Y[1:i,1:j];
                D = Y[i+1:m+1,1:j];
                E = Y[1:i,j+1:3];
                F = Y[i+1:m+1,j+1:3];
                T2[i,j]=(C[+]*F[+])/(E[+]*D[+]);
        end;
end;
print T2;
quit;
```

Odds ratios for dumping severity example

X		
61	28	7
68	23	13
58	40	12
53	38	16

Cross_Product_Ratio	
0.7368697	2.2608696
2.0389805	0.5307692
1.0396226	1.4035088

T1	
0.9226891	1.8163265
1.6934866	0.8571429
1.1364296	1.4358974

T2	
1.3826018	1.8617347
1.735059	1.3333333
1.5490106	1.5274725

Ordinal Loglinear models fitted

```
data DSEO;
input op1 op2 op3 ds1 ds2 num x;
cards;
 1  0  0  1  0 61  6
 1  0  0  0  1 28  0
 1  0  0 -1 -1  7 -6
 0  1  0  1  0 68  2
 0  1  0  0  1 23  0
 0  1  0 -1 -1 13 -2
 0  0  1  1  0 58 -2
 0  0  1  0  1 40  0
 0  0  1 -1 -1 12  2
-1 -1 -1  1  0 53 -6
-1 -1 -1  0  1 38  0
-1 -1 -1 -1 -1 16  6
;
proc genmod data=DSEO order=data;
model num = op1 op2 op3 ds1 ds2 x / link=log dist=poisson lrci type3 obstats;
title 'Uniform association model';
run;

data DSEOR;
input op1 op2 op3 ds1 ds2 num tau1 tau2 tau3;
cards;
 1  0  0  1  0 61  2  0  0
 1  0  0  0  1 28  0  0  0
 1  0  0 -1 -1  7 -2  0  0
 0  1  0  1  0 68  0  2  0
 0  1  0  0  1 23  0  0  0
 0  1  0 -1 -1 13  0 -2  0
 0  0  1  1  0 58  0  0 -2
 0  0  1  0  1 40  0  0  0
 0  0  1 -1 -1 12  0  0  2
-1 -1 -1  1  0 53 -2 -2 -2
-1 -1 -1  0  1 38  0  0  0
-1 -1 -1 -1 -1 16  2  2  2
;
proc genmod data=DSEOR order=data;
model num = op1 op2 op3 ds1 ds2 tau1 tau2 tau3 / link=log dist=poisson lrci type3 obstat
title 'Row effects model';
run;
data DSEOC;
input op1 op2 op3 ds1 ds2 num tau1 tau2;
cards;
 1  0  0  1  0 61  3  0
 1  0  0  0  1 28  0  3
 1  0  0 -1 -1  7 -3 -3
 0  1  0  1  0 68  1  0
 0  1  0  0  1 23  0  1
 0  1  0 -1 -1 13 -1 -1
```

```
 0   0   1   1   0 58 -1   0
 0   0   1   0   1 40   0 -1
 0   0   1 -1 -1 12   1   1
-1 -1 -1   1   0 53 -3   0
-1 -1 -1   0   1 38   0 -3
-1 -1 -1 -1 -1 16   3   3
;
proc genmod data=DSEOC order=data;
model num = op1 op2 op3 ds1 ds2 tau1 tau2 / link=log dist=poisson lrci type3 obstats;
title 'Col effects model';
run;
```

The GENMOD Procedure

Model Information	
Data Set	WORK.DSEO
Distribution	Poisson
Link Function	Log
Dependent Variable	num

Number of Observations Read	12
Number of Observations Used	12

Parameter Information	
Parameter	**Effect**
Prm1	Intercept
Prm2	op1
Prm3	op2
Prm4	op3
Prm5	ds1
Prm6	ds2
Prm7	x

Criteria For Assessing Goodness Of Fit			
Criterion	**DF**	**Value**	**Value/DF**
Deviance	5	4.5898	0.9180
Scaled Deviance	5	4.5898	0.9180
Pearson Chi-Square	5	4.4773	0.8955
Scaled Pearson X2	5	4.4773	0.8955
Log Likelihood		1136.6737	
Full Log Likelihood		-33.3309	
AIC (smaller is better)		80.6618	
AICC (smaller is better)		108.6618	
BIC (smaller is better)		84.0562	

Algorithm converged.

The GENMOD Procedure

Analysis Of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	Likelihood Ratio 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	3.3319	0.0620	3.2065	3.4498	2889.43	<.0001
op1	1	-0.2010	0.1016	-0.4050	-0.0062	3.91	0.0479
op2	1	-0.0330	0.0857	-0.2041	0.1321	0.15	0.7001
op3	1	0.0993	0.0859	-0.0717	0.2656	1.33	0.2480
ds1	1	0.7605	0.0722	0.6207	0.9042	110.84	<.0001
ds2	1	0.1332	0.0793	-0.0226	0.2889	2.82	0.0932
x	1	0.0407	0.0164	0.0088	0.0732	6.15	0.0132
Scale	0	1.0000	0.0000	1.0000	1.0000		

Note: The scale parameter was held fixed.

LR Statistics For Type 3 Analysis			
Source	DF	Chi-Square	Pr > ChiSq
op1	1	4.10	0.0430
op2	1	0.15	0.6991
op3	1	1.31	0.2520
ds1	1	119.81	<.0001
ds2	1	2.81	0.0937
x	1	6.29	0.0122

The GENMOD Procedure

Model Information	
Data Set	WORK.DSEOR
Distribution	Poisson
Link Function	Log
Dependent Variable	num

Number of Observations Read	12
Number of Observations Used	12

Parameter Information	
Parameter	**Effect**
Prm1	Intercept
Prm2	op1
Prm3	op2
Prm4	op3
Prm5	ds1
Prm6	ds2
Prm7	tau1
Prm8	tau2
Prm9	tau3

Criteria For Assessing Goodness Of Fit			
Criterion	**DF**	**Value**	**Value/DF**
Deviance	3	4.4034	1.4678
Scaled Deviance	3	4.4034	1.4678
Pearson Chi-Square	3	4.3991	1.4664
Scaled Pearson X2	3	4.3991	1.4664
Log Likelihood		1136.7669	
Full Log Likelihood		-33.2377	
AIC (smaller is better)		84.4754	
AICC (smaller is better)		174.4754	
BIC (smaller is better)		88.8396	

Algorithm converged.

The GENMOD Procedure

Analysis Of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	Likelihood Ratio 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	3.3317	0.0620	3.2063	3.4497	2887.16	<.0001
op1	1	-0.1838	0.1136	-0.4164	0.0308	2.62	0.1058
op2	1	-0.0598	0.1075	-0.2783	0.1447	0.31	0.5780
op3	1	0.1090	0.0988	-0.0893	0.2991	1.22	0.2700
ds1	1	0.8644	0.1445	0.5839	1.1512	35.80	<.0001
ds2	1	0.1335	0.0794	-0.0223	0.2893	2.83	0.0925
tau1	1	0.0549	0.0723	-0.0865	0.1974	0.58	0.4474
tau2	1	0.0147	0.0722	-0.1273	0.1564	0.04	0.8388
tau3	1	0.1040	0.1196	-0.1336	0.3366	0.76	0.3846
Scale	0	1.0000	0.0000	1.0000	1.0000		

Note: The scale parameter was held fixed.

LR Statistics For Type 3 Analysis			
Source	DF	Chi-Square	Pr > ChiSq
op1	1	2.79	0.0949
op2	1	0.32	0.5745
op3	1	1.19	0.2760
ds1	1	37.03	<.0001
ds2	1	2.82	0.0930
tau1	1	0.58	0.4469
tau2	1	0.04	0.8388
tau3	1	0.75	0.3871

The GENMOD Procedure

Model Information	
Data Set	WORK.DSEOC
Distribution	Poisson
Link Function	Log
Dependent Variable	num

Number of Observations Read	12
Number of Observations Used	12

Parameter Information	
Parameter	**Effect**
Prm1	Intercept
Prm2	op1
Prm3	op2
Prm4	op3
Prm5	ds1
Prm6	ds2
Prm7	tau1
Prm8	tau2

Criteria For Assessing Goodness Of Fit			
Criterion	**DF**	**Value**	**Value/DF**
Deviance	4	4.2107	1.0527
Scaled Deviance	4	4.2107	1.0527
Pearson Chi-Square	4	4.1205	1.0301
Scaled Pearson X2	4	4.1205	1.0301
Log Likelihood		1136.8632	
Full Log Likelihood		-33.1413	
AIC (smaller is better)		82.2827	
AICC (smaller is better)		130.2827	
BIC (smaller is better)		86.1619	

Algorithm converged.

The GENMOD Procedure

Analysis Of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	Likelihood Ratio 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	3.3348	0.0618	3.2098	3.4523	2914.46	<.0001
op1	1	-0.1894	0.1030	-0.3963	0.0080	3.38	0.0658
op2	1	-0.0285	0.0860	-0.2002	0.1372	0.11	0.7401
op3	1	0.0956	0.0860	-0.0756	0.2621	1.23	0.2667
ds1	1	0.7569	0.0721	0.6173	0.9006	110.06	<.0001
ds2	1	0.1244	0.0804	-0.0335	0.2824	2.39	0.1220
tau1	1	0.0822	0.0327	0.0186	0.1470	6.32	0.0120
tau2	1	-0.0223	0.0362	-0.0935	0.0489	0.38	0.5380
Scale	0	1.0000	0.0000	1.0000	1.0000		

Note: The scale parameter was held fixed.

LR Statistics For Type 3 Analysis			
Source	DF	Chi-Square	Pr > ChiSq
op1	1	3.53	0.0602
op2	1	0.11	0.7394
op3	1	1.21	0.2706
ds1	1	118.83	<.0001
ds2	1	2.39	0.1223
tau1	1	6.44	0.0112
tau2	1	0.38	0.5381

Ordinal Logit model fitted

```
data d1;
input operation  severity f;
cards;
-1.5  1 61
-1.5  0 28
-1.5 -1  7
-0.5  1 68
-0.5  0 23
-0.5 -1 13
 0.5  1 58
 0.5  0 40
 0.5 -1 12
 1.5  1 53
 1.5  0 38
 1.5 -1 16
;
ods graphics on;
proc logistic data=d1;
freq f;
model severity=operation / covb;     title 'AgrestiOrdinalp120';
run;
ods graphics off;
```

The LOGISTIC Procedure

Model Information	
Data Set	WORK.D1
Response Variable	severity
Number of Response Levels	3
Frequency Variable	f
Model	cumulative logit
Optimization Technique	Fisher's scoring

Number of Observations Read	12
Number of Observations Used	12
Sum of Frequencies Read	417
Sum of Frequencies Used	417

Response Profile		
Ordered Value	**severity**	**Total Frequency**
1	-1	48
2	0	129
3	1	240

Probabilities modeled are cumulated over the lower Ordered Values.

Model Convergence Status
Convergence criterion (GCONV=1E-8) satisfied.

Score Test for the Proportional Odds Assumption		
Chi-Square	**DF**	**Pr > ChiSq**
0.0211	1	0.8846

Model Fit Statistics		
Criterion	**Intercept Only**	**Intercept and Covariates**
AIC	779.420	774.812
SC	787.487	786.912
-2 Log L	775.420	768.812

The LOGISTIC Procedure

Testing Global Null Hypothesis: BETA=0			
Test	**Chi-Square**	**DF**	**Pr > ChiSq**
Likelihood Ratio	6.6081	1	0.0102
Score	6.6042	1	0.0102
Wald	6.4872	1	0.0109

Analysis of Maximum Likelihood Estimates						
Parameter		**DF**	**Estimate**	**Standard Error**	**Wald Chi-Square**	**Pr > ChiSq**
Intercept	**-1**	1	-2.0742	0.1549	179.2222	<.0001
Intercept	**0**	1	-0.3202	0.1001	10.2272	0.0014
operation		1	0.2247	0.0882	6.4872	0.0109

Odds Ratio Estimates			
Effect	**Point Estimate**	**95% Wald Confidence Limits**	
operation	1.252	1.053	1.488

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	44.0	**Somers' D**	0.128
Percent Discordant	31.2	**Gamma**	0.170
Percent Tied	24.7	**Tau-a**	0.072
Pairs	48672	**c**	0.564

Estimated Covariance Matrix			
Parameter	**Intercept__1**	**Intercept_0**	**operation**
Intercept__1	0.024005	0.006526	-0.00162
Intercept_0	0.006526	0.010028	-0.00077
operation	-0.00162	-0.00077	0.007781

Using Kaplan-Meier and Cox regression to study graduation in the Department of Statistics

Kelotseetseng Letlhogile 13153766

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Miss I.Maharela, Co-supervisor: Prof D.Chen

Department of Statistics, University of Pretoria

30 October 2017

Abstract

According to the National Plan for Higher Education assembled in 2001 by the Department of Education, South Africa is one of the countries in the world with the lowest graduation rate of 15% [7]. This is a concern since changes have taken place in the employment distribution and there is huge amount of shortage of high-level skills in the labour market [7]. Although this is a national problem, this study will only focus on graduation of undergraduates in the Department of Statistics at the University of Pretoria. The aim of the research is to master the theory on survival analysis, give a clear understanding on Kaplan-Meier methods and Cox regression and implement it to study and analyze graduation rates of undergraduate students.

Declaration

I, *Kelotseetseng Amanda Letlhogile,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Kelotseetseng Amanda Letlhogile

_ _

Prof Ding-Geng Chen, Iketle Maharela

_ _

Date

Acknowledgements

A number of people deserve recognition for help and support I received from them. It is my pleasure to extend my gratitude towards all of them.

Firstly, I would like to sincerely thank my Co-supervisor, Prof Ding-Geng Chen and supervisors, Mrs Iketle Maharela and Mr Hossein Masoumi. Their constant guidance and support is highly appreciated. Their commitment on helping me, providing me with helpful ideas never went unnoticed.

Secondly, I would like to acknowledge the financial support from the Center for AI Research. Their generosity is highly appreciated for this academic year.

Contents

List of Figures

List of Tables

1 Introduction

Graduation rate is the percentage of a university's students who complete the degree in a prescribed period. There is a minimum number of years in which a student is supposed to graduate but they may exceed the recommended time limit [4]. Therefore, study duration may last longer than anticipated and this results in long tails for the corresponding distributions. The study of graduation rates is important since it is essential to estimate the likelihood and duration of graduation of students enrolled in a program [4]. The estimation indicates the potential performance for the university, the longer the duration, the more funding is required for education and lastly, also because the results are used to predict future occurrences [4].

In South Africa, a degree is a requirement to enter a rewarding profession and the monthly earnings of an individual with matric and individual increased from 182% in 2000 and 241% in 2007 [14]. Furthermore, deferred degree or a failure to obtain one is an opportunity cost because the time spent at university could be used for more productive purposes [14]. The research topic involves application of Kaplan-Meier method and Cox regression to thoroughly investigate the duration of graduation of undergraduate students in the Department of Statistics.

Survival Analysis, also known as duration or event history analysis in various fields, is defined as the set of statistical methods for analyzing data which focus on the occurrence and duration of events of interest. Kaplan-Meier and Cox regression will both be used in this research to study the survival data of undergraduates.

Kaplan-Meir method was introduced by Edward L. Kaplan and Paul Meier back in 1958. They both published a paper explaining how to work with censored data/observations. Eventually Kaplan-Meier curves and estimates became a popular approach to deal with survival times. It is an estimator mainly used to analize censored data. Its curve is easy to calculate and few assumptions are required. This method is a good choice since our data is censored.

Kalamatianou from Greece researched the duration of undergraduates in a Greek university. 10 313 students were used as a sample. He proposed the general distribution of the duration of studies. He uses two survival models: parametric and non-parametric. For non-parametric estimation, he uses the Kaplan-Meier Method and under parametric, maximum likelihood function, confidence intervals for the parameters and goodness of fit are used to estimate graduation rate of Social and Political Science degree at Greek university.[4]

The results show that the parametric model proposed describes the distribution provided by the

non-parametric estimation [4]. It was observed that there is a small percentage of students graduating just after the minimum time whereas a high percentage of them never graduate [4]. Genders where also compared, the test shows that there is a significant difference between duration of studies of male and female students. It is therefore concluded that female students graduated at faster rate and earlier than most men.[4] The method in his paper provides a useful methodology for estimating the duration of studies. Similarly to Kalamatianou's paper, this paper uses the Kaplan-Meier Method and Cox regression to carry out the same objective.

Mike Murray from the university of KwaZulu-Natal released a paper that introduced a methodology that can be used to assist in identifying student and/or institutional factors influencing the consequences a student faces when they dropout [11]. This new methodology was applied to the student registered at the University of KwaZulu-Natal from year 2004 to 2012 [11]. The results are that white females are more likely to graduate, which implies that, on average, they will need fewer extra credit points to graduate [11]. Also, being an African with some form of financial aid and leaving in residence increases the duration of a student lingering in the system before an academic exclusion. [11]

R.Christopher from the University of Cape Town uses a survival analysis approach to examine the determinants of graduation and exclusion at the University of Cape Town. In his samples, he selected South African students in Commerce, EBE and Science faculties registered from 2006 to 2013 [14]. A person-period longitudinal data set was created to track the student's progress every year until they graduated, excluded or were censored.[14]

General conclusions drawn from the results were that females from the Commerce and Science faculties are more likely to graduate than be academically excluded as compared to males [14]. Age is not a major factor in building university's success. Students on financial aid are more likely to be excluded than graduate and those who had a higher high school GPAs are more likely to graduate.[14]

Xu in 2012 expanded an adjusted Kaplan-Meier estimator to decrease impacts by utilizing inverse probability of treatment weight (IPTW) [18]. The article proposes a weighted log-rank looking at survival functions among treatment groups and compares both the adjusted Kaplan-Meier estimates and modified log-rank and Wilcoxon tests.[17][18]

Pourhoseingholi et al compares two survival regression methods in his paper, namely Cox regression and parametric models,[12]. For parametric models he performed Exponential, Weibull and Lognormal regression to compare the efficiency of the model. [12]. In multivariate analysis, Cox and exponential are almost identical. And in univariate analysis, the data supports log normal regression more that it does with parametric models and parametric models provide more precise estimates as compared to Cox

regression. [12]

2 Methodology

This section outlines the statistical methods used in the study.

2.1 Basic functions in Survival Analysis

Let T be a non-zero random variable denoting the time to event of interest, graduation. The waiting time to graduation is referred to as survival time.

2.1.1 Survival Function

Assume that T is continuous random variable which has the density function $f(t)$, the probability density function of t which explains the likelihood of observing at time t and $F(t)$ is the cumulative density function which explains the probability of observing T less than or equal to some time t [15]. The survival function denotes the probability that graduation has not taken place yet by time t. These functions and their relationships are mathematically expressed below.

In general, the probability of observing the survival time in the interval [a,b] is

$$P(a \leq T \leq b) = \int_a^b f(t)dt = F(b) - F(a) \tag{1}$$

The cumulative density function is defined as:

$$F(t) = \int_0^t f(t)dt \tag{2}$$

This relationship above also implies that

$$f(t) = \frac{dF(t)}{dt} \tag{3}$$

This density function must satisfy two properties, namely:

i) $f(x) \geq 0 \; for \; all \; x$ and

ii) $\int_{-\infty}^{\infty} f(x)dx = 1$

The survival function $S(t)$ is

$$S(t) = 1 - F(t) = P\{T > t\} \tag{4}$$

which has the following properties:

i) It is non-increasing

ii) $t = 0$, $S(t) = 1$, that is the probability for surviving after time 0 is 1

iii) $At\, t = \infty$, $S(t) = S(\infty) = 0$, that is, as time goes to infinity, the survival curve goes to 0

Theoretically, the survival function is smooth.

2.1.2 Hazard Function

This function describes the likelihood of graduation occurring at time t conditional on the student's graduating at that time t [15]. It is the probability that an individual dies between t and $t+\triangle$ divided by the probability that the individual survived to t [16]. This survival function is the probability to graduate at or beyond time t. [15]

Mathematically, the hazard function is defined as,

$$\begin{aligned} h(t) &= P\{t < T < (t+\triangle) \mid T > t\} \\ &= P\{expiring \in (t, t+\triangle) \mid survived\, past\, time\, t\} \\ &= f(t)/(1-F(t)) \\ &= f(t)/S(t) \end{aligned} \tag{5}$$

In words, the rate of graduating at time t is the density of events at t, divided by the probability of studying until t without experiencing the event.

Cumulative hazard function is defined as follow

$$\begin{aligned} H(t) &= \int_0^t h(t)S(t) \\ &= \int_0^t -\frac{dln\,[S(t)]}{dt}dt \\ &= -\,ln[S(t)] \end{aligned} \tag{6}$$

This can be thought of as the sum of risks a student faces from time 0 to time t. The following relations exist between these functions:

i)

$$\begin{aligned} h(t) &= -\frac{S'(t)}{S(t)} \\ &= -\frac{d}{dt} logS(t) \end{aligned}$$

ii)

$$S(t) = \exp(-H(t)) \; since \; S(0) = 1$$

iii)

$$f(t) = h(t)S(t)$$

2.2 Types of censoring in Survival Analysis

Censoring is an important feature in survival analysis. It is most likely possible that in this study, some students may not experience graduation therefore the data may be incomplete [14]. Some of the student will not have graduated by the end of this study but will eventually do so in the future [14]. Also some will never experience graduation even though they are still attending at the end of the programs. [14] [3]

Let $T_1, T_2, T_3, ..., T_n$ be independently, identically distributed with distribution $F(t)$, where T_i denotes the survival time of the i^{th} student. [9]

2.2.1 Type 1 Censoring

The most common type of censoring in survival analysis is right-censoring. This type of censoring occurs when student do not graduate during the observation period of the study [14]. This is considered as non-informative since the fact that a student is censored is does not mean that he/she will not graduate [14]

Let t_c be some fixed pre-assigned censoring time. $Y_1, Y_2, Y_3, ..., Y_n$ observed where [9]

$$Y_i = \left\{ \begin{array}{ll} T_i & if \, T_i \leq t_c \\ t_c & if \, T_i > t_c \end{array} \right\} \tag{7}$$

2.2.2 Type 2 Censoring

It occurs when it is known in advance, when the event of interest will occur. This is advantageous since the number of failures are already known in advance.

Let $r < n$ be a fixed number and $T_{(1)} < T_{(2)} < T_{(3)} < ... < T_{(n)}$ be ordered statistics of $T_1, T_2, T_3, ..., T_n$. Observations stops after the $r - th$ failure therefore $T_{(1)}, T_{(2)}, T_{(3)}, ..., T_{(r)}$ are observed [9]. Full ordered sample observed is $Y_{(1)} = T_{(1)} \cdots Y_{(r)} = T_{(r)} Y_{(r+1)} = T_{(r)} \cdots Y_{(n)} = T_{(r)}$.

2.2.3 Random Censoring

This type occurs when the number of censored observations and censoring levels are random. When a student leaves the university during the study period before graduating he/she has a random censored

value.This type is common in clinical trials.

Let $C_1, C_2, C_3, ..., C_n$ be independent identically distributed variables with distribution function $G(f)$ [9]. C_i the censoring time corresponding with T_i. $(Y_1, \delta_1), (Y_2, \delta_2), ..., (Y_n, \delta_n)$ are observed where [9]

$$Y_i = max(T_i, C_i) \tag{8}$$

$$\delta_i = \left\{ \begin{array}{ll} 1 & if\ T_i \leq C_i, that\ is\ , T_i\ is\ not\ censored \\ 0 & if\ T_i > C_i, that\ is\ ,\ T_i is\ censored \end{array} \right\} \tag{9}$$

2.2.4 Other types of censoring

The other 2 types of censoring are left and interval censoring. Left censoring occurs when the student have already graduated but it is not known when exactly the graduation took place [14]. Interval censoring, which is less common, occurs when it is known that graduation took place between two point in time but not sure of the exact time [14][3]. In this study, a year in which a student experience graduation is known therefore only right censoring is involved [14][3]. Truncation is another feature common to censoring. They are both mostly confused because the form of their non-parametric maximum likelihood estimates are similar [8]. Truncation is a model for selection bias[8] which means that, when estimating truncated data, it uses methods for selection bias models.[8]

2.3 Survival Methods

2.3.1 Non-parametric method

These type of methods assume that the data distribution cannot be defined in terms of a finite set of parameters.

Kaplan-Meier

Kaplan-Meier estimate, also called the product limit estimate involves computations of probabilities of an event occurring at a certain point in time. These successive probabilities are multiplied by probabilities computed earlier to get the final estimates [2]. It is widely used to estimate and graph the survival probabilities as a function of time.

Consider the following model:

$$S(t) = \left\{ \begin{array}{ll} 1 & t < \beta \\ \alpha + (\gamma - \alpha) S_0(t - \beta) & t \geqslant \beta \end{array} \right\} \tag{10}$$

where β is the minimum time required to graduate, α accounts for students that will never graduate and represent the students that complete later than the minimum period [4]. S_0 is the distribution of students that did not experience graduation immediately. Let $t_1 < t_2 < ... < t_k$ be the observed k graduation times of n students. Let $d_1, ..., d_k$ be the number of observed events at $t_1, ..., t_k$ respectively. The number of individuals at risk, that is graduating before t_j is n_j. $R(t_j)$ denotes set of individuals graduating just before t_j, $j = 1, \ldots, k$.[4][13]

Let

$$h_j = P(T = t_j | T \geq t_j) \tag{11}$$

be the hazard function for the discrete distribution. [5][4] It follows that the survival function is

$$S(t) = \prod_{j \epsilon R(t)'} (1 - h_j)\,,\, t \geq 0 \tag{12}$$

, where $R(t)'$ is the complement of the risk at t. The likelihood function of the above survival function is

$$\hat{S}(t) = \prod_{j \epsilon R(t)'} (1 - \hat{h}_j) \tag{13}$$

h_j is the conditional probability of graduation taking place, $n_j - h_j$ is the number of student who will not graduate. And $1 - h_j$ is the probability of graduating after time t_j given that students will still be active on the programme until h_j. Therefore the likelihood function is

$$logL\,[h_1,\, h_2, \ldots,\, h_k] = \sum_{j=1}^{k} [d_j log h_j + (n_j - d_j) log\,\{1 - h_j\}] \tag{14}$$

The maximum likelihood estimator of h_j is

$$\hat{h}_i = \frac{d_i}{n_i} \tag{15}$$

where $i = 1,\, 2, \ldots,\, k$, therefore the estimator for the survivor function is

$$\hat{S}(t) = \prod_{j \epsilon R(t)'} \left(1 - \frac{d_i}{n_i}\right) \tag{16}$$

which is known as the Kaplan-Meier estimator, and its variance is given as

$$\widehat{V}(\widehat{S}(t)) = \left[\widehat{S}(t)\right]^2 \sum_{j \epsilon R(t)'} \frac{d_j}{n_j(n_j - d_j)} \tag{17}$$

The justification of the estimate is that,

$$P(survive\,to\,t_1) = P(survive\ from\,t_0\,to\,t_1) \tag{18}$$

and

$$P(survive\,from\,t_1\,to\,t_2) = P(survived\ from\,t_1\,to\,t_2 \mid an\,indvidual\,survived\,from\,t_0\,to\,t_1) \tag{19}$$

and so on. Since there are no graduation between t_{i-1} and t_i, the probability between these corresponding times is zero[13] . The conditional probability of graduating at t_i given that an individual did not graduate before is estimated by $\frac{d_i}{n_i}$ and the conditional probability of not graduating is $\left(1-\frac{d_i}{n_i}\right)$. In statistical terms,

$$P(graduating\,at\,t_i \mid student\,did\,not\,graduate\,before) = \frac{d_i}{n_i} \tag{20}$$

[13]

The asymptotic confidence intervals for $S(t)$ is

$$\widehat{S}(t) \pm z_{\alpha/2}\sqrt{\widehat{V}(\widehat{S}(t))} \tag{21}$$

where $z_{\alpha/2}$ is the $\alpha 100\%$ percentile of the normal distribution at α significance level. The maximum likelihood estimators for the parameters are as follows:

The estimator for β is :

$$\widehat{\beta} = min\{t_i : i = 1, 2, ..., k\} \tag{22}$$

$\alpha = S(t_{max} \dot{+} \varepsilon)$, t_{max} denotes the maximum observed duration of studies, is the estimated probability that an individual will never graduate. Gribbin and McClean (1990) states that the maximum likelihood estimator(MLE) of t_{max} is t_k, so applying the invariance property of likelihood estimators stated below.

Theorem 1. *Invariance property of MLE: Suppose $X_1, X_2, X_3.., X_n$ is a random sample from a population with distribution function $f(x;\theta)$. Let $\hat{\theta}$ is the MLE of θ. If $\tau = u(\theta)$ is one to one function of θ, then for any function τ , the MLE of τ is $\widehat{\tau} = u\left(\hat{\theta}\right)$*

Proof. Let $L(\theta)$ be the likelihood function in terms of θ, that is

$$L(\theta) = \prod_{i=1}^{n} f(x_i;\theta) \tag{23}$$

Let $L*(\tau)$ be the likelihood function in terms of τ. To find $L*(\tau)$, write the distribution of X in terms of τ. Assume that $u : R \to R$ is a one-to-one function, that is, if $\tau = u(\theta)$, then $\theta = u^{-1}(\tau)$.

Therefore $f(x;\theta) = f(x;u^{-1}(\tau))$. Hence

$$\begin{aligned} L(\theta) =& \prod_{i=1}^{n} f(x_i;\theta) \\ =& \prod_{i=1}^{n} f(x_i;u^{-1}(\tau)) \\ =& L(u^{-1}(\tau)) \ = L*(\tau) \end{aligned} \tag{24}$$

The maximum value of L is obtained when $\theta = \hat{\theta}$ and in terms of τ when $\tau = \widehat{\tau}$ .Because of the relationship in (24) the maximums, which must be the same on both sides of (24), it is concluded that $\widehat{\theta} = u^{-1}(\widehat{\tau})$ or $\widehat{\tau} = u\left(\widehat{\theta}\right)$ □

, the maximum likelihood estimator for α is

$$\widehat{\alpha} = \widehat{S}(t_k) \tag{25}$$

Using the above estimators, the estimator of the survival function in (12) reduces to [4][5]:

$$\widehat{S}(t) = \left\{ \begin{array}{cc} 1 & t < \beta \\ \widehat{S}(t_k) + \left\{\widehat{S}(\widehat{\beta}) - \widehat{S}(t_k)\right\} \prod\limits_{j \epsilon R(t)'} \left(1 - \frac{d_i}{n_i}\right) & t \geqslant \beta \end{array} \right\} \tag{26}$$

2.3.2 Semi-parametric method

The model is based on a parametric regression model, it does not specify an assumptions regarding the probability distribution on a parametric regression model. [1]

Cox Regression

Cox regression is a method used to investigate the effect of several variables upon the time a specific event happened. It builds a model that predicts time-to-event data which results in a survival function that estimates the probability of an event occurring at any time t. It is regarded as a semi-parametric because it does not assume any probability distribution even though it is based on a parametric regression model [14].

> Cox writes : Suppose then that $h_0(t)$ is arbitrary. No information can be contributed about β by time intervals in which no failures occur because the components $h_0(t)$ might be conceivably be identically zero in such intervals. We therefore argue conditionally on the sets of instants at which failures occur; in discrete time we shall condition also on the observed multiplicities.

Once we require a method of analysis holding for all $h_0(t)$, consideration of this conditional distribution seems inevitable. [6][9]

This suggests that Cox's partial likelihood should be regarded as an ordinary likelihood function, with which to find the maximum likelihood estimate, the score statistic and sample information matrix must be used. [6]

Let $T_1, T_2, T_3, ..., T_n$ and $C_1, C_2, C_3, ..., C_n$ be independent variables. As mentioned above, C_i is the censoring time corresponding with the survival time T_i. $(Y_1, \delta_1), (Y_2, \delta_2), ..., (Y_n, \delta_n)$ are observed where [9]

$$Y_i = max(T_i, C_i) \tag{27}$$

$$\delta_i = I(T_i \leq C_i) \tag{28}$$

as well as $x_1,\ x_2,\ ...,\ x_n$. [9] Note the following vector notation of independent variable, also known as covariates associated with the dependent variable T_i :

$$\underset{\sim}{x}_i = (x_{i1},\ x_{i2},\ ...,\ x_{ip})' \tag{29}$$

Recall the following hazard function:

$$h(t;x) = \frac{f(t;x)}{1 - F(t;x)} \tag{30}$$

which clearly shows dependence of the distribution of T on the vector $\underset{\sim}{x}$. The proportional hazards model assume that

$$h(t;x) = h_0(t)\, e^{\beta' \underset{\sim}{x}} \tag{31}$$

where $\underset{\sim}{\beta} = (\beta_1,\ \beta_2, ...,\ \beta_n)'$ is the vector of regression coefficients and $h_0(t)$is the baseline hazard function [9] . Hazard rate is defined as the multiplication of constants with the function, $h_0(t)$. Note that this scalar depends on regression coefficients and the covariates, $x_i's$. The assumption in (33) is known as the baseline function since all covariates are the same [16][9], that is, $x_1 = x_2 = ... = x_n$.

Consider the ordered observation times below,

$$Y_{(1)}, Y_{(2)}, Y_{(3)}, ..., Y_{(n)} \tag{32}$$

and let δ_i be an indicator for censoring, and x_i is the covariate corresponding with $y_{(i)}$. Also, let

$$R_{(i)} = R(y_{(i)}-) \tag{33}$$

For all uncensored time y_i,

$$P\left(graduating \ \in \ (y_{(i)}, y_{(i)} + \triangle) \mid R_{(i)}\right) = \sum_{j \epsilon R_{(i)}} e^{\beta^{'} x_j} h_0\left(y_{(i)}\right) \triangle y \tag{34}$$

$$P(i \ graduating \ at \ time \ y_{(i)} \mid one \ graduation \ \in \ R_{(i)} \ at \ time \ \ y_{(i)}) = \frac{e^{\underset{\sim}{\beta}^{'} \underset{\sim}{x}_{(i)}}}{\sum\limits_{j \epsilon R_{(i)}} e^{\underset{\sim}{\beta}^{'} \underset{\sim}{x}_j}} \tag{35}$$

Taking the product of these conditional probabilities result in the conditional likelihood below:

$$L_C(\underset{\sim}{\beta}) = \prod_u \frac{e^{\underset{\sim}{\beta}^{'} \underset{\sim}{x}_{(i)}}}{\sum\limits_{j \epsilon R_{(i)}} e^{\underset{\sim}{\beta}^{'} \underset{\sim}{x}_j}} \tag{36}$$

From the quote above, Cox suggests his conditional likelihood as an ordinary likelihood, below is the score vector and the sample information matrix:

$$\frac{\partial}{\partial \underset{\sim}{\beta}} log \ L_c(\underset{\sim}{\beta}) = \left(\frac{\partial}{\partial \beta_1} log \ L_c(\underset{\sim}{\beta}), \ldots, \frac{\partial}{\partial \beta_p} log \ L_c(\underset{\sim}{\beta})\right)^{'} \tag{37}$$

$$\begin{aligned} \underset{\sim}{i}(\underset{\sim}{\beta}) &= -\frac{\partial^2}{\partial \underset{\sim}{\beta}^2} log \ L_c(\underset{\sim}{\beta}) \\ &= \begin{pmatrix} \frac{\partial^2}{\partial \beta_1 \partial \beta_1} log \ L_c(\underset{\sim}{\beta}) & \cdots & \frac{\partial^2}{\partial \beta_1 \partial \beta_p} log \ L_c(\underset{\sim}{\beta}) \\ \vdots & & \vdots \\ \frac{\partial^2}{\partial \beta_p \partial \beta_1} log \ L_c(\underset{\sim}{\beta}) & \cdots & \frac{\partial^2}{\partial \beta_p \partial \beta_p} log \ L_c(\underset{\sim}{\beta}) \end{pmatrix} \end{aligned} \tag{38}$$

required to solve

$$\frac{\partial}{\partial \underset{\sim}{\beta}} log \ L_c(\underset{\sim}{\beta}) = 0$$

which uses iterative methods.

This modeling method is recommended by most researchers mainly because of its powerful semi-parametric method of calculating survival probabilities and adjusting for any other dominant variables simultaneously [16]. Other interesting characteristics of the model include, creation of survival function estimates, the use of the partial likelihood function, no parametric assumptions and relative risk type measure of association [16].

2.3.3 Parametric method

Parametric models assume that the distribution of the hazard rate is given [14][10]. In this case, the hazard rate follows the specified function over time. Since estimating is based on the duration and the information of graduation, it makes the use of the data more efficient [10]. The common parametric models are, exponential, Weibull and log-normal models. This method will not be used in this research.

Weibull distribution

This type of distribution is a general form of the exponential distribution [10].

Assume $T \sim Weibull(\lambda,\ p)$, its probability density function is given as

$$f(t) = \lambda p t^{p-1} e^{-\lambda t^p} \tag{39}$$

, where $p > 0$ and $\lambda > 0$.

The hazard function is given by

$$h(t) = \lambda p t^{p-1} \tag{40}$$

Property of Weibull Model:

i)

$$S(t) = e^{-\lambda t^p}$$

$$-logS(t) = \lambda t^p$$

$$log(-logS(t)) = log(\lambda) + plog(t)$$

that is, $log(-logS(t))$ is linear with log of time.

3 Application

In this section, Kaplan-Meier and Cox regression methods are applied using SAS and output is interpreted accordingly.

3.1 Procedures

The SAS survival analysis procedures include ICLIFETEST, ICPHREG, LIFEREG, LIFETEST , PHREG, SURVEYPHREG. However for the purpose of this paper, only LIFETEST and PHREG will be used since they provide non-parametric estimates function by Kaplan-Meier method and regression analysis of sur-

vival data based on the Cox proportional hazards model respectively. The SAS codes used to illustrate the application of both methods, as well as some of the output is included in the appendix.

3.2 Data

Due to limited access of the University Of Pretoria's graduation data, a sample data below will be used to illustrate the methods discussed above. The data is discrete since a student can only graduate at the end of a semester. [14]

In this study, data of 50 students will be examined. It examines several factors, that is, gender, Student GPA and total financial aid per semester. Follow up time for student begins at the time a student started is enrolled for first semester. The variables used are:

- Duration: How many semester a student is registered for
- Semester: the semester a student is registered for
- Student
- Event: indicator variable, which is 1 if graduation took place and 0 if it didn't
- Graduate: The censoring variable, graduation took place=1, otherwise=0
- Gender: female, male
- Semester GPA
- Semester Total Aid

The dependent variable is time to an event which is the time at which the student got registered until graduation takes place or the study ended.

The data used is structured in the following way.

Student	Semester	Duration	Event	Graduate	Gender	Semester GPA	Semester Total Aid
1	1	2	0	0	Female	3.47	5500.00
1	2	2	0	0	Female	3.25	500.00
2	1	10	1	0	Female	2.54	3681.00
2	2	10	1	0	Female	2.95	1981.00
2	3	10	1	0	Female	3.22	2620.00
2	4	10	1	0	Female	2.00	2781.00
2	5	10	1	0	Female	1.97	500.00
2	6	10	1	0	Female	2.50	500.00
2	7	10	1	0	Female	1.44	0.00
2	8	10	1	0	Female	0.93	0.00
2	9	10	1	0	Female	2.50	0.00
2	10	10	1	0	Female	2.58	2500.00
3	1	8	0	0	Male	3.10	2500.00
3	2	8	0	0	Male	3.23	1200.00
3	3	8	0	0	Male	3.40	5697.00
3	4	8	0	0	Male	3.29	5698.00
3	5	8	0	0	Male	3.15	5697.00
3	6	8	0	0	Male	3.33	5698.00
3	7	8	0	0	Male	3.84	6642.00
3	8	8	0	0	Male	3.65	3128.00

Table 1: An extract of the data used for the research report

Below is the data exploration of this report. This is important in data analysis because the researcher familiarizes himself or herself with the distributions and typical values of each variable individually, as well as relationship between pairs or sets of variables. In SAS, proc univariate provides an easy look into distributions of variable separately while proc corr examines bivariate relationships.

Simple Statistics						
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
duration	271	7.57565	2.87917	2053	1.00000	10.00000
sem	271	4.28413	2.73870	1161	1.00000	10.00000
grad	271	0.04797	0.21410	13.00000	0	1.00000
gpa	271	2.98155	0.67743	808.00000	0.93000	3.84000
total	271	3505	2216	949903	0	6642

Table 2: Data exploration: Simple Statistics for data variables described above

Pearson Correlation Coefficients, N = 271 Prob > |r| under H0: Rho=0					
	duration	sem	grad	gpa	total
duration	1.00000	0.52827 <.0001	0.18936 0.0017	0.13210 0.0297	0.12609 0.0381
sem	0.52827 <.0001	1.00000	0.46936 <.0001	0.05827 0.3392	0.02819 0.6440
grad	0.18936 0.0017	0.46936 <.0001	1.00000	0.06894 0.2580	0.10643 0.0803
gpa	0.13210 0.0297	0.05827 0.3392	0.06894 0.2580	1.00000	0.67686 <.0001
total	0.12609 0.0381	0.02819 0.6440	0.10643 0.0803	0.67686 <.0001	1.00000

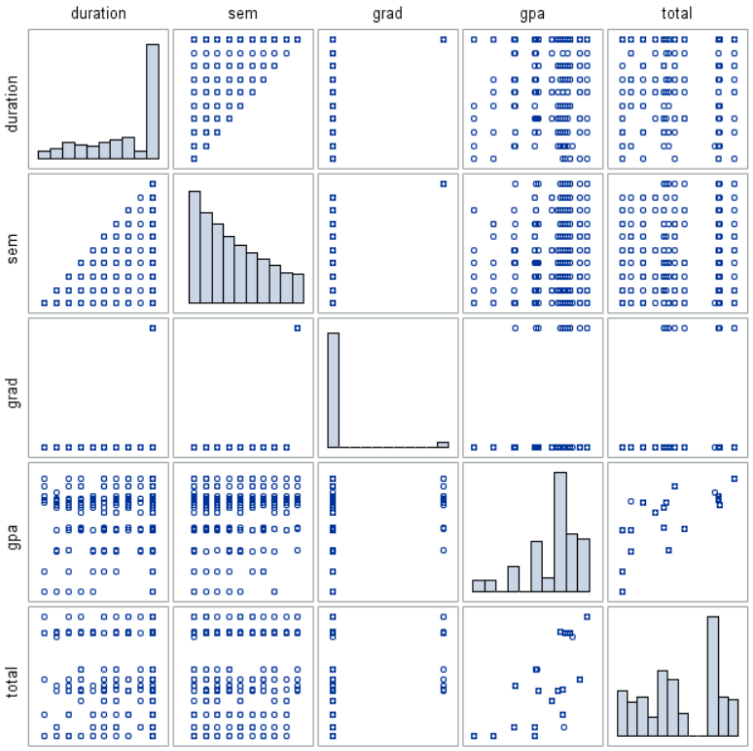


Figure 1: Data explorations of variables

The distribution generating the observed survival times is not known but proc univariate in SAS can be used to see how it looks like using non-parametric methods. In the table above, the mean of time to event data is 7.5 semesters, with graduation of 0.047 and GPA is 2.98. There is no correlation between the variables, therefore the variable vary in these data.

Figure 3 below shows the histogram for the graduation data set in Table 1.

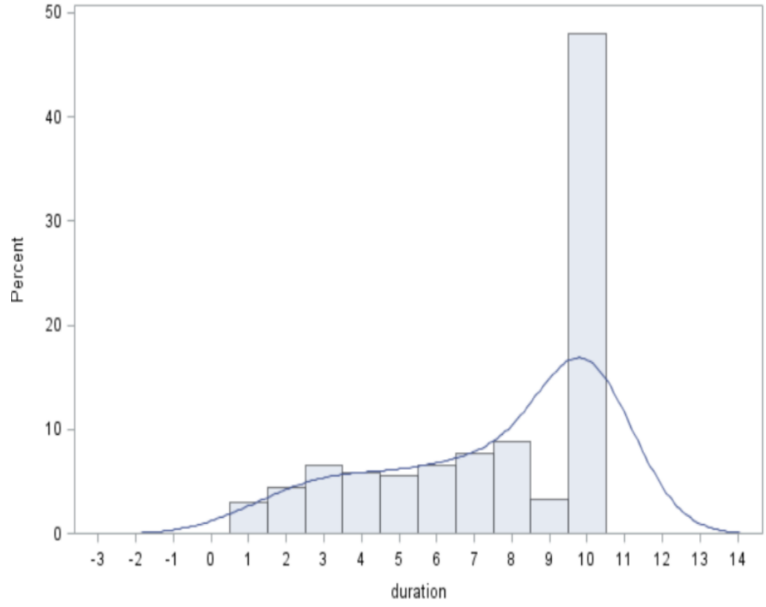


Figure 2: Probability density function and Histogram of duration

In the above figure, the correspondence between probability density functions(line graph) and histograms is visible. The graph is fitted and it is clear that our data is not normally distributed. This makes since in non-parametric modeling, the data is not required to fit a normal distribution. The graph is skewed to the left.

gender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Female	152	56.09	152	56.09
Male	119	43.91	271	100.00

Table 3: One way frequency tables for the variables, gender

Variable	N	Mean	Std Dev	Minimum	Maximum
duration	271	7.5756458	2.8791712	1.0000000	10.0000000
grad	271	0.0479705	0.2140992	0	1.0000000
gpa	271	2.9815498	0.6774275	0.9300000	3.8400000

Table 4: Summary statistics for the numerical variable, duration, GPA and grad

The average for the GPA is 2.98154 and the average duration a student is enrolled for is 7.57564

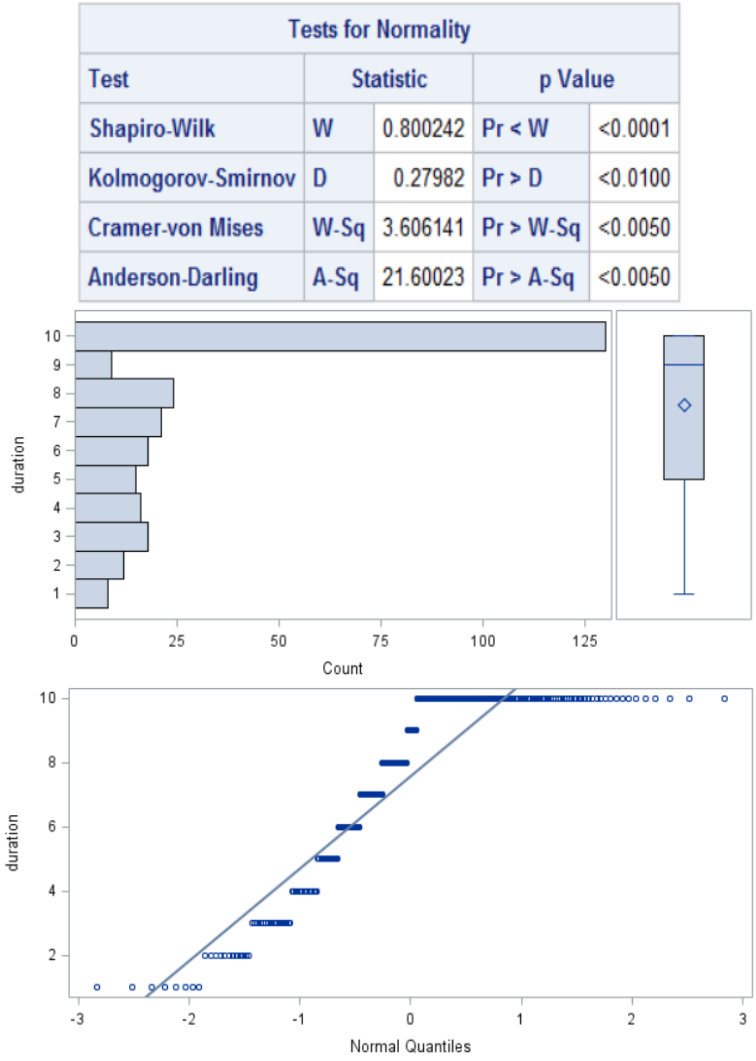


Figure 3: Extensive descriptive statistics for the variable duration

Test for normality:

H_0 : Data have a normal distribution

H_1 :Data does not have normal distribution

Since the p-value for the Shapiro-Wilk statistic is equal less than 0.0001, we reject the null hypothesis at 5% significance level. Therefore the distribution of the data does differ from the normal distribution.

These results correspond to those of Figure 2.

3.3 Application using the Kaplan-Meier method

The Kaplan-Meier estimator, or product limit estimator, is the estimator used by most software packages in light of its simple applicable approaches. [16]. The Kaplan-Meier estimator infuses data from all of the observations available, both censored and uncensored, by considering any point in time as a sequence of steps characterized by observed time[16]. In absence of censoring, the estimator is basically the sample

proportion of observations with event times greater than t [16].

Tables of Kaplan-Meier Estimates are obtained and interpreted below from the proc lifetest.

For the lifetest procedure, failure time variables is specified, in our case grad.rate.

SAS assumes that all times are censored.

The graph of the cumulative distribution function estimate is obtained below using proc univariate.

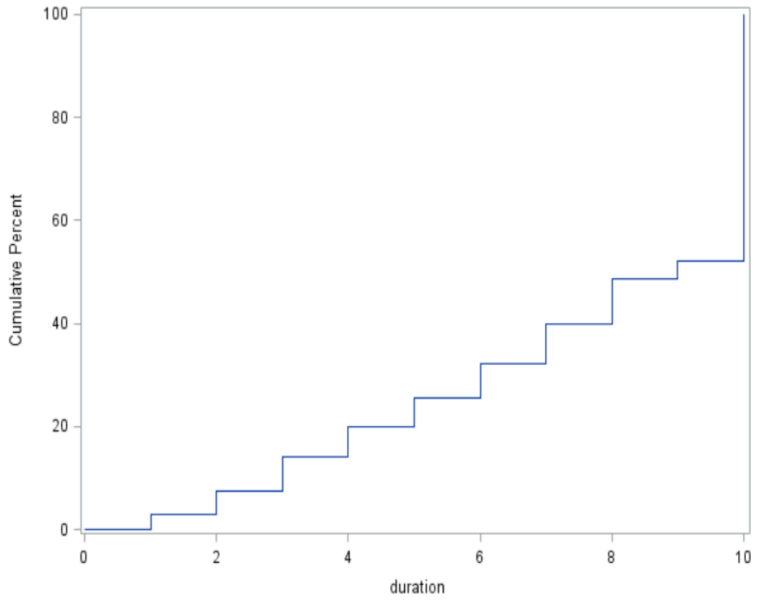


Figure 4: Cumulative distribution function of graduation data in Table 1

In Figure 3 above, the probability of completing half of the required years is way less than 50%. During time intervals where graduation is likely to take place, the c.d.f increases at a faster rate.

The survival function estimate of $S(t)$ is plotted below using the proc lifetest. This denotes the probability that graduation has not yet taken place by time t. In the figure below, at the beginning, where $t = 0$, the survival probability is 1, showing that it is certain that no one will graduation when they just started studies. As time goes by, for example, at $t = 4$, this probability is just below 0.8 that students have not graduated yet. At risk in the graph denotes set of student who are capable of experiencing graduation before t.

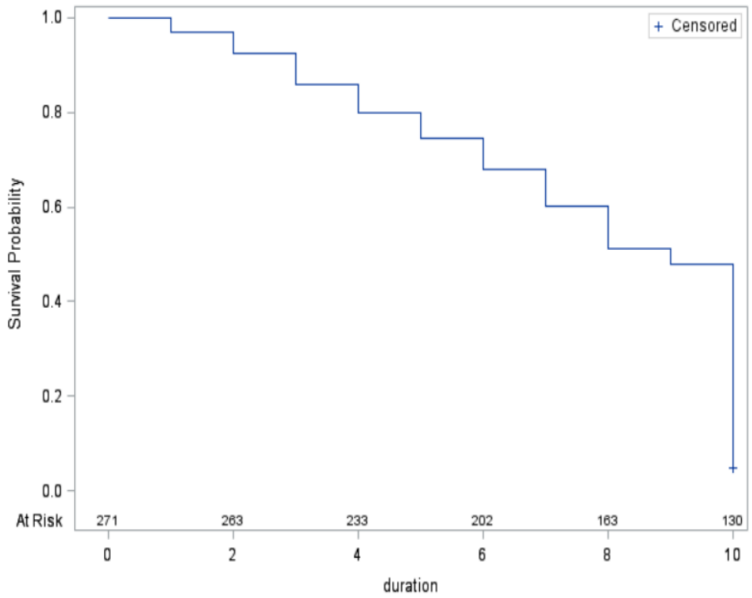


Figure 5: Survival function estimate of graduation data in Table 1

The hazard function can also be estimated using proc lifetest in SAS.

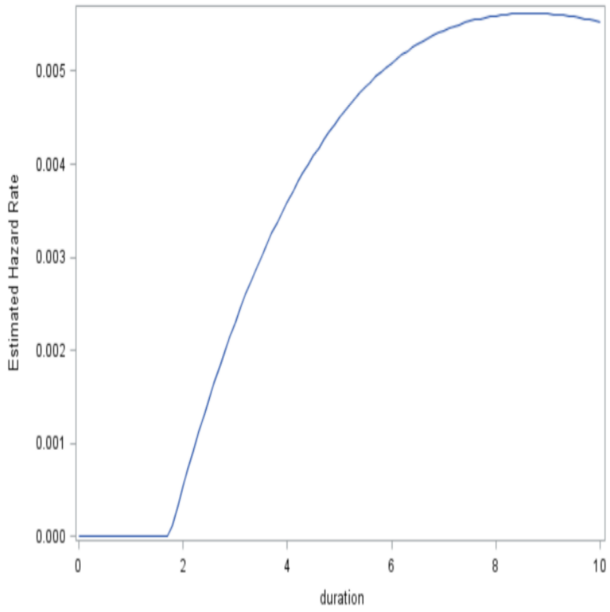


Figure 6: Hazard function of graduation data in Table 1

This is estimated using proc lifetest and then the results are sent to proc sgplot for plotting. The rate of graduation at any time t is shown in the figure above. It is see that this rate decreases gradually after the $8-th$ semester, that is, $t = 8$.

The cumulative hazard function is demonstrated below.

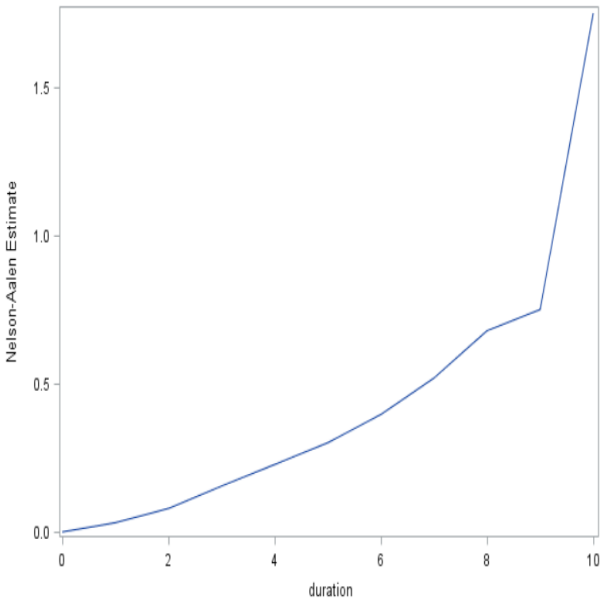


Figure 7: Cumulative hazard function of graduation data in Table 1

On the table below, the Kaplan-Meier estimates of the survival function are displayed.

Product-Limit Survival Estimates							
duration	Number at Risk	Observed Events	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.0000	271	0	1.0000	0	0	0	271
1.0000	.	.	.	.	.	1	270
1.0000	.	.	.	.	.	2	269
1.0000	.	.	.	.	.	3	268
1.0000	.	.	.	.	.	4	267
1.0000	.	.	.	.	.	5	266
1.0000	.	.	.	.	.	6	265
1.0000	.	.	.	.	.	7	264
1.0000	271	8	0.9705	0.0295	0.0103	8	263
2.0000	.	.	.	.	.	9	262
2.0000	.	.	.	.	.	10	261
2.0000	.	.	.	.	.	11	260
2.0000	.	.	.	.	.	12	259
2.0000	.	.	.	.	.	13	258
2.0000	.	.	.	.	.	14	257
2.0000	.	.	.	.	.	15	256
2.0000	.	.	.	.	.	16	255
2.0000	.	.	.	.	.	17	254
2.0000	.	.	.	.	.	18	253
2.0000	.	.	.	.	.	19	252
2.0000	263	12	0.9262	0.0738	0.0159	20	251

Table 5: The Kaplan-Meier survival estimates of graduation data in Table 1

Every row corresponds to a time interval in the "duration" column of that row.For the time interval in the first row is from 0 semesters to just before first semester. During this interval there is 271 students at risk, meaning who can graduate and none of the graduate as "Observed Events" equals 0 and the "survival" function estimate is 1. In the next interval, from semester 1 to just before the second semester 8 students graduated, this is shown in 8 rows of "duration"=1.00 and by "Observed Events"=8 in the last row when "duration"=1.00. Note that the probabilities in the Survival column are unconditional and are interpreted as the probability of graduating from the interval, follow up time until the semester number in the "duration" column.

Looking at later survival times in the table below:

10.0000		.	.	.	.	.	254	17
10.0000		.	.	.	.	.	255	16
10.0000		.	.	.	.	.	256	15
10.0000		.	.	.	.	.	257	14
10.0000		130	117	0.0480	0.9520	0.0130	258	13
10.0000	*	.	0	.	.	.	258	12
10.0000	*	.	0	.	.	.	258	11
10.0000	*	.	0	.	.	.	258	10
10.0000	*	.	0	.	.	.	258	9
10.0000	*	.	0	.	.	.	258	8
10.0000	*	.	0	.	.	.	258	7
10.0000	*	.	0	.	.	.	258	6
10.0000	*	.	0	.	.	.	258	5
10.0000	*	.	0	.	.	.	258	4
10.0000	*	.	0	.	.	.	258	3
10.0000	*	.	0	.	.	.	258	2
10.0000	*	.	0	.	.	.	258	1
10.0000	*	.	0	0.0480	0.9520	.	258	0

Table 6: Last survival estimates

From above, it is evident that there are a number of records where no graduation took place during "duration" = 10. The "*" indicates censored observations.

The graph of the Kaplan-Meier estimate is shown below.

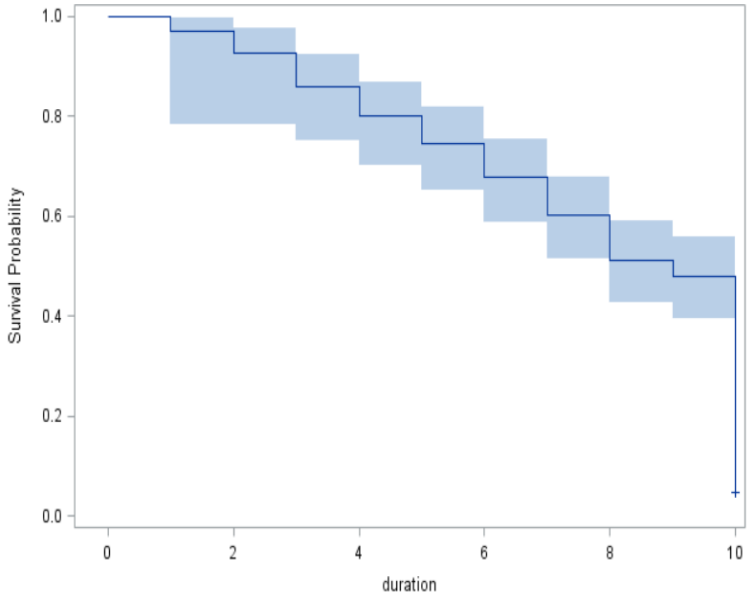


Figure 8: Kaplan-Meier curve for graduation data in Table 1

The step graph is the Kaplan-Meier graph and the shading around it is the 95% confidence bands. When a student graduate at a point in time, the step functions drops and in between these times the graphs stays constant. The survival function drop is constant over time. The survival function will never reach zero, instead it will remain at the survival probability estimated at the previous interval, which is 10 in this case. The 95% confidence interval is calculated for the whole survival function.

Suppose that there may be difference in the survival functions among some of the groups in the study. The test for equality in of these survival functions can be performed using the non-parametric method in the following manner:

Test	Chi-Square	DF	Pr > Chi-Square
Log-Rank	11.2036	1	0.0008
Wilcoxon	6.9545	1	0.0084
-2Log(LR)	0.6784	1	0.4101

Table 7: Testing equality over the strata, gender

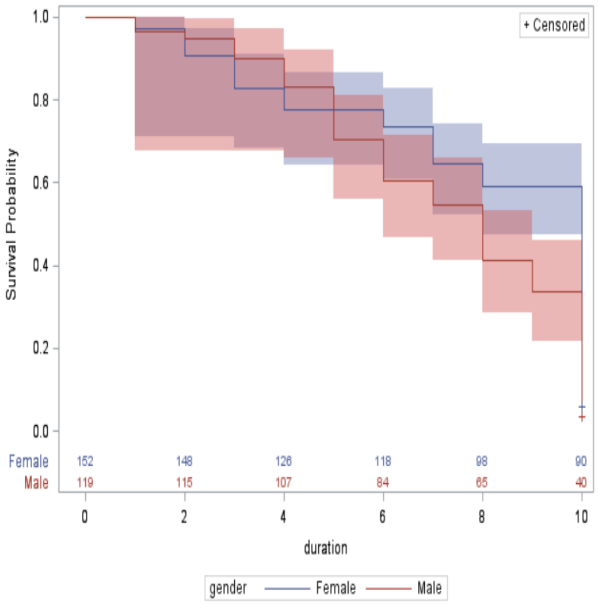


Figure 9: Non-parametric tests among the gender group

In the Kaplan-Meier estimator graph above, which is stratified by gender, it appears that, in general, males have the worst experience is survival. This means that females in the graduation data provided in Table 1, are more likely to graduate as compared to males and this is true since looking in the data in the table, females graduate more over the study period.

This is reinforced in the test of equality. Using Figure 13 to test the equality among the gender strata:

H_0 : There is equality among the gender group or $S_{female}(t) = S_{male}(t)$

H_1 : There is no equality among the gender group or $S_{female}(t) \neq S_{male}(t)$

All the p-values are less than 5% hence the null hypothesis that there is equality among the gender group is rejected.

3.4 Application using Cox regression

A simple model in proc phreg is demonstrated below. The effects of categorical variable, gender, and a continuous variable gate on the hazard rate. Specifying gender as categorical, it is entered on the class statement, on the model statement, left hand side of the equation is the time variable, duration and on the right side is grad, the censoring variable with the censoring value in brackets.

Model Fit Statistics		
Criterion	Without Covariates	With Covariates
-2 LOG L	126.556	125.817
AIC	126.556	129.817
SBC	126.556	130.947

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	0.7391	2	0.6911
Score	0.6601	2	0.7189
Wald	0.6550	2	0.7207

Type 3 Tests			
Effect	DF	Wald Chi-Square	Pr > ChiSq
gender	1	0.0478	0.8269
gpa	1	0.6528	0.4191

Analysis of Maximum Likelihood Estimates								
Parameter		DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio	Label
gender	Female	1	0.13450	0.61505	0.0478	0.8269	1.144	gender Female
gpa		1	0.42956	0.53168	0.6528	0.4191	1.537	

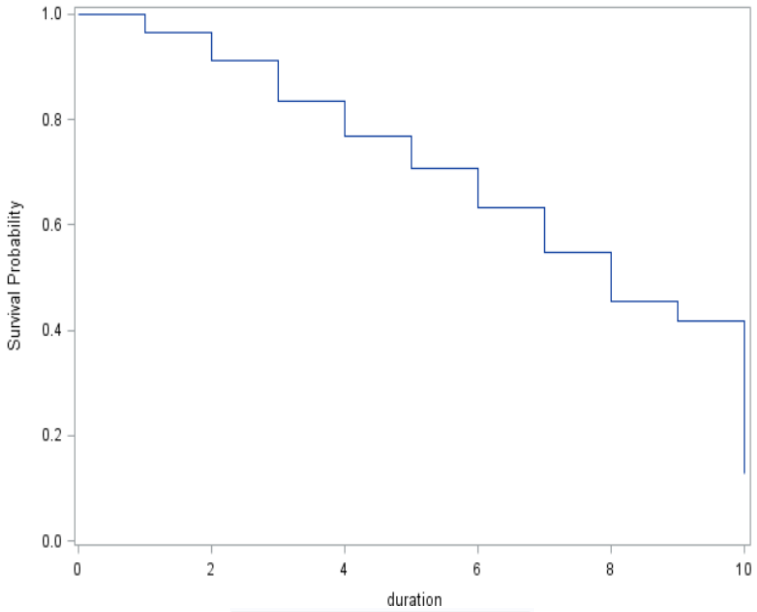
Table 8: A simple Cox regression model

Model fit statistics: Displays fit statistics which are used to compare and select models. This is the first model so there is no other model to compare it with.

Testing Global Hypothesis: BETA=0: The hypothesis that all coefficients in the model are equal to 0 is displayed here. This tests whether the model can predict the changes in the hazard rate. the likelihood ratio test is preferred for small samples. The p-value is $0.6911 > 5\%$ therefore we do not reject the hypothesis. It appears that all regression coefficients are zero.

Analysis of Maximum Likelihood Estimates: Model coefficients, tests of significance, and exponential coefficient as hazard ration are displayed. It seems that females have $13,45\%$ increase in the hazard rate compared to males while the GPA, the hazard rate increase by 43% . There is no intercept since in Cox regression, it is held into the baseline hazard function, which is not specified.

The survival and baseline hazard function after Cox regression looks like this:



Reference Set of Covariates for Plotting	
gpa	gender
2.981550	Male

Figure 10: Survival function after fitting Cox regression model

In this model, the curve is for males with GPA of 2.98155 The survival function does not differ from the one above in any way.

4 Conclusion

The main purpose of this paper was to use two survival methods,namely, Kaplan-Meier and Cox regression to study the university rates of undergraduates in Department of Statistics, University of Pretoria. Due to limited access of the university's student database, graduation sample of 50 students were used in order to show the application of the two survival methods. According to the data and after fitting the model, the probability of graduation decreases with time. Female students are more likely to graduate as compared to male students.

This study's limitation is the fact that data was not accessible, which would have made it relatable and the sample size would have been reasonably more to work with. The Cox regression partial likelihood could be investigated further so that the exact estimators of the parameter is explicitly stated theoretically. The study can also be enhanced by getting the exact estimates of parameters using a more realistic dataset.

References

[1] Paul D Allison. Survival analysis of backward recurrence times. *Journal of the American Statistical Association*, 80(390):315–322, 1985.

[2] Manish Goel, Pardeep Khanna, and Jugal Kishore. Understanding survival analysis: Kaplan-Meier estimate. *International Journal of Ayurveda Research*, 1(4):274, 2010.

[3] Gregory R Hancock, Ralph O Mueller, and Laura M Stapleton. *The Reviewerâs Guide to Quantitative Sethods in the Social Sciences*. Routledge, 2010.

[4] Aglaia G Kalamatianou and Sally McClean. The perpetual student: Modeling duration of undergraduate studies based on lifetime-type educational data. *Lifetime Data Analysis*, 9(4):311–330, 2003.

[5] Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481, 1958.

[6] Chap T Le. *Applied Survival Analysis*. Wiley, 1997.

[7] Moeketsi Letseka and Simeon Maile. *High university drop-out rates: A threat to South Africa's Future*. Human Sciences Research Council Pretoria, 2008.

[8] Micha Mandel. Censoring and truncationâhighlighting the differences. *The American Statistician*, 61(4):321–324, 2007.

[9] Rupert G Miller Jr. *Survival Analysis*, volume 66. John Wiley & Sons, 2011.

[10] Maryam Montaseri, Jamshid Yazdani Charati, and Fateme Espahbodi. Application of parametric models to a survival analysis of hemodialysis patients. *Nephro-urology Monthly*, 8(6), 2016.

[11] Mike Murray. Factors affecting graduation and student dropout rates at the University of Kwazulu-Natal. *South African Journal of Science*, 110(11-12):01–06, 2014.

[12] Mohamad Amin Pourhoseingholi, Ebrahim Hajizadeh, Bijan Moghimi Dehkordi, Azadeh Safaee, Alireza Abadi, and Mohammad Reza Zali. Comparing Cox regression and parametric models for survival of patients with gastric carcinoma. *Asian Pacific Journal of Cancer Prevention*, 8(3):412, 2007.

[13] Germán Rodrıguez. Non-parametric estimation in survival models. 2005.

[14] Christopher Rooney. *Using survival analysis to identify the determinants of academic exclusion and graduation in three faculties at UCT*. PhD thesis, University of Cape Town, 2015.

[15] SAS Seminar. Introduction to survival analysis in SAS. UCLA: Statistical consulting group, 2015.

[16] T Smith and B Smith. Kaplan Meier and Cox Proportional hazards modeling: Hands on Survival Analysis. In *Workshop research paper*, 2003.

[17] Jun Xie and Chaofeng Liu. Adjusted Kaplan–Meier estimator and log-rank test with inverse probability of treatment weighting for survival data. *Statistics in Medicine*, 24(20):3089–3110, 2005.

[18] Stanley Xu, Susan Shetterly, David Powers, Marsha A Raebel, Thomas T Tsai, P Michael Ho, and David Magid. Extension of Kaplan-Meier methods in observational studies with time-varying treatment. *Value in Health*, 15(1):167–174, 2012.

Appendix

```
data d2;
input student sem duration event grad gender $ gpa total @@;
cards ;
1 1 2 0 0 Female 3.47 5500
1 2 2 0 0 Female 3.25 500
2 1 10 1 0 Female 2.54 3681
2 2 10 1 0 Female 2.95 1981
2 3 10 1 0 Female 3.22 2620
2 4 10 1 0 Female 2.00 2781
2 5 10 1 0 Female 1.97 500
2 6 10 1 0 Female 2.50 500
2 7 10 1 0 Female 1.44 0
2 8 10 1 0 Female 0.93 0
2 9 10 1 0 Female 2.50 0
2 10 10 1 1 Female 2.58 2500
3 1 8 0 0 Male 3.10 2500
3 2 8 0 0 Male 3.23 1200
3 3 8 0 0 Male 3.40 ....
2 4 0 0 Male 3.65 3128 50 3 4 0 0 Male 2.54 3681 50 4 4 0 0 Male 2.55 3681 ;
/*The probability density function*/
proc univariate data = d2 /*(where=(grad=0))*/;
var duration;
histogram duration / kernel;
run;
proc freq data=d2;
tables gender;
run;
proc means data=d2;
var duration  grad gpa;
run;
proc univariate normal plot data=d2;
var duration ;
run;
```

```
/*The cumulative function*/
proc univariate data = d2 /*(where=(grad=0))*/;
var duration;
cdfplot duration;
run;
/*the survival function*/ proc lifetest data=d2/*(where=(grad=0))*/ plots=survival(atrisk);
time duration*grad(1);
run;
/*The hazard function*/
proc lifetest data=d2 /*(where=(grad=1))*/ plots=hazard(bw=15); time duration*grad(0);
run;
/*The cumulative hazard function*/ ods output ProductLimitEstimates = ple;
proc lifetest data=d2(where=(grad=0)) nelson outs=outd2;
time duration*grad(1);
run;
proc sgplot data = ple;
series x = duration y = CumHaz;
run;
/*Exploring Data*/
proc corr data = d2 plots(maxpoints=none)=matrix(histogram);
var duration sem grad gpa total;
run;
/*Obtaining and interpreting tables of KM*/
proc lifetest data=d2 atrisk outs=outd2;
time duration*grad(1);
run;
/*Graphing the KM estimates*/
proc lifetest data=d2 atrisk plots=survival(cb) outs=outd2;
time duration*grad(1);
run;
/*Comparing survival functions using nonparametric tests*/
proc lifetest data=d2 atrisk plots=survival(atrisk cb) outs=outd2; strata gender;
time duration*grad(1);
run;
```

```
/*Fitting a simple Cox regression model*/
proc phreg data = d2;
class gender;
model duration*grad(0) = gender gpa;;
run;
/*Producing graphs of the survival and baseline hazard function after Cox regression*/
proc phreg data=d2 plots=survival;
class gender;
model duration*grad(1) = gender gpa;;
run;
```

Simultaneous equation methods

Cheyeza Mabuza 13087097

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr. J Kleyn

Department of Statistics, University of Pretoria

31 October 2017

Abstract

In this essay we will consider simultaneous equation estimation. In the context of single-equation models the dependent variable (Y), is dependent on X variables known as explanatory variables. In practice there are situations where such unidirectional relation between X and Y variables cannot be maintained, as it may be possible that the explanatory variable not only affect the dependent but the dependent variable can also be an explanatory variable in a system of equations and this is the reason why it is better to have the variables estimated simultaneously. The estimation of such variables is done in simultaneous equation methods.

The goal of this essay is to understand what simultaneous equation methods are and this will be done by considering the identification problem: underidentification, just or exact identification and overidentification. The differences between estimation of a just identified equation with indirect least square (ILS) and method of two stage least squares (2SLS) which is used to estimate parameters of an overidentified equation will be illustrated. Practical applications will the used to illustrate the different estimation techniques using the available SAS procedures.

Declaration

I, *Cheyeza Antoinette Mabuza,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Cheyeza Antoinette Mabuza

_ _

Dr. Judy Kleyn

_ _

Date

Acknowledgments

Firstly I would like to thank the SARChI bursary for the financial support, secondly I would also like to give special thanks to my supervisor Dr. Judy Kleyn for mentorship and constantly steering me into the right direction for my research and lastly I would like to my family and friends for their support and unconditional love.

Contents

List of Figures

List of Tables

1 Introduction

Simultaneous equation models are statistical models that are in form of a set of simultaneous equation systems.

Consider the following M equations model:

$$Y_{1t} = \beta_{12}Y_{2t} + \beta_{13}Y_{3t} + ... + \beta_{1M}Y_{Mt} + \gamma_{11}X_{1t} + \gamma_{12}X_{2t} + ... + \gamma_{1k}X_{kt} + \mu_{1t}$$

$$Y_{2t} = \beta_{21}Y_{1t} + \beta_{23}Y_{3t} + ... + \beta_{1M}Y_{Mt} + \gamma_{21}X_{1t} + \gamma_{22}X_{2t} + ... + \gamma_{2k}X_{kt} + \mu_{2t}$$

$$Y_{3t} = \beta_{31}Y_{1t} + \beta_{32}Y_{2t} + ... + \beta_{3M}Y_{Mt} + \gamma_{31}X_{1t} + \gamma_{32}X_{2t} + ... + \gamma_{3k}X_{kt} + \mu_{3t}$$

.

.

.

$$Y_{MT} = \beta_{M1}Y_{1t} + \beta_{M2}Y_{2t} + ... + \beta_{M,M-1}Y_{M-1,t} + \gamma_{M1}X_{1t} + \gamma_{M2}X_{2t} + ... + \gamma_{Mk}X_{kt} + \mu_{Mt}$$

where $Y_1, Y_2, ..., Y_M = M$ endogenous variables

$X_1, X_2, ..., X_k = K$ predetermined variables

$\mu_1, \mu_2, ..\mu_M = M$ stochastic disturbances

$t = 1, 2, ...T =$ total number of observations

$\beta's =$ coefficients of the endogenous variables

$\alpha's =$ coefficients of the predetermined variables

This report will make references to the above model as specified in [6] to facilitate the discussion around simultaneous equation methods.

A simultaneous equation system is a system of two or more equations where a variable explained in one equation can be written as an explanatory variable in another equation of the same model. In simultaneous equation models, the term endogenous variables denotes jointly dependent variables. The endogenous value is determined within the model whereas the variables which are truly non stochastic are called exogenous or predetermined variables since their variables are determined outside the model [6].

In a simultaneous system of equations certain equations are estimable while others are not, therefore identification problem should be considered. The identification problem looks at whether estimates of parameters from a structural equation can be obtained and if the estimates can be obtained, then that particular equation is identified; however, if the estimates of the parameters cannot be obtained then that particular equation is underidentified [6]. This essay will illustrate the different ways to determine whether a certain simultaneous

equation model consists of identified equations or not through a thorough discussion on the different methods of identification as well as the use of identification rules. The essay will also look at methods that can be used to estimate the parameters of simultaneous equation models and consider their merits as well as their limitations.

As mentioned before the endogenous variables in simultaneous equation models may appear as an explanatory equation in other equation of the same system, hence ordinary least squares (OLS) method of estimating variables may not used to estimate variables and for this reason alternative methods have been developed and can be used for estimating such simultaneous equations.

The relationship between variables in econometrics is often a single-type, hence a single-type equations exists between variables in economic relations [11]. A model representing such relationships is one where the dependent variable (Y) is expressed as a linear function of the explanatory variables (X's). The key assumption in such a model is that a cause-and-effect relationship exists between the variables where the explanatory variable is the cause and the dependent variable is the effect. Although, in practice there often exist a two-way relationship among economic variables; where one economic variable affects other economic variables. Simultaneous equation models will be considered, models where one dependent variable can also be a explanatory variable in a system of equations[10].

Today's econometrics is mainly influenced by the Cowles Commission, the commission consists of a team of econometricians and economists, the team did most of their work at the University of Chicago [1]. Haavelmo's work discusses how one can use probability approach when formulating models that are econometric in nature [1]. Koopmans, Mesharck and Hood provided the appropriate statistical ways of dealing with simultaneous equation models, their work was highly influential in the Cowles Commission [1].

In simultaneous equation models, the **full information maximum likelihood (FIML) method,** a system method, can be used to estimated parameters of equations within that particular model**.** However, in practice this type of method is not commonly used [11]. There are several reasons why this method is not used, one is that the computational burden for such a method is too high. Another reason why system methods such as FIML is not widely used in practice is that this method leads to solutions that are nonlinear and therefore making it very difficult to determine those solutions and also if there is a specification error in one or more equation then that error will be carried through to the entire system [9]. Since, system methods are easily affected by specification errors, for this reason simultaneous equation methods are popularly used .

To assess whether a structural equation is identified, the structural equation should be written as a re-

duced form equation, where the endogenous variable is expressed as a function of the exogenous variables. The process of writing structural equations as reduced form equations is time consuming but it is avoidable by applying rules of identification as discussed in Gujarati [6] namely; **The order condition of identifiability** and **the rank condition of identifiability.** If an equation in a simultaneous equation model is identified either exactly identified or overidentified, then different methods of estimating the parameters of such equations are considered. The methods of estimation fall under two categories namely; the system method as well as single-type methods [5]. Shreya et al [16] discusses single-equation methods these are methods which are usually used in practice, in other words; OLS, ILS and 2SLS. The OLS method is usually not the best method to use in practice when estimating parameters of simultaneous equations. The ILS method is usually the best method to use to estimate parameters of just or exactly identified equations, in ILS method OLS procedures are applied to the reduced-form equation and one can estimate the original structural coefficients from the reduced form coefficients. The 2SLS method is specifically used to estimate the parameters of overidentified equations.

The two-stage least square method of linear estimation of coefficients was developed by Theil. Basmann independently developed a similar solution under the name of the generalized classical method of linear estimation which leads to equivalent estimators [10]. Several studies have been conducted more recently on the effectiveness of the method and its limitations. The two-stage least square method was developed to replace existing methods (indirect least squares, least variance ratio, or limited-information single equation) by providing a method of more general applicability while being less expensive to apply. In the first stage of a two-stage least squares solution ordinary least squares methodology is applied to the entire system of predetermined variables to obtain estimates for the endogenous variables [14]. In the second stage, these estimates are substituted in the system and ordinary least squares is applied again to a particular equation or a set of equations of the system to estimate the required parameters. The method is not only relatively fast in terms of time required for calculations but the estimates derived can be shown to be asymptotically unbiased, consistent, and have minimum variance [11].

2 Background Theory

The previous section introduced the topic of simultaneous equation methods and its features, now in order to carry out the analysis of different estimation methods used to estimate simultaneous equation models one has to first understand the identification problem. This section will deal with an extensive discussion of the identification problem as well as the two methods used to estimate simultaneous equation models, namely;

ILS and 2SLS.

2.1 Notation and Terminology

Some notation and terminology that will be used in this report is listed as follows:

- **Endogenous variable:** jointly dependent variable determined within a model (stochastic), typically denoted by Y_{it} in the general-M model.
- **Exogenous or predetermined variable:** a non stochastic variable whose value is determined outside a model, typically denoted by X_{it} in the general-M model.
- **Stochastic disturbance:** random variable, typically denoted by μ_{it} the general-M model.
- **Reduced form equation:** a mathematical equation where an endogenous variable is expressed as a function of the exogenous variables and its random (stochastic) variables.

2.2 The Identification Problem

The identification problem in econometrics and statistics context is the inability to obtain estimates of the parameters of the reduced form equation, in other words the identification problem addresses the problem of whether the parameters of a particular reduced form equation can be estimated or not [6]. If the parameters of the reduced form equation can be estimated, then that equation is said to be identified; however, if it can not be estimated then the equation is said to be underidentified or unidentified.

Two types of identified equation exist; namely, exactly (just or fully) and overidentified. An exactly identified equation is one where numerical estimates of the reduced form equation can be obtained and an overidentified equation is one where more than one numerical estimate can be obtained for the structural parameters.

Underidentification

To facilitate our discussion of underidentification, consider the following demand-and-supply model as specified in Gujarati [6].

Demand function:

$$Q_t^d = \alpha_0 + \alpha_1 P_t + \mu_{1t} \qquad \alpha_1 < 0 \tag{1}$$

Supply function:

$$Q_t^s = \beta_0 + \beta_1 P_t + \mu_{2t} \qquad \beta_1 < 0 \tag{2}$$

using the equilibrium condition:

$$Q_t^d = Q_t^s \tag{3}$$

where Q^d = quantity demanded

Q^s= quantity supplied

t = time

P= price

the equilibrium condition gives the following

$$\alpha_0 + \alpha_1 P_t + \mu_{1t} = \beta_0 + \beta_1 P_t + \mu_{2t} \tag{4}$$

solving equation (4)

$$\begin{aligned} (\alpha_1 - \beta_1)P_t &= (\beta_0 - \alpha_0) + (\mu_{2t} - \mu_{1t}) \\ P_t &= \frac{\beta_0 - \alpha_0}{\alpha_1 - \beta_1} + \frac{\mu_{2t} - \mu_{1t}}{\alpha_1 - \beta_1} \end{aligned}$$

So we get

$$P_t = \Pi_0 + v_t \tag{5}$$

where

$$\Pi_0 = \frac{\beta_0 - \alpha_0}{\alpha_1 - \beta_1} \tag{6}$$

$$v_t = \frac{\mu_{2t} - \mu_{1t}}{\alpha_1 - \beta_1} \tag{7}$$

Now substituting (5) into (1) or (2)

$$
\begin{aligned}
Q_t &= \alpha_0 + \alpha_1(\Pi_0 + v_t) + \mu_{1t} \\
&= \alpha_0 + \alpha_1(\frac{\beta_0 - \alpha_0}{\alpha_1 - \beta_1} + \frac{\mu_{2t} - \mu_{1t}}{\alpha_1 - \beta_1}) + \mu_{1t} \\
&= \alpha_0 + \frac{\alpha_1\beta_0 - \alpha_1\alpha_0}{\alpha_1 - \beta_1} + \frac{\alpha_1\mu_{2t} - \alpha_1\mu_{1t}}{\alpha_1 - \beta_1} + \mu_{1t} \\
&= \frac{\alpha_0(\alpha_1 - \beta_1) + \alpha_1\beta_0 - \alpha_1\alpha_0 + \alpha_1\mu_{2t} - \alpha_1\mu_{1t} + \mu_{1t}(\alpha_1 - \beta_1)}{\alpha_1 - \beta_1} \\
&= \frac{\alpha_0\alpha_1 - \alpha_0\beta_1 + \alpha_1\beta_0 - \alpha_1\alpha_0 + \alpha_1\mu_{2t} - \alpha_1\mu_{1t} + \mu_{1t}\alpha_1 - \mu_{1t}\beta_1}{\alpha_1 - \beta_1} \\
&= \frac{-\alpha_0\beta_1 + \alpha_1\beta_0 + \alpha_1\mu_{2t} - \mu_{1t}\beta_1}{\alpha_1 - \beta_1} \\
&= \frac{\alpha_1\beta_0 - \alpha_0\beta_1}{\alpha_1 - \beta_1} + \frac{\alpha_1\mu_{2t} - \beta_1\mu_{1t}}{\alpha_1 - \beta_1}
\end{aligned}
$$

the following equilibrium quantity is obtained:

$$Q_t = \Pi_1 + w_t \tag{8}$$

where

$$\Pi_1 = \frac{\alpha_1\beta_0 - \alpha_0\beta_1}{\alpha_1 - \beta_1} \tag{9}$$

$$w_t = \frac{\alpha_1\mu_{2t} - \beta_1\mu_{1t}}{\alpha_1 - \beta_1} \tag{10}$$

Note that equations (5) and (8) are reduced form equations, the demand-and-supply model has the following structural coefficients $(\alpha_0, \alpha_1, \beta_0, \beta_1)$ that need to be estimated, however, these structural coefficients can not be estimated uniquely. This is due to the fact that the equations (5) and (8) contains all four structural coefficients and mathematically it is not possible to estimate four structural unknowns from two known reduced form equations. This implies that given P (price) and Q(quantity) one cannot guarantee that the estimated parameters are for the demand or supply function as it is not possible to estimate the four structural coefficients.

Just or Exact Identification

The demand or supply function could not be identified in the previous section as similar variables P and Q are present in both the functions and no additional information is given, hence underidentified. Consider the following demand-and-supply model where an additional variable is added as specified in Gujarati [6]:

Demand function:

$$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 I_t + \mu_{1t} \qquad \alpha_1 < 0,\, \alpha_2 > 0 \tag{11}$$

Supply function:

$$Q_t = \beta_0 + \beta_1 P_t + \mu_{2t} \qquad \beta_1 < 0 \tag{12}$$

Where I is the consumer income, which is a predetermined (or exogenous variable) and all the other variables are defined the same way as before.

Notice that the only difference in the models is that an additional variable I is added to the demand function, it is known from economic theory that consumer income plays a vital role when it comes to the demand of goods and services. Thus, including consumer income in the demand function gives additional and important information about consumer behaviour.

Again the using the equilibrium equation:

$$\alpha_0 + \alpha_1 P_t + \alpha_2 I_t + \mu_{1t} = \beta_0 + \beta_1 P_t + \mu_{2t} \tag{13}$$

Now solving equation (13) we get the following equilibrium point for P_t (refer to the appendix for calculations)

$$P_t = \Pi_0 + \Pi_1 I_t + v_t \tag{14}$$

where

$$\Pi_0 = \frac{\beta_0 - \alpha_0}{\alpha_1 - \beta_1} \tag{15}$$

$$\Pi_1 = -\frac{\alpha_2}{\alpha_1 - \beta_1} \tag{16}$$

$$v_t = \frac{\mu_{2t} - \mu_{1t}}{\alpha_1 - \beta_1} \tag{17}$$

Now substituting (14) into (11) or (12) , we obtain the following equilibrium quantity (refer to the appendix for calculations):

$$Q_t = \Pi_2 + \Pi_3 I_t + w_t \tag{18}$$

where

$$\Pi_2 = \frac{\alpha_1\beta_0 - \alpha_0\beta_1}{\alpha_1 - \beta_1} \tag{19}$$

$$\Pi_3 = -\frac{\alpha_2\beta_1}{\alpha_1 - \beta_1} \tag{20}$$

$$w_t = \frac{\alpha_1\mu_{2t} - \beta_1\mu_{1t}}{\alpha_1 - \beta_1} \tag{21}$$

Notice that both equations (14) and (18) are reduced form equations and so the OLS method can be used to estimate their parameters. Since the demand-and-supply model has the following structural coefficients $(\alpha_0, \alpha_1, \alpha_2, \beta_0, \beta_1)$ that need to be estimated and there are only four reduced form coefficients $(\Pi_0, \Pi_1, \Pi_2, \Pi_3)$ to estimate them. It is not mathematically possible to have unique estimates for all structural coefficients. However, the parameters of the supply function are identifiable. Parameters of the demand function cannot be estimated as there is no unique way of estimating them and hence the demand function is unidentified. As mentioned parameters of the supply function are identifiable or estimable since

$$\beta_0 = \Pi_2 - \beta_1\Pi_0$$

$$\beta_1 = \frac{\Pi_3}{\Pi_1}$$

Overidentification

In order to illustrate overidentification consider the following, since income and consumer wealth are very big influencers of demand, suppose that the demand function (11) is modified as follows and supply function is the same as before:

Demand function:

$$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 I_t + \alpha_3 R_t + \mu_{1t} \tag{22}$$

Supply function:

$$Q_t = \beta_0 + \beta_1 P_t + \beta_2 P_{t-1} + \mu_{2t} \tag{23}$$

The variable R represent wealth and the other variables are defined as before, now setting the demand function equal to the supply function, the following equilibrium price and quantity are obtained:

$$P_t = \Pi_0 + \Pi_1 I_t + \Pi_2 R_t + \Pi_3 P_{t-1} + v_t \tag{24}$$

$$Q_t = \Pi_4 + \Pi_5 I_t + \Pi_6 R_t + \Pi_7 P_{t-1} + w_t \tag{25}$$

where

$$\Pi_0 = \frac{\beta_0 - \alpha_0}{\alpha_1 - \beta_1} \qquad \Pi_1 = -\frac{\alpha_2}{\alpha_1 - \beta_1} \tag{26}$$

$$\Pi_2 = \frac{\alpha_3}{\alpha_1 - \beta_1} \qquad \Pi_3 = \frac{\beta_2}{\alpha_1 - \beta_1} \tag{27}$$

$$\Pi_4 = \frac{\alpha_1\beta_0 - \alpha_0\beta_1}{\alpha_1 - \beta_1} \qquad \Pi_5 = -\frac{\alpha_2\beta_1}{\alpha_1 - \beta_1} \tag{28}$$

$$\Pi_6 = \frac{\alpha_3\beta_1}{\alpha_1 - \beta_1} \qquad \Pi_7 = -\frac{\alpha_1\beta_2}{\alpha_1 - \beta_1} \tag{29}$$

$$w_t = \frac{\alpha_1\mu_{2t} - \beta_1\mu_{1t}}{\alpha_1 - \beta_1} \qquad v_t = \frac{\mu_{2t} - \mu_{1t}}{\alpha_1 - \beta_1} \tag{30}$$

Notice that the demand and supply model has seven structural coefficients and only eight equations are available to estimate them (equations 26-30). Note that unique estimation of all parameters for the model is impossible as there more equations than the unknowns to be estimated. This implies that there is an over-sufficiency of information to identify the supply curve, which is the opposite of underidentification where there is too little information as discussed above. Hence the supply curve is overidentified.

2.3 Identification Rules

In the previous section it was shown how one can determine whether a system of simultaneous equation is identified by examining its reduced form equation. This process is time consuming, and so this section will look at two methods that can be applied to see whether a system of simultaneous equation is identified or not, namely; the order condition of identifiability and the rank condition of identifiability. Both these

methods are a less time consuming way of determining whether equations in a simultaneous equation model are identified or not.

Consider the following notation which will be used in facilitating the discussion around the rule of identification, M denotes the number of end variables in the system of equations, m denotes the number of end variables in a specific equation, K denotes the number of exogenous variables in the system of equations and lastly k denotes the number of exogenous variables in a specific equation.

The Order Condition of Identifiability

The following necessary but not sufficient identification condition is known as the order condition and it is defined in two different ways:

- **Definition 1** In a simultaneous equation model with M simultaneous equations, for a particular equation to be identified, that equation needs to omit exactly M-1 variables (both endogenous and predetermined). If the equation omits exactly M-1 variables, then that equation is said to be just identified; however, if that particular equation omits more than M-1 then the equation is overidentified.
- **Definition 2** In a simultaneous equation model with M simultaneous equations, for a particular equation to be identified, the number of predetermined variables should always be more than the number of endogenous variables contained in that equation minus 1, i.e,

$$K - k \geq m - 1$$

An equation is identified if $K-k \geq m-1$ but if $K-k > m-1$ then that particular equation is overidentified. This method of determining whether an equation is identified will be illustrated with an example in section 3.

The Rank Condition of Identifiability

As discussed above the order condition is a necessary but not sufficient condition for identification, this implies that even if the condition holds it can happen that the equation is not identified. Thus a condition that will be sufficient and necessary is needed to check if an equation is identified or not. The rank condition of identifiability is necessary and sufficient, and it is stated as follows:

- In a simultaneous equation model with M simultaneous equations, where there are M endogenous

variables, an equation from such a model is identified if it possible to obtain one or more determinant of order (M-1)*(M-1) which is not zero, which is obtained from coefficient of variables excluded from a specific equation, but still included in other equations of the model. This method of will also be illustrated with an example in section 3.

2.4 Estimation Methods

The previous section looked at the nature of simultaneous equation models, their features and rules that can be used to identify equations. This section discusses the estimation of the parameters of such models by applying some of the rules outlined in the previous sections. This will be done by looking at indirect least squares (ILS) method, which is a method used to estimate the parameters of a just identified equation and method of two stage least squares (2SLS), this is a method used to estimate an overidentified equation.

Estimation of a Just Identified Equation: Indirect Least Square (ILS)

The estimation of just identified equation using ILS method involves the following three steps:

- **Step 1** Derive the reduced form equations from the structural equation where the dependent variable in each equation is expressed as a function of only independent (exogenous) variables and random (stochastic) error term.
- **Step 2** Now the OLS method can be used in the reduced form equations individually, as the explanatory variables in these equations are determined outside the model and therefore not corresponding with random variables or stochastic disturbances. Therefore the estimates attained here will be consistent.
- **Step 3** Finally get original structural coefficient estimates of a reduced form coefficients in previous step. It is pointed out in section 2.2, if a particular equation is fully identified then a one-to-one relationship among the structural and reduced form coefficients exists, hence unique estimates are derived.

The steps outlined above illustrate the fact that the name ILS is indeed derived from structural coefficients which are indirectly obtained from the OLS estimates of the reduced form equations. An example of this estimation method will illustrated in section 3.

Estimation of an Overidentified Equation: Two-Stage Least Squares (2SLS)

In this section 2SLS will be discussed as an estimation method for overidentified equations. Unlike ILS, 2SLS

can be used to obtain one estimates for each parameter in an overidentified equation; however, 2SLS can also be used to get estimates of a just identified equation and these estimates will be identical to the ones obtained using ILS.

To illustrate the 2SLS, consider the following model as specified in Gujarati [6]:

Income function:

$$Y_{1t} = \beta_{10} + \qquad + \beta_{11} Y_{2t} + \gamma_{11} X_{1t} + \gamma_{12} X_{2t} + \mu_t \tag{31}$$

Money supply function:

$$Y_{2t} = \beta_{20} + \beta_{21} Y_{1t} \tag{32}$$

where

Y_1 = income

Y_2 =stock of money

X_1 =investment expenditure

X_2 =government expenditure on goods and services

Note that X_1 and X_2 are predetermined variables. The income function (31) shows that the income is a function of Y_2, X_1 and X_2, while money supply function (32) is a function of (31), this clearly shows that a simultaneous-equation problem between these two functions exist. Such a problem can also be tested using the simultaneity test discussed below.

Simultaneity Test

A simultaneity test is used to test whether a simultaneous-equation problem exists, this is done by applying OLS since OLS estimators will be consistent and efficient if no simultaneity problem exists. However, if simultaneity exists then 2SLS method will also produce estimates that are efficient as well as consistent. This implies that a simultaneity test, determines whether the dependent variable which will be the regressor and error term have any correlation.

If the order condition of identification is used to determine whether the equations in this model are identified, it is observed that the income equation is underidentified whereas the money supply function is overidentified. This implies that the ILS method cannot be used to identify the money supply function as there exists two

estimates for β_{21}, therefore 2SLS should be used to estimate the money supply function.

The 2SLS process of estimating the parameters of the money supply function is explained in the following two stages:

Stage 1: Firstly one needs to remove any correlation that may exist between Y_1 and μ_2, this is done by first regressing Y_1 upon all exogenous variables that are in the entire model. i.e, regress Y_1 on both X_1 as well as X_2 :

$$Y_{1t} = \hat{\Pi}_0 + \hat{\Pi}_1 X_{1t} + \hat{\Pi}_2 X_{2t} + \hat{\mu}_t \tag{33}$$

where $\hat{\mu}_t$ is the usual OLS residuals, from equation (33) the following estimates are obtained:

$$\hat{Y}_{1t} = \hat{\Pi}_0 + \hat{\Pi}_1 X_{1t} + \hat{\Pi}_2 X_{2t} \tag{34}$$

Note that $\hat{Y}_{1t}$ is an estimate of the average of Y that is obtained from the X's. It is observed that (33) is written as a reduced form regression since Y_{1t} is written as a function of only the predetermined variables. Therefore equation (33) is expressed as

$$Y_{1t} = \hat{Y}_{1t} + \hat{\mu}_t \tag{35}$$

this shows that a random Y_1 is expressed in two segments, with $\hat{Y}_{1t}$ that is a expressed as a function of the X's and stochastic term $\hat{\mu}_t$. Now by OLS, it is seen that no correlation between $\hat{Y}_{1t}$ and $\hat{\mu}_t$ exists.

Stage 2: The overidentified money supply function is expressed as:

$$\begin{aligned} Y_{2t} &= \beta_{20} + \beta_{21}(\hat{Y}_{1t} + \hat{\mu}_t) + \mu_{2t} \\ &= \beta_{20} + \beta_{21}\hat{Y}_{1t} + (\mu_{2t} + \beta_{21}\hat{\mu}_t) \\ &= \beta_{20} + \beta_{21}\hat{Y}_{1t} + (\mu_t^*) \end{aligned} \tag{36}$$

where $\mu_t^* = \mu_{2t} + \beta_{21}\hat{\mu}_t$

Now comparing equation (36) with equation (32), it is observed that the equations are similar, the only difference is that Y_1 is now written as $\hat{Y}_{1t}$. Equation (33) is advantageous as it is uncorrelated with μ_t^* asymptotically, this implies that as the sample size increases the correlation between $\hat{Y}_{1t}$ and μ_t^* decreases. Therefore, OLS method is now appropriate to use in equation (32), this will result in having consistent and efficient estimates of parameters of the money supply function.

The 2SLS process deals with removing effects of the stochastic disturbance from the explanatory variable Y_1 in the money supply function. Hence, the first stage started with regressing Y_1 upon all exogenous variables that exist in the entire model, which then enables one to get estimates of $\hat{Y}_{1t}$ and the estimates are consistent and efficient; meaning that the estimates converge to their true values as the sample size increases.

Estimation of Standard Errors of 2SLS Estimators

The standard errors of the estimates in stage 2 of the 2SLS procedure are considered in this section. When using the OLS method it can be shown that they are not always good estimates for true standard error estimates. This implies the estimates need to be corrected using the method illustrated below.

As stated previously the standard error estimates obtained in stage 2 are not always good estimates for the actual standard errors. Now to illustrate this consider the money-supply model given in equations (32) and (33).

$$Y_{2t} = \beta_{20} + \beta_{21}\hat{Y}_{1t} + \mu_t^* \tag{37}$$

where

$$\mu_t^* = \mu_{2t} + \beta_{21}\hat{\mu}_t \tag{38}$$

Running regression on equation (37), the standard error of $\hat{\beta}_{21}$ can be derived as follows:

$$var(\hat{\beta_{21}}) = \frac{\hat{\sigma^2}_{\mu^*}}{\sum \hat{y_{1t}}} \tag{39}$$

where

$$\hat{\sigma^2}_{\mu^*} = \frac{\sum(\hat{\mu_t^*})^2}{n-2} = \frac{\sum(Y_{2t} - \hat{\beta}_{20} + \hat{\beta}_{21}\hat{Y}_{1t})^2}{n-2} \tag{40}$$

Note that $\hat{\sigma^2}_{\mu_2}$ and $\sigma^2_{\mu_*}$ are no the same, the first one is an unbiased estimate real value of the variance of μ_2. The true variance of $\hat{\sigma^2}_{\mu_2}$ can be obtained by first considering the following

$$\hat{\mu_{2t}} = Y_{2t} - \hat{\beta}_{20} + \hat{\beta}_{21}\hat{Y}_{1t}$$

Note that $\hat{\beta}_{20}$ and $\hat{\beta}_{21}$ are estimates that obtained from the first regression, there the true variance of $\hat{\sigma^2}_{\mu_2}$ is:

$$\hat{\sigma_\mu^2} = \frac{\sum(\hat{\mu_t^*})^2}{n-2} = \frac{\sum(Y_{2t} - \hat{\beta}_{20} + \hat{\beta}_{21}Y_{1t})^2}{n-2} \tag{41}$$

The difference between equations (40) and (41) is that in equation (41) is that the true value Y_1 is used instead of the estimation from the regression performed in stage 1.

A practical example which gives a step-by-step approach of obtaining estimates using 2SLS will be discussed in section 3.

3 Application

3.1 The Order Condition of Identifiability Examples

The order condition of identifiability is illustrated based on the following three examples.

Example 1

To show the order condition, suppose the following demand-and-supply model defined as it was defined in section 2.2 equations (1) and (2):

Demand function:

$$Q_t^d = \alpha_0 + \alpha_1 P_t + \mu_{1t} \qquad \alpha_1 < 0$$

Supply function:

$$Q_t^s = \beta_0 + \beta_1 P_t + \mu_{2t} \qquad \beta_1 < 0$$

Notice how this model has no predetermined variables and two endogenous variables P and Q . To be identified both these equations must exclude at least M-1 = 1 b variables, but because this condition does not hold here, none of these equations are identified.

Example 2

Secondly consider the model used in section 2.2 equations (11) and (12):

Demand function:

$$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 I_t + \mu_{1t} \qquad \alpha_1 < 0, \alpha_2 > 0$$

Supply function:

$$Q_t = \beta_0 + \beta_1 P_t + \mu_{2t} \qquad \beta_1 < 0$$

It can be observed from the above-mentioned model that I is a predetermined variable and P and Q are endogenous variables. Using the second definition of the order condition of identifiability, it is observed that the demand function is unidentified. However, the supply function is just identified as exactly M-1=1 variable I_t is excluded from the function.

Example 3

Lastly consider the following demand-and- supply model:

Demand function:

$$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 I_t + \alpha_3 R_t + \mu_{1t} \tag{42}$$

Supply function:

$$Q_t = \beta_0 + \beta_1 P_t + \beta_2 P_{t-1} + \mu_{2t} \tag{43}$$

Again in the two models P_t and Q_t are endogenous variables and I_t,R_t and P_{t-1} are exogenous variables. It is seen that the demand function excludes exactly one variable P_{t-1} and so it is exactly identified. Now I_t and R_t are both excluded and so the supply function is said to be overidentified. This also illustrate the fact that the order condition of identifiability is necessary but not always sufficient.

The previous examples showed that an equation is identifiable if that particular equation excludes one or more equations that are appear in other equations of the same model. This is known as the exclusion criterion [11].

3.2 The Rank Condition of Identifiability Example

In this section the rank condition of identifiability is discussed based on the following hypothetical model as stated in [6].

$$
\begin{array}{llll}
Y_{1t} - \beta_{10} & - \beta_{12} Y_{2t} - \beta_{13} Y_{3t} + \gamma_{11} X_{1t} = \mu_{1t} \\
Y_{2t} - \beta_{20} & - \beta_{23} Y_{3t} - \gamma_{21} X_{1t} - \gamma_{22} X_{2t} = \mu_{2t} \\
Y_{3t} - \beta_{30} - \beta_{31} Y_{1t} & - \gamma_{31} X_{1t} - \gamma_{32} X_{2t} = \mu_{3t} \\
Y_{4t} - \beta_{40} - \beta_{41} Y_{1t} - \beta_{42} Y_{2t} & - \gamma_{43} X_{3t} = \mu_{4t}
\end{array}
$$

Figure 1: Hypothetical Model

Now to determine whether the equations are identified, the hypothetical model (figure 1) can rewritten in table form as shown in table 1.

Equations	1	Y_1	Y_2	Y_3	Y_4	X_1	X_2	X_3
1st	$-\beta_{10}$	1	$-\beta_{12}$	$-\beta_{13}$	0	$-\gamma_{11}$	0	0
2nd	$-\beta_{20}$	0	1	$-\beta_{23}$	0	$-\gamma_{21}$	$-\gamma_{22}$	0
3rd	$-\beta_{30}$	$-\beta_{31}$	0	1	0	$-\gamma_{31}$	$-\gamma_{32}$	0
4th	$-\beta_{40}$	$-\beta_{41}$	$-\beta_{42}$	0	1	0	0	$-\gamma_{43}$

Table 1: Coefficients of Variables

For the following section note that (K-k) is the number of exogenous variables excluded and (m-1) is the number of endogenous variables in the model minus one.

Equations	(K-k)	(m-1)	Identified?
1st	2	2	Exactly
2nd	1	1	Exactly
3rd	1	1	Exactly
4th	2	2	Exactly

Table 2: Order Of Identification

Previous discussions show that the rank condition indicates whether a particular equation considered is identified or not, while order condition indicates whether that particular equation is exactly identified or overidentified.

3.3 Estimation Methods Illustrative Examples

3.3.1 Indirect Least Squares

To illustrate the ILS method,consider the following demand and supply model as stated in Gujarati [6].

Demand function:

$$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 X_t + \mu_{1t} \qquad \alpha_1 < 0 \tag{44}$$

Supply function:

$$Q_t = \beta_0 + \beta_1 P_t + \mu_{2t} \qquad \beta_1 < 0 \tag{45}$$

where

Q = quantity

P = price

X = income or expenditure

Note that X is exogenous, as seen before the supply function is exactly identified but the demand function is not identified.

Since the demand function is identified, then its parameters are estimated using ILS. The reduced form equations of the preceding structural equations are given by:

$$P_t = \Pi_0 + \Pi_1 X_t + w_t \tag{46}$$

$$Q_t = \Pi_2 + \Pi_3 X_t + v_t \tag{47}$$

Note that the reduced form coefficients are given by the Π's and are obtained by getting linear combinations of the structural coefficients, also note that ω and ν are linear combinations of μ_1 and μ_2.

Note that P_t and Q_t are dependent variables and are written in terms of only one the predetermined variable X (income) and random disturbances. Therefore, parameters of P_t and Q_t may be estimated using OLS. The estimates are given as

$$\hat{\Pi}_1 = \frac{\sum p_t x_t}{\sum x_t^2} \tag{48}$$

$$\hat{\Pi}_0 = \bar{P} - \hat{\Pi}_1 \bar{X} \tag{49}$$

$$\hat{\Pi}_3 = \frac{\sum q_t x_t}{\sum x_t^2} \tag{50}$$

$$\hat{\Pi}_2 = \bar{Q} - \hat{\Pi}_3 \bar{X} \tag{51}$$

where the lowercase letters show the observed values from sample means and also note that $\bar{Q}$ and $\bar{P}$ are the sample observations of Q and P. The $\hat{\Pi}_i$'s are consistent and asymptotically efficient, this implies that the estimates converge to their true values as the sample size increases.

Since the primary objective is to determine the structural coefficients, as shown previously the supply function is exactly identified with the following reduced form coefficients

$$\beta_0 = \Pi_2 - \beta_1 \Pi_0$$

and

$$\beta_1 = \frac{\Pi_3}{\Pi_1}$$

Therefore, the estimates are

$$\hat{\beta}_0 = \hat{\Pi}_2 - \hat{\beta}_1 \hat{\Pi}_0 \tag{52}$$

$$\hat{\beta}_1 = \frac{\hat{\Pi}_3}{\hat{\Pi}_1} \tag{53}$$

These are ILS estimators remember that the demand function is not estimable as it not identified.

Now to give numeric results, consider data in table 3 given in the appendix as well as the obtained SAS regression results, the the results were obtained by regressing price (P) on per capita real consumption expenditure (X) and regressing quantity (Q) on per capita real consumption expenditure (X). The following results were obtained:

$$\begin{aligned} \hat{P}_t &= 90.9601 + 0.0007 X_1 \qquad (54) \\ se &= (4.0517) \quad (0.0002) \\ t &= (22.4499) \quad (3.0060) \\ R^2 &= 0.2440 \end{aligned}$$

$$\begin{aligned} \hat{Q}_t &= 59.7618 + 0.0020X_1 \quad (55) \\ se &= (1.5600) \quad (0.00009) \\ t &= (38.3080) \quad (20.9273) \\ R^2 &= 0.9399 \end{aligned}$$

Now using equations (52) and (53), the following ILS estimates are obtained:

$$\hat{\beta}_0 = -183.7043 \quad (56)$$

$$\hat{\beta}_1 = 2.6766 \quad (57)$$

The estimated ILS regression is given by:

$$Q_t = -183.7043 + 2.6766P_t \quad (58)$$

3.3.2 Two-Staged Least Squares

This section gives the practical illustration of the discussion of the 2SLS method as discussed in section 2. The same money-supply model is considered. OLS method can be used to estimate the parameters of the money supply function, however, the obtained estimates will be inconsistent as a high correlation between the random variable Y_1 and μ_2. Assume that a proxy for Y_1 can be obtained where it is uncorrelated with μ_2. So now, a way of obtaining such an instrumental variable is needed. The instrumental variable can be obtained by using the 2SLS, which was developed independently.

Stage 1 Regression Regress the stochastic explanatory variable income Y_1, on the exogenous variables X_1 and X_2, then the results are obtained as follows:

$$\begin{aligned} \hat{Y}_{1t} &= 2689.848 + 1.8700X_{1t} + 2.0343X_{2t} \\ se &= (67.9874) \quad (0.1717) \quad (0.1075) \\ t &= (39.5639) \quad (10.8938) \quad (18.9295) \\ R^2 &= 0.9964 \end{aligned} \tag{59}$$

From the regression coefficient it is observed that 99.64% of the variation of Y_1 is explained by the regression model.

Stage 2 Regression This stage discusses estimating parameters of the money-supply function equation (32), replacing Y_1 with the estimated Y_1 obtained in stage 1. The obtained variables are as follows:

$$\begin{aligned} \hat{Y}_{2t} &= -2240.18 + 0.702\hat{Y}_{1t} \\ se &= (127.372) \quad (0.0178) \\ t &= (-19.1579) \quad (44.5246) \\ R^2 &= 0.9831 \end{aligned} \tag{60}$$

The estimated standard errors given in equation (60) should be corrected using the method that is discussed in section 2.4. Effecting this correction gives the following estimates:

$$\begin{aligned} \hat{Y}_{2t} &= -2240.18 + 0.702\hat{Y}_{1t} \\ se &= (126.9598) \quad (0.0212) \\ t &= (-17.3354) \quad (37.3812) \\ R^2 &= 0.9803 \end{aligned} \tag{61}$$

It is observed that the estimated standard errors given in equation (61) do not differ much from the estimates given in equation (55), this is due to the fact that correlation coefficient in stage 1 regression is very high. This implies the estimates obtained from classic OLS and stage 2 of 2SLS will be the similar;

however, this may not always be the case in practice. Hence, the second stage of 2SLS needs to be performed always.

4 Conclusion

This research project looked at simultaneous equation models, discussing their features, their estimation and some of the statistical problems associated with them. This was done by considering the identification problem: underidentified, just or exact identification and overidentification by looking at the difference between estimation of a just identified equation with indirect least square (ILS) and estimation of an overidentified equation with method of two stage least squares (2SLS).

Shortfall of this research is that only two methods of identification were considered; therefore, as an improvement one can also consider other methods of identification as introduced in the section 1 and can also have the system of equation represented in matrix form. System of equations in matrix form will make it easy to use SAS IML to estimate structural equations.

References

[1] Carl F Christ. The Cowles Commission's contributions to econometrics at Chicago, 1939-1955. *Journal of Economic Literature*, 32(1):30–59, 1994.

[2] Marcel G Dagenais. Parameter estimation in regression models with errors in the variables and autocorrelated disturbances. *Journal of Econometrics*, 64(1):145–163, 1994.

[3] Davidson, Russels, Mackinnon, and James G. *Estimation and Inference in Econometrics*. JSTOR, 1993.

[4] Damodar Gujarati. *Econometrics by Example*. Palgrave Macmillan, 2014.

[5] Damodar N Gujarati and Dawn C Porter. *Essentials of Econometrics*. McGraw-Hill Singapore, 1999.

[6] Damoder N Gujarati. *Basic Econometrics*. Tata McGraw-Hill Education, 2009.

[7] Jerry A Hausman. *Specification and Estimation of Simultaneous Equation Models*. Elsevier, 1983.

[8] James J Heckman. *Dummy Endogenous Variables in a Simultaneous Equation System*. National Bureau of Economic Research Cambridge, Mass., USA, 1977.

[9] Michael D Intriligator. *Econometric Models, Techniques, and Applications*. Prentice-Hall Englewood Cliffs, NJ, 1978.

[10] Harry H Kelejian. Two-stage least squares and econometric systems linear in parameters but nonlinear in the endogenous variables. *Journal of the American Statistical Association*, 66(334):373–374, 1971.

[11] Jan Kimenta. *Elements of Econometrics*. Peterson Evergreen, 1986.

[12] Lawrence Robert Klein. *Economic Fluctuations in the United States, 1921-1941*. Wiley, 1950.

[13] LR Klein. *A Textbook of Econometrics*. Evanston Peterson, 1953.

[14] Tjalling C Koopmans. *Identification Problem in Economic Model Construction*. JSTOR, 1949.

[15] Tjalling Charles Koopmans. *Statistical Inference in Dynamic Economic Models*. Number 10. Wiley, 1950.

[16] B. Shreya, J. Sargam, S. Nevrekar, and D. Kumar. Identification of simultaneous equation models. *Article on the identification of simultaneous equation models*, 2001.

[17] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton university press, 2007.

[18] Arnold Zellner. An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American Statistical Association*, 57(298):348–368, 1962.

Appendix

Just or Exact Identification Calculation:

Demand function:

$$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 I_t + \mu_{1t} \qquad \alpha_1 < 0, \alpha_2 > 0$$

Supply function:

$$Q_t = \beta_0 + \beta_1 P_t + \mu_{2t} \qquad \beta_1 < 0$$

Now Solving for P_t from the following equilibrium condition:

$$\begin{aligned} \alpha_0 + \alpha_1 P_t + \alpha_2 I_t + \mu_{1t} &= \beta_0 + \beta_1 P_t + \mu_{2t} \\ (\alpha_1 - \beta_1) P_t &= (\beta_0 - \alpha_0) - \alpha_2 I_t + (\mu_{2t} - \mu_{1t}) \\ P_t &= \frac{\beta_0 - \alpha_0}{\alpha_1 - \beta_1} + \frac{-\alpha_2}{\alpha_1 - \beta_1} I_t + \frac{\mu_{2t} - \mu_{1t}}{\alpha_1 - \beta_1} \end{aligned}$$

Now substituting the value of P_t into the demand or supply function to get Q_t:

$$\begin{aligned} Q_t &= \alpha_0 + \alpha_1 \left(\frac{\beta_0 - \alpha_0}{\alpha_1 - \beta_1} + \frac{-\alpha_2}{\alpha_1 - \beta_1} I_t + \frac{\mu_{2t} - \mu_{1t}}{\alpha_1 - \beta_1}\right) + \alpha_2 I_t + \mu_{1t} \\ &= \frac{\alpha_0\alpha_1 - \alpha_0\beta_1}{\alpha_1 - \beta_1} + \frac{\alpha_1\beta_0 - \alpha_0\alpha_1}{\alpha_1 - \beta_1} + \frac{-\alpha_1\alpha_2}{\alpha_1 - \beta_1} I_t + \frac{\alpha_1\mu_{2t} - \alpha_1\mu_{1t}}{\alpha_1 - \beta_1} + \frac{\alpha_1\alpha_2 I_t - \alpha_2\beta_1 I_t}{\alpha_1 - \beta_1} + \frac{\alpha_1\mu_{1t} - \beta_1\mu_{1t}}{\alpha_1 - \beta_1} \\ &= \frac{\alpha_1\beta_0 - \alpha_0\beta_1}{\alpha_1 - \beta_1} + \frac{-\alpha_2\beta_1 I_t}{\alpha_1 - \beta_1} + \frac{\alpha_1\mu_{2t} - \beta_1\mu_{1t}}{\alpha_1 - \beta_1} \end{aligned}$$

Overidentification Calculations:

Demand function:

$$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 I_t + \alpha_3 R_t + \mu_{1t}$$

Supply function:

$$Q_t = \beta_0 + \beta_1 P_t + \beta_2 P_{t-1} + \mu_{2t}$$

Solving for P_t from the equilibrium condition:

$$\begin{aligned}
\alpha_0 + \alpha_1 P_t + \alpha_2 I_t + \alpha_3 R_t + \mu_{1t} &= \beta_0 + \beta_1 P_t + \beta_2 P_{t-1} + \mu_{2t} \\
(\alpha_1 - \beta_1) P_t &= (\beta_0 - \alpha_0) - \alpha_2 I_t - \alpha_3 R_t + \beta_2 P_{t-1} + (\mu_{2t} - \mu_{1t}) \\
P_t &= \frac{\beta_0 - \alpha_0}{\alpha_1 - \beta_1} + \frac{-\alpha_2 I_t}{\alpha_1 - \beta_1} + \frac{-\alpha_3 R_t}{\alpha_1 - \beta_1} + \frac{\beta_2}{\alpha_1 - \beta_1} + \frac{\mu_{2t} - \mu_{1t}}{\alpha_1 - \beta_1}
\end{aligned}$$

Now substituting the value of P_t into the demand or supply function to get Q_t:

$$\begin{aligned}
Q_t &= \alpha_0 + \alpha_1 \left(\frac{\beta_0 - \alpha_0}{\alpha_1 - \beta_1} + \frac{-\alpha_2 I_t}{\alpha_1 - \beta_1} + \frac{-\alpha_3 R_t}{\alpha_1 - \beta_1} + \frac{\beta_2}{\alpha_1 - \beta_1} + \frac{\mu_{2t} - \mu_{1t}}{\alpha_1 - \beta_1}\right) + \alpha_2 I_t + \alpha_3 R_t + \mu_{1t} \\
&= \frac{\alpha_0}{\alpha_1 - \beta_1} + \frac{\alpha_1\beta_0 - \alpha_0\alpha_1}{\alpha_1 - \beta_1} + \frac{-\alpha_1\alpha_2 I_t}{\alpha_1 - \beta_1} + \frac{-\alpha_1\alpha_3 R_t}{\alpha_1 - \beta_1} + \frac{\alpha_1\beta_2}{\alpha_1 - \beta_1} + \frac{\alpha_1\mu_{2t} - \alpha_1\mu_{1t}}{\alpha_1 - \beta_1} + \frac{\alpha_2 I_t}{\alpha_1 - \beta_1} + \frac{\alpha_3 R_t}{\alpha_1 - \beta_1} + \frac{\mu_{1t}}{\alpha_1 - \beta_1} \\
&= \frac{\alpha_0}{\alpha_1 - \beta_1} + \frac{\alpha_1\beta_0 - \alpha_0\alpha_1}{\alpha_1 - \beta_1} + \frac{-\alpha_1\alpha_2 I_t}{\alpha_1 - \beta_1} + \frac{-\alpha_1\alpha_3 R_t}{\alpha_1 - \beta_1} + \frac{\alpha_1\beta_2}{\alpha_1 - \beta_1} + \frac{\alpha_1\mu_{2t} - \alpha_1\mu_{1t}}{\alpha_1 - \beta_1} + \frac{\alpha_2 I_t(\alpha_1 - \beta_1)}{\alpha_1 - \beta_1} \\
&\quad + \frac{\alpha_3 R_t(\alpha_1 - \beta_1)}{\alpha_1 - \beta_1} + \frac{\mu_{1t}(\alpha_1 - \beta_1)}{\alpha_1 - \beta_1}
\end{aligned}$$

Therefore,

$$Q_t = \Pi_4 + \Pi_5 I_t + \Pi_6 R_t + \Pi_7 P_{t-1} + w_t$$

where

$$\Pi_4 = \frac{\alpha_1\beta_0 - \alpha_0\beta_1}{\alpha_1 - \beta_1} \qquad \Pi_5 = -\frac{\alpha_2\beta_1}{\alpha_1 - \beta_1}$$

$$\Pi_6 = \frac{\alpha_3\beta_1}{\alpha_1 - \beta_1} \qquad \Pi_7 = -\frac{\alpha_1\beta_2}{\alpha_1 - \beta_1}$$

$$w_t = \frac{\alpha_1\mu_{2t} - \beta_1\mu_{1t}}{\alpha_1 - \beta_1}$$

Data Sets

The following table gives data on crop production, crop prices and per capita personal consumption expenditure, 2007, Dollars, United States, 1975-2004. This is the data that is used in section 3 to facilitate the discussion around estimation methods discussed in section 2.

Observation	Index of Crop Production (1996 = 100), Q	Index of Crop Prices Received by Farmers (1990 - 1992 = 100), P	Real per Capita Personal Consumption Expenditure, X
1975	66	88	4789
1976	67	87	5282
1977	71	83	5804
1978	73	89	6417
1979	78	98	7073
1980	75	107	7716
1981	81	111	8439
1982	82	98	8945
1983	71	108	9775
1984	81	111	10589
1985	85	98	11406
1986	82	87	12048
1987	84	86	12766
1988	80	104	13685
1989	86	109	14546
1990	90	103	15349
1991	90	101	15772
1992	96	101	16485
1993	91	102	17204
1994	101	105	18004
1995	96	112	18665
1996	100	127	19490
1997	104	115	20323
1998	105	107	21291
1999	108	97	22491
2000	108	96	23862
2001	108	99	24722
2002	107	105	25501
2003	108	111	26463
2004	112	117	27937

Table 3: Crops Data Set

Observation	GDP (Y_1)	M2 (Y_2)	GPDI (X_1)	FEDEXP (X_2)	TB6 (X_3)
1970	3 771.9	626.5	427.1	201.1	6.562
1971	3 898.6	710.3	475.7	220.0	4.511
1972	4 105.0	802.3	532.1	244.4	4.466
1973	4 341.5	855.5	594.4	261.7	7.178
1974	4 319.6	902.1	550.6	293.3	7.926
1975	4 311.2	1 016.2	453.1	346.2	6.122
1976	4 540.9	1 152.0	544.7	374.3	5.266
1977	4 750.5	1 270.3	627.0	407.5	5.510
1978	5 015.0	1 366.0	702.6	450.0	7.572
1979	5 173.4	1 473.7	725.0	497.5	10.017
1980	5 161.7	1 599.8	645.3	585.7	11.374
1981	5 291.7	1 755.4	704.9	672.7	13.776
1982	5 189.3	1 910.3	606.0	748.5	11.084
1983	5 423.8	2 126.5	662.5	815.4	8.75
1984	5 813.6	2 310.0	857.7	877.1	9.80
1985	6 053.7	2 495.7	849.7	948.2	7.66
1986	6 263.6	2 732.4	843.9	1 006.0	6.03
1987	6 475.1	2 831.4	870.0	1 041.6	6.05
1988	6 742.7	2 994.5	890.5	1 092.7	6.92
1989	6 981.4	3 158.5	926.2	1 167.5	8.04
1990	7 112.5	3 278.6	895.1	1 253.5	7.47
1991	7 100.5	3 379.1	822.2	1 315.0	5.49
1992	7 336.6	3 432.5	889.0	1 444.6	3.57
1993	7 532.7	3 484.0	968.3	1 496.0	3.14
1994	7 835.5	3 497.5	1 099.6	1 533.1	4.66
1995	8 031.7	3 640.4	1 134.0	1 603.5	5.59
1996	8 328.9	3 815.1	1 234.2	1 665.8	5.09
1997	8 703.5	4 031.6	1 387.7	1 708.9	5.18
1998	9 066.9	4 379.0	1 524.1	1 734.9	4.85
1999	9 470.3	4 641.1	1 642.6	1 787.6	4.76
2000	9 817.0	4 920.9	1 735.5	1 864.4	5.92
2001	9 890.7	5 430.3	1 598.4	1 969.5	3.39
2002	10 048.8	5 774.1	1 557.1	2 101.1	1.69
2003	10 301.0	6 062.0	1 613.1	2 252.1	1.06
2004	10 703.5	6 411.7	1 770.6	2 383.0	1.58
2005	11.048.6	6 669.4	1 866.3	2 555.9	3.40

Y_1 =GDP, gross domestic product (billions of chained 2000 dollars)
Y_2 =M2, money supply (billions of dollars)
X_1 =GPDI, gross private domestic investment (billions of chained 2000 dollars)
X_2 =FEDEXP, Federal government expenditure (billions of dollars)
X_3 =TB6, 6-month Treasury bill rate (%)

Table 4: Economic Report

Indirect Least Squares

SAS Code Used:

```
data crop;

input obs Q P X;

cards;

1975 66 88 4789

1976 67 87 5282

1977 71 83 5804

1978 73 89 6417

1979 78 98 7073
```

```
1980 75 107 7716
1981 81 111 8439
1982 82 98 8945
1983 71 108 9775
1984 81 111 10589
1985 85 98 11406
1986 82 87 12048
1987 84 86 12766
1988 80 104 13685
1989 86 109 14546
1990 90 103 15349
1991 90 101 15772
1992 96 101 16485
1993 91 102 17204
1994 101 105 18004
1995 96 112 18665
1996 100 127 19490
1997 104 115 20323
1998 105 107 21291
1999 108 97 22491
2000 108 96 23862
2001 108 99 24722
2002 107 105 25501
2003 108 111 26463
2004 112 117 27937
;
run;

/*regression of price on per capita real consumption expenditure*/
proc reg data = crop;
      model P =  X ;
run;
```

```
/*regression of quantity on per capita real consumption expenditure*/
proc reg data = crop;
      model Q =  X ;
  run;
```

Relevent SAS Output:

The REG Procedure
Model: MODEL1
Dependent Variable: P

Number of Observations Read	30
Number of Observations Used	30

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	749.36668	749.36668	9.03	0.0055
Error	28	2322.49998	82.94643		
Corrected Total	29	3071.86667			

Root MSE	9.10749	R-Square	0.2439
Dependent Mean	102.06667	Adj R-Sq	0.2169
Coeff Var	8.92308		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	90.95995	4.05208	22.45	<.0001
X	1	0.00073581	0.00024480	3.01	0.0055

The SAS System

The REG Procedure
Model: MODEL1
Dependent Variable: Q

Number of Observations Read	30
Number of Observations Used	30

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	5385.00915	5385.00915	437.73	<.0001
Error	28	344.45752	12.30205		
Corrected Total	29	5729.46667			

Root MSE	3.50743	R-Square	0.9399
Dependent Mean	89.53333	Adj R-Sq	0.9377
Coeff Var	3.91746		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	59.75969	1.56052	38.29	<.0001
X	1	0.00197	0.00009428	20.92	<.0001

Two-Staged Least Squares

SAS Code Used:

```
options nocenter ;
data table_4;
input Y1 Y2 X1 X2 X3;
cards;
```

3771.9 626.5 427.1 201.1 6.562

3898.6 710.3 475.7 220.0 4.511

4105.0 802.3 532.1 244.4 4.466

4341.5 855.5 594.4 261.7 7.178

4319.6 902.1 550.6 293.3 7.926

4311.2 1016.2 453.1 346.2 6.122

4540.9 1152.0 544.7 374.3 5.266

4750.5 1270.3 627.0 407.5 5.510

5015.0 1366.0 702.6 450.0 7.572

5173.4 1473.7 725.0 497.5 10.017

5161.7 1599.8 645.3 585.7 11.374

5291.7 1755.4 704.9 672.7 13.776

5189.3 1910.3 606.0 748.5 11.084

5423.8 2126.5 662.5 815.4 8.75

5813.6 2310.0 857.7 877.1 9.80

6053.7 2495.7 849.7 948.2 7.66

6263.6 2732.4 843.9 1006.0 6.03

6475.1 2831.4 870.0 1041.6 6.05

6742.7 2994.5 890.5 1092.7 6.92

6981.4 3158.5 926.2 1167.5 8.04

7112.5 3278.6 895.1 1253.5 7.47

7100.5 3379.1 822.2 1315.0 5.49

7336.6 3432.5 889.0 1444.6 3.57

7532.7 3484.0 968.3 1496.0 3.14

7835.5 3497.5 1099.6 1533.1 4.66

8031.7 3640.4 1134.0 1603.5 5.59

8328.9 3815.1 1234.2 1665.8 5.09

8703.5 4031.6 1387.7 1708.9 5.18

9066.9 4379.0 1524.1 1734.9 4.85

9470.3 4641.1 1642.6 1787.6 4.76

9817.0 4920.9 1735.5 1864.4 5.92

9890.7 5430.3 1598.4 1969.5 3.39

```
10048.8 5774.1 1557.1 2101.1 1.69
10301.0 6062.0 1613.1 2252.1 1.06
10703.5 6411.7 1770.6 2383.0 1.58
11048.6 6669.4 1866.3 2555.9 3.40
;
run;

proc Iml;
use table_4;
read all var _ALL_ into Matrix[colname=varNames];
close d1;
show names;
*print Matrix;



n = nrow(Matrix);

y1 = Matrix[,1];
ones = j(n,1,1);
X = ones||matrix[,3]||Matrix[,4];
*print x;

BetaHat=inv(X‘*X)*X‘*y1;
res=y1-X*BetaHat ;
yhat1=x*betahat;

*print yhat1;
*print BetaHat;

y2=matrix[,2];
x2=ones||yhat1;
*print x2;
```

```
betahat2=inv(x2‘*x2)*x2‘*y2;
print BetaHat betahat2;
```

Relevent SAS Output:

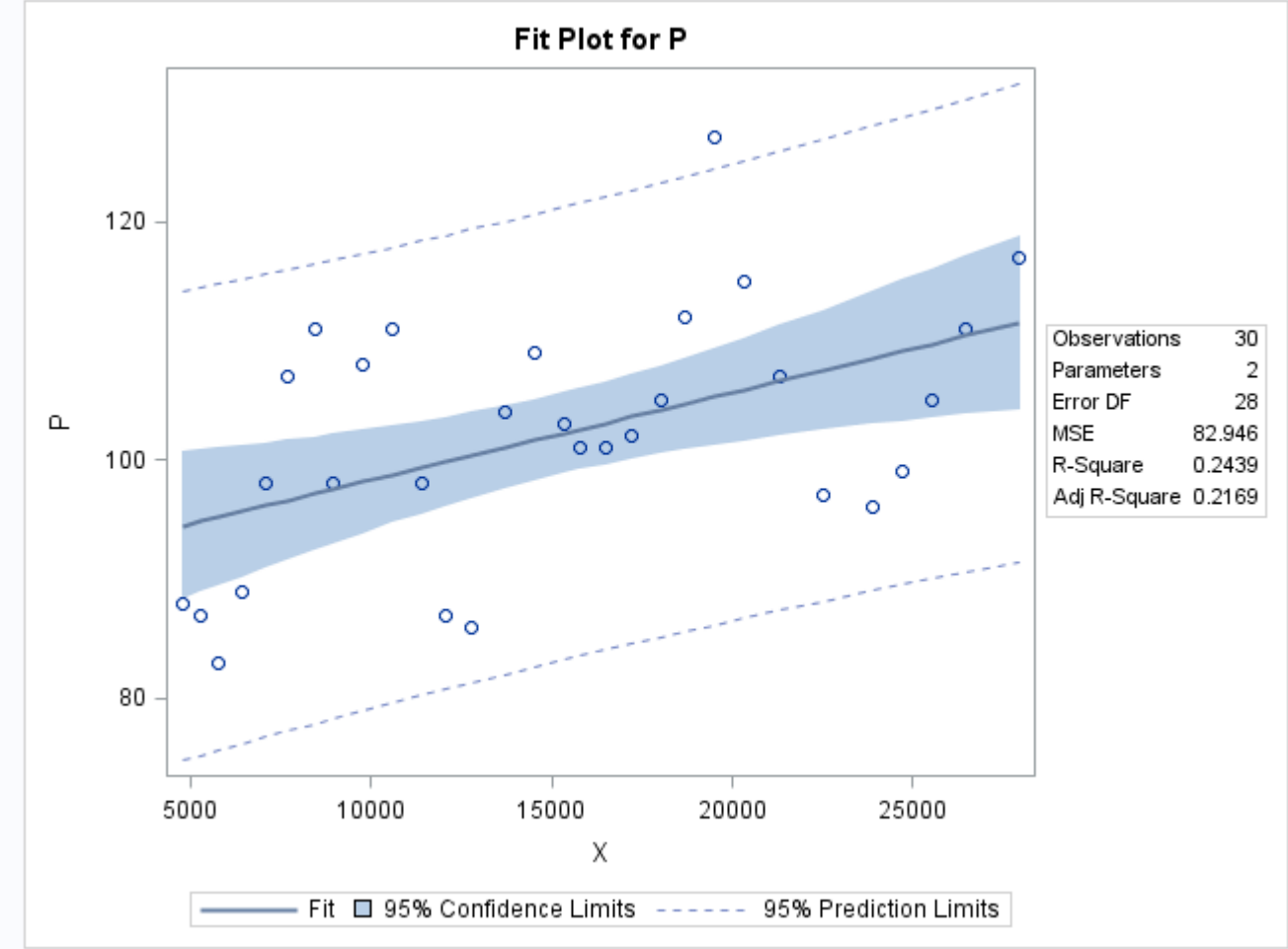
The SAS System

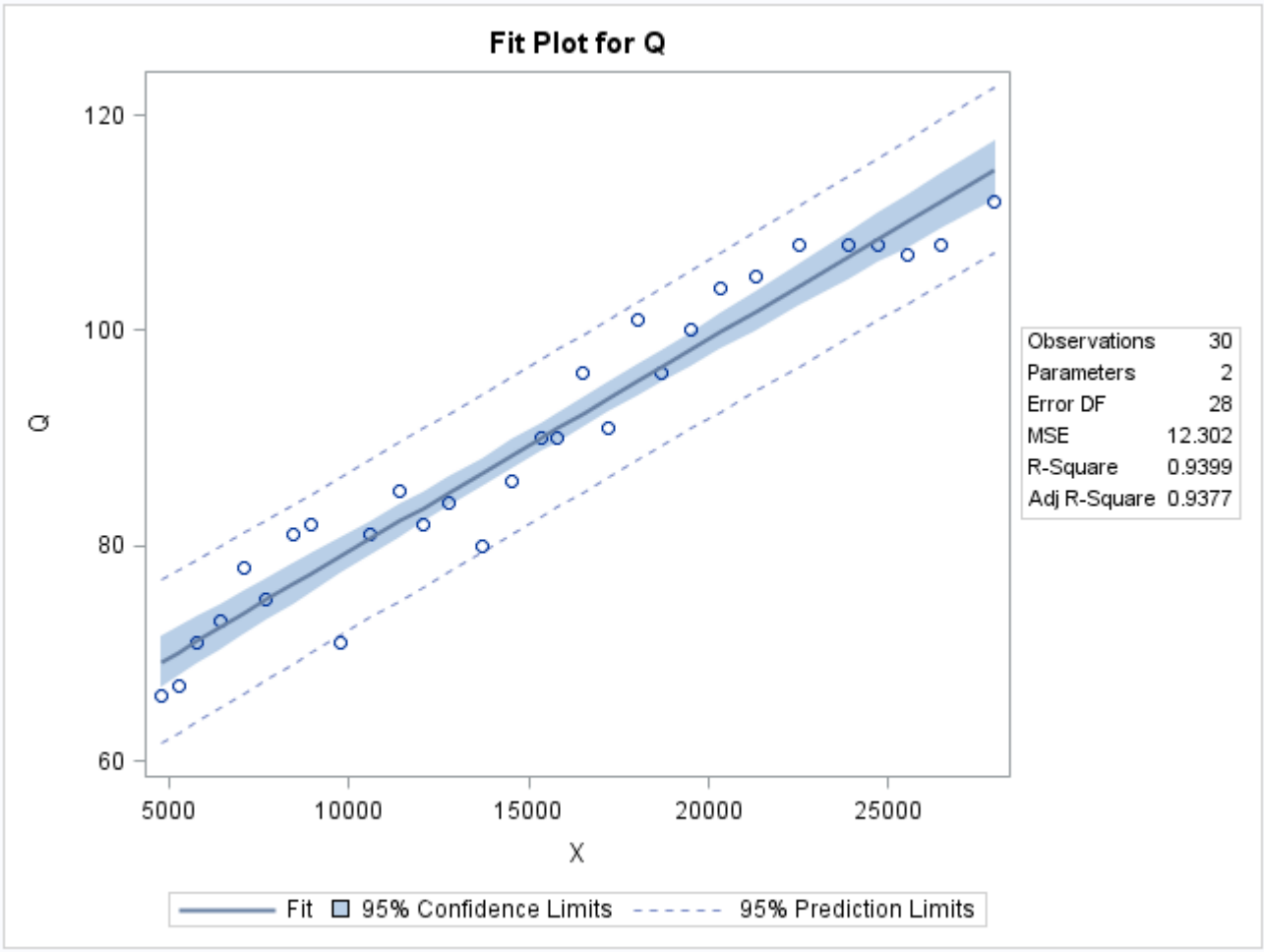
SYMBOL	ROWS	COLS	TYPE	SIZE	
Matrix	36	5	num	8	colname=varNames
varNames	1	5	char	2	

Number of symbols = 7 (includes those without values)

BetaHat	betahat2
2689.849	-2440.2
1.8699572	0.7919561
2.0343381	

Relevent SAS Output: Fit Plots for Price and Quantity


Fit Plot for P
P
120
100
80
5000
10000
15000
20000
25000
X
Observations 30
Parameters 2
Error DF 28
MSE 82.946
R-Square 0.2439
Adj R-Square 0.2169
Fit
95% Confidence Limits
95% Prediction Limits


Fit Plot for Q
120
100
80
60
Q
5000
10000
15000
20000
25000
X
Observations 30
Parameters 2
Error DF 28
MSE 12.302
R-Square 0.9399
Adj R-Square 0.9377
Fit
95% Confidence Limits
95% Prediction Limits

Spatial density estimation

Kabelo Mahloromela 14194237

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor(s): Dr I Fabris-Rotelli, Miss C Kraamwinkel

Department of Statistics, University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

30 October 2017

Abstract

Kernel density estimation is a data smoothing technique that can be applied in a spatial context. The analysis of spatial point patterns can be promoted by the implementation of the `R` software. This research will give particular focus to the `spatstat` package within `R` and the function that computes a kernel smoothed intensity function from spatial point data, namely `density.ppp`.The function and each of its parameters will be investigated and their effect on the kernel density estimate inspected. To this end we will analyze the effects of adjusting kernel options, allowing for edge effects and bandwidth selection for spatial point patterns for the standard rectangular window and the convex hull window.

Declaration

I, *Kabelo Mahloromela,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____

Kabelo Mahloromela

_____

Inger Fabris-Rotelli, Christine Kraamwinkel

_____

Date

Acknowledgements

I would like to show special gratitude to my supervisors for their continued support and guidance. With their assistance, I have garnered a repository of skills and experience. I would also like to acknowledge the Center for AI research, Meraka Institute, CSIR for their financial support.

Contents

List of Figures

List of Tables

1 Introduction

Spatial point patterns are characterized by the arrangement of objects or entities that are distributed in space or a region. A point pattern may be defined by a set B whose elements are a set of locations for observed values of data in a region R. The region over which data points are collected is termed a window. The window in which objects are studied may be various geometric shapes, such as a square, a complex polygon and could include non-convex shapes.

Figure 1 is an illustration that shows an example of a point pattern in which the points of two types are plotted on a plane[1]. Different symbols are used to distinguish markers that relay additional information. A subset of all the points can be viewed in an irregular window, also illustrated in Figure 1.

The aim of a statistical analysis may be to estimate parameters of a population distribution. When data is collected, meticulous attention needs to be paid to the selection of the sample window, the region of observation, as they indicate where data has not been collected. The choice of window may have implicit effects on the results of parameter estimates and may give erroneous output if not selected correctly. Figure 2 is an example of the spatial distribution of data[2]. The colors indicate wind speed, arrows indicate wind direction, April, July, October and January are the months of spring, summer, autumn and winter.

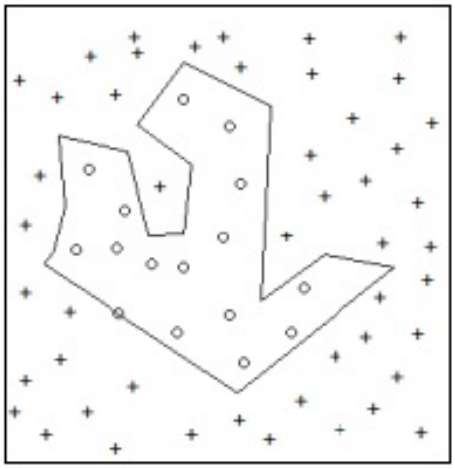


Figure 1: A point pattern plot

The patterns formed by the points are analyzed in many scientific disciplines, as a result, a vast range of applications may be considered. This includes but is not limited to biological cells, animals, plants or star constellations, amongst others [15]. Information derived from point pattern data may aid in the statistical inquiry and evaluation of the distribution of a collection of points on a plane or two dimensional surface.

[1] https://i0.wp.com/bio7.org/Spatstat_Introduction_files/figure-markdown_strict+autolink_bare_uris/unnamed-chunk-21-1.png?w=456. Date accessed 24 February 2017

[2] http://article.sciencepublishinggroup.com/html/10.11648.j.ijema.20160403.15.html. Date Accessed: 03 March 2017

Many statistical-based packages are available for such analysis. Baddeley and Turner are some of several statisticians that are involved with statistical computing, working mainly with spatial point patterns. They have authored and co-authored in literature such as [1, 3] . These are the main references that will be consulted for the research to gain mastery of the research topic.

`spatstat`[3, 2] is a package in `R` that equips researchers with practical techniques for the statistical analysis of spatial point patterns. Initially, the `spatstat` package could only be used for two dimensional point patterns. It can now support multi-dimensional patterns such as temporal data, that is spatial point patterns over time. In `spatstat`, a point pattern is denoted by an object class "ppp" and "owin" is an object class assigned to the window, the area of study. In this research we are going to examine the mathematics behind the `density.ppp` function, the available window options in `spatstat`, and edge effects.

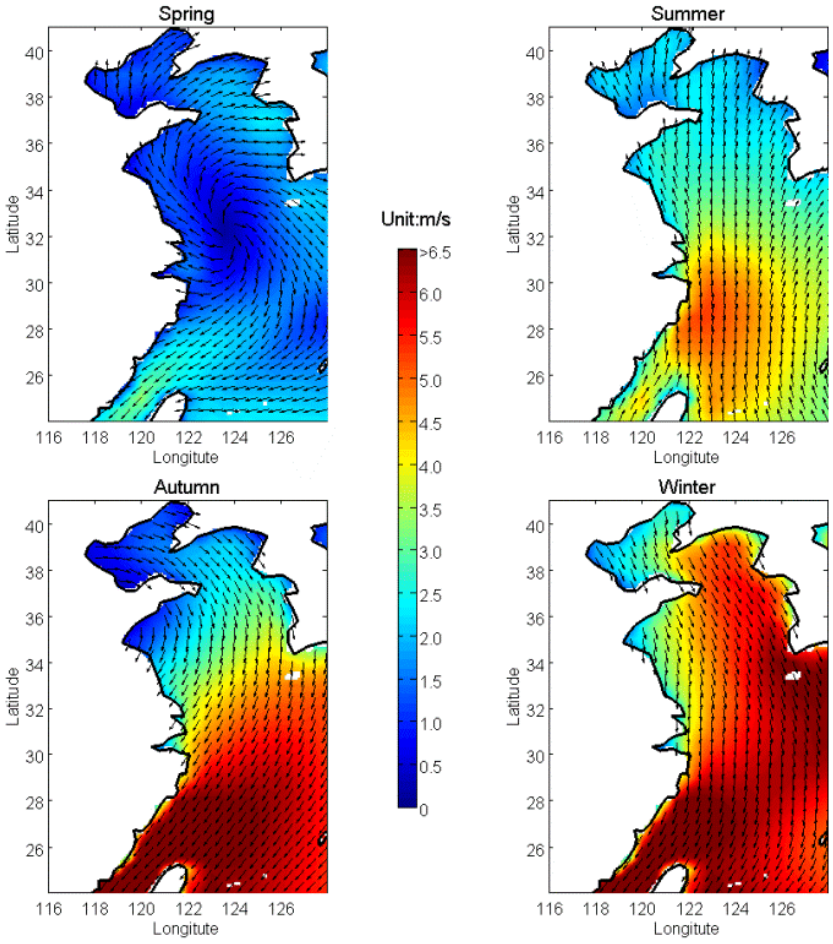


Figure 2: Seasonal spatial distribution of the wind vector data in 2003-2013

2 Literature review

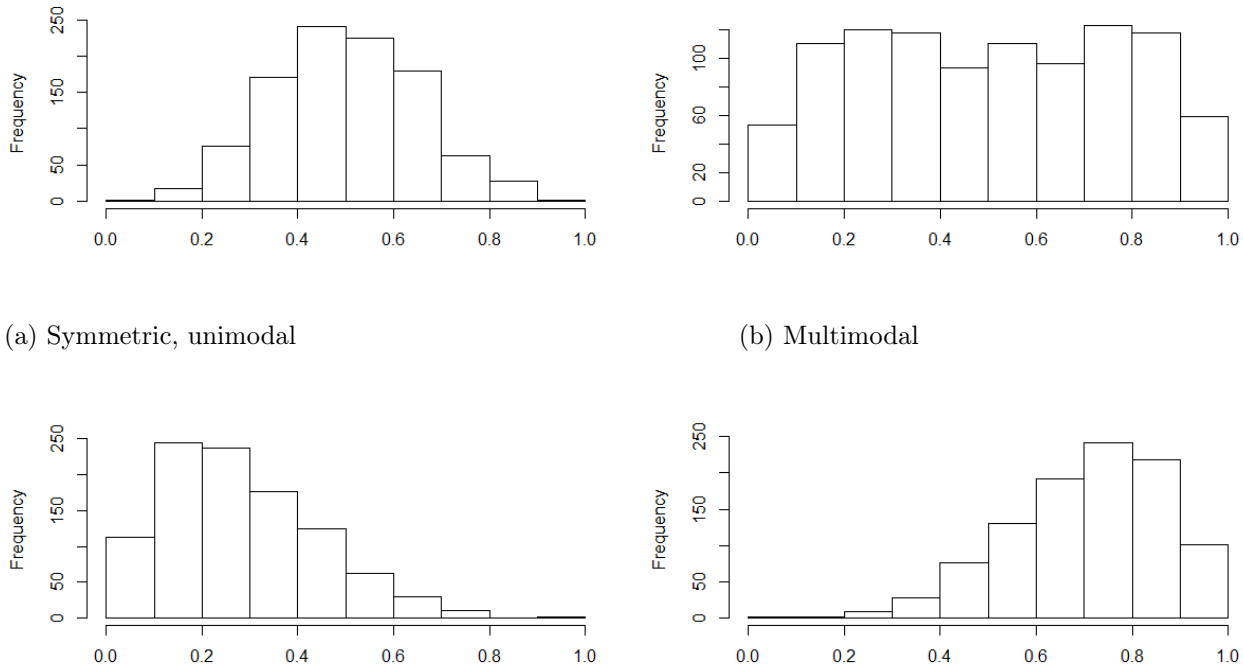
Several techniques exist for analyzing spatial data and more specifically spatial densities. Some of the earlier methods for spatial point pattern analysis was developed by Peter J. Diggle in a monograph published in 1983 and other works such as [8], which paid particular attention to areas in biology and plant ecology. The methods developed had other applications in fields such as geography and environmental sciences. A revised edition of these works was later published which refined the methods utilized to analyze spatial data [7]. This included nonparametric methods for spatial intensity estimation. In the same publication, Diggle lent more emphasis to the mathematical concepts behind the statistical analysis of spatial density estimation [7]. Other statisticians such as Ripley [12] and Upton and Fingleton [15] have surveyed tailored methods for the application of spatial data analysis. In their joint works, Upton and Fingleton [15] consider methods for spatial intensity estimation. With innovations in technology over the past decade, computer based methods have facilitated the exploration of spatial data and has spurred the development of spatial computing software by authors such Baddeley and Turner [1, 2, 3] for spatial intensity estimates for point pattern data. There are a number of spatial analysis software tools available that aid in the analysis of spatial point data and the estimation of kernel smoothed intensities. Among them are software packages such as `spatstat`, `clusterpy` and `LuciadLightspeed` .

3 Kernel density estimation in one dimension

3.1 Histograms

A histogram is one of several graphical methods that is used to show basic information about a data set. This may include, but is not limited to, measures of location such as the median data value, the mean value of the sample data, and the spread of the data over a domain. A histogram can give information about the shape of a probability distribution, namely the presence of symmetry or skewness, and unimodal or multimodal classes of data. Figure 3 illustrates some examples of the shape that a histogram created from data may have. The bell shape of (a) is characteristic of a symmetric data set, (b) illustrates a typical multimodal data set, and skew data will exhibit traits illustrated by (c) and (d).

The histogram is the oldest and most widely used density estimator that equips researchers with practical techniques for the statistical analysis of data. An interval covered by the data set is segmented into sub-intervals $[x_0 + mh,\, x_0 + (m+1)h\}$, that are termed bins, where h denotes the bin width, x_0 a point of origin and m an integer. The height of a bar on a histogram is a positive integer x^* that denotes the number of X_i data points that are in the that bin. If we have that $x_1, x_2, ..., x_k$ are the sample values observed for the random variable X with unknown density $f(x)$, where n denotes the sample size, the



(a) Symmetric, unimodal

(b) Multimodal

(c) Skewed right

(d) Skewed left

Figure 3: Illustration of histograms created from simulated data sets.

histogram is then defined by

$$\hat{f}(x) = \frac{(\text{number of } X_i \text{ in the same bin as } x)}{nh}. \tag{1}$$

There are features of a histogram that can be observed from its construction. The height of a bar is determined by the number of values observed in a bin and, depending on the bin width, an observation can be closer to an observation in the neighboring bin than it is to points in its own bin. Some disadvantages of using a histogram as a graphical method to display data are:

- A histogram is not smooth
- A histogram depends on the end points and the width of the bins.

3.1.1 Example

Suppose that a data set with $n = 10$ observations illustrated in Table adjust=11 is sampled.

i	1	2	3	4	5	6	7	8	9	10
x_i	4	8	35	25	73	68	91	5.4	36	55

Table 1: A data set

Consider two bin widths $h_1 = 10$ and $h_2 = 20$. Using Equation 1 and the data set in Table 1, it follows that

$$\hat{f}_1(x) = \frac{(\text{number of } X_i \text{ in the same bin as } x)}{nh_1}. = \begin{cases} 0.03 & \text{if } 0 < x \leq 10 \\ 0.01 & \text{if } 20 < x \leq 30 \text{ and } 50 < x \leq 80 \\ 0.02 & \text{if } 30 < x \leq 40 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\hat{f}_2(x) = \frac{(\text{number of } X_i \text{ in the same bin as } x)}{nh_2}. = \begin{cases} 0.015 & \text{if } 0 < x \leq 40 \\ 0.005 & \text{if } 40 < x \leq 60 \text{ and } 80 < x \leq 100 \\ 0.01 & \text{if } 60 < x \leq 80 \\ 0 & \text{otherwise} \end{cases}.$$

If we let $x = 25$ then

$$\hat{f}_1(25) = \frac{(\text{number of } X_i \text{ in the same bin as } 25)}{(10)(10)} = 0.01$$

and

$$\hat{f}_2(25) = \frac{(\text{number of } X_i \text{ in the same bin as } 25)}{(10)(20)} = 0.015.$$

3.2 Weighted functions

Consider the continuous random variable, X, from a distribution $f(x)$. To determine the probability that the random variable falls with in a specific interval $(x - h, x + h)$ we would use the expression

$$P(x - h < X < x + h) = \int_{x-h}^{x+h} f(t)dt \approx 2hf(x).$$

Thus

$$f(x) \approx \frac{1}{2h} P(x - h < X < x + h).$$

If we have $x_1, x_2, ..., x_n$ are the sample values observed for the random variable X, this probability can be estimated by

$$\hat{f}(x) = \frac{(\text{number of observations in the interval } (x - h, x + h))}{2nh}. \tag{2}$$

The density estimate $\hat{f}(x)$ can then be expressed as

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} w(x - x_i, h) \tag{3}$$

where

$$w(t,h) = \begin{cases} \frac{1}{2h} & \text{for } |t| < h \\ 0 & \text{otherwise} \end{cases}$$

is a weighting function based on rectangles. Other weighting functions can be based on triangles or of a Gaussian form. Using a weighting function to find an estimate for the density of a population gives an estimate $\hat{f}$ that still retains its property of discontinuity.

3.3 Kernels

The weighting function, $w(t,h)$, in Equation 3 can be expressed more generally as

$$w(t,h) = \frac{1}{h}K\left(\frac{t}{h}\right)$$

where K denotes a function of a single variable termed the kernel, which is essentially a standardized weighting function i.e. $h = 1$, determining the shape.
[9] For our purposes, the kernel is a function that exhibits the following properties:

1. $\int K(t)dt = 1$
2. $\int tK(t)dt = 0$
3. $K(-t) = K(t)$
4. $\int t^2K(t)dt < \infty$

Consequently, the kernel is any non-negative function that is integrable over its whole domain with a value equal to one, centered at zero, symmetric about its center, and has first and second moments that exist. Table 2 gives several examples of kernel functions with the properties describe above.

3.3.1 Example

Consider the cosine kernel $K(t) = \frac{\pi}{4}\cos(\frac{\pi}{2}t)$ for $|t| \leq 1$ and zero otherwise. We note that $K(t) \geq 0$ for all $|t| \leq 1$. It follows that

$$\int_{-\infty}^{\infty} K(t)dt = \int_{-1}^{1} \frac{\pi}{4}\cos\left(\frac{\pi}{2}t\right)dt = \frac{1}{2}\sin\left(\frac{\pi}{2}t\right)\Big|_{-1}^{1} = \frac{1}{2}(1-(-1)) = 1.$$

Kernel	$K(t)$
Epanechnikov	$\frac{3}{4\sqrt{5}}(1-\frac{1}{5}t^2)$ for $\lvert t\rvert < \sqrt{5}$, 0 otherwise
Gaussian	$\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}t^2}$ for all t
Uniform	$\frac{1}{2a}$ for $-a \leq t \leq a$, 0 otherwise
Triangular	$1-\lvert t\rvert$ for $\lvert t\rvert < 1$, 0 otherwise
Cosine	$\frac{\pi}{4}\cos(\frac{\pi}{2}t)$ for $\lvert t\rvert \leq 1$
Logistic	$\frac{1}{e^t+2+e^{-t}}$
Sigmoid function	$\frac{2}{\pi(e^t+e^{-t})}$
Silverman kernel(insert citation)	$\frac{1}{2}e^{-\frac{\lvert t\rvert}{\sqrt{2}}}sin(\frac{4\lvert t\rvert+\sqrt{2}\pi}{\sqrt{36}})$

Table 2: Examples of kernel functions commonly utilized

If we also consider

$$\int_{-\infty}^{\infty} tK(t)dt = \int_{-1}^{1} t\frac{\pi}{4}\cos\left(\frac{\pi}{2}t\right)dt.$$

Using integration by parts, we get that

$$\int_{-1}^{1} t\frac{\pi}{4}\cos\left(\frac{\pi}{2}t\right)dt = \int_{\mathbb{R}^2} \boldsymbol{t}K(\boldsymbol{t})d\boldsymbol{t} = \boldsymbol{0}t\frac{1}{2}\sin\left(\frac{\pi}{2}t\right)\Big|_{-1}^{1} - \int_{-1}^{1}\frac{1}{2}\sin\left(\frac{\pi}{2}t\right)dt = \frac{1}{2}(1-1)+\frac{1}{\pi}(1-1) = 0.$$

Furthermore

$$K(-t) = \frac{\pi}{4}\cos\left(\frac{\pi}{2}(-t)\right) = \frac{\pi}{4}\cos\left(\frac{\pi}{2}t\right) = K(t),$$

and the last property,

$$\int_{-\infty}^{\infty} t^2K(t)dt = \int_{-1}^{1} t^2\frac{\pi}{4}\cos\left(\frac{\pi}{2}t\right)dt.$$

Using Integration by parts, we get that

$$\int_{-1}^{1} t^2\frac{\pi}{4}\cos\left(\frac{\pi}{2}t\right)dt = \frac{1}{2}t^2\sin\left(\frac{\pi}{2}t\right)\Big|_{-1}^{1} - \left[\frac{2}{\pi}t\cos\left(\frac{\pi}{2}t\right)\Big|_{-1}^{1} + \int_{-1}^{1}\frac{2}{\pi}\cos\left(\frac{\pi}{2}t\right)dt\right] = 1-\frac{8}{\pi^2} < \infty.$$

Thus $K(t)$ satisfies all the requirements for a kernel function.

3.4 Univariate kernel density estimation

In Section 2.1, the use of a histogram as a density estimator was discussed. Some disadvantages of using a histogram as a density estimate were that a histogram is not smooth. To resolve this problem, the kernel density estimator defined by

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{4}$$

may be used, where K denotes the kernel function with the properties listed in Section 2.3, which will determine the shape of the weighting function (i.e. rectangular, triangular, Gaussian, etc.), $x_1, x_2, ..., x_n$ are the sample values observed for the random variable X, and h, the bandwidth or smoothing parameter, which determines the width of the weighting function. The choice of h, the bandwidth, is crucial in estimating the kernel smoothed intensity for a population. A large choice for h will mask the structure of the data and over-smooth the density estimate, whereas, a small h will under-smooth the density estimate. The effects of the choice of bandwidth are illustrated in Figure 4[3].

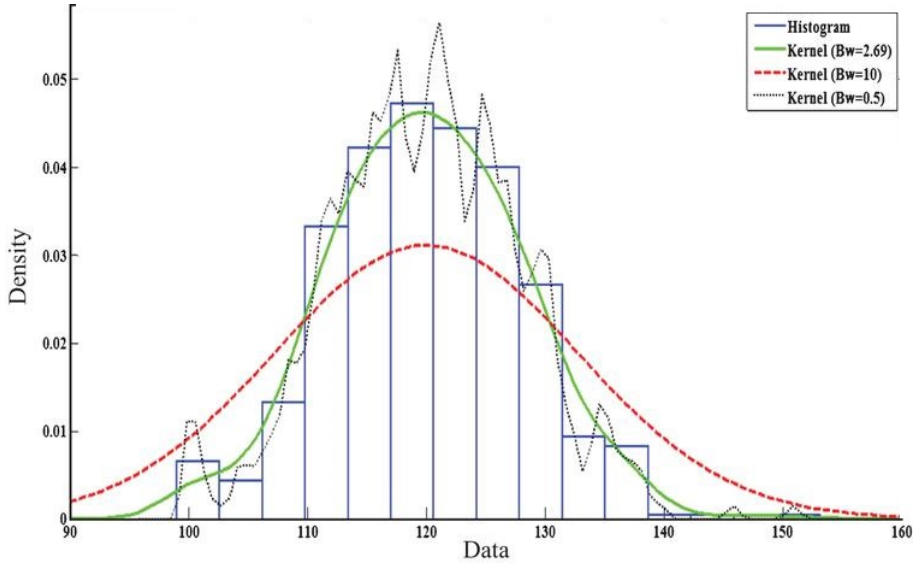


Figure 4: Density estimation of output power (MW) for unit 1 at hour 11 with different bandwidths.

When the choice of bandwidth is relatively small (i.e. $h = 0.5$), the kernel density estimate will fit the data adeptly, but the density function would be jagged and spiked. If we choose a large bandwidth, say $h = 10$, we get a smooth function for the density estimate. Even though the kernel density estimate, for a bandwidth of size $h = 10$, gives a smooth estimate for the probability density function, the estimate deviates considerably from the distribution of the data. A good choice for h would be a value that gives

[3]https://www.google.co.za/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjJ1sWV-6vTAhUF0RQKHUUJBj8QjRwIBw&url=http%3A%2F%2Fcontent.iospress.com%2Farticles%2Fjournal-of-intelligent-and-fuzzy-systems%2Fifs2149&psig=AFQjCNEstJnJgM2g8b1udrn19JxWmDb1Zg&ust=1492534672751762. Date accessed: 17 April 2017.

a smooth estimate for the density function and that emulates the distribution of the data.

4 Kernel density estimation in two dimensions

The expression for the kernel density estimate in Equation 4 can be extended to the multivariate case. That is, instead of estimating the density function for a random variable X by using sample data, $x_1, x_2, ..., x_n$, we can estimate the density function for a k-dimensional random vector, $\boldsymbol{X}$, by using a random sample $\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_n$, where

$$\boldsymbol{X} = \begin{bmatrix} X_1 \\ X_2 \\ ... \\ X_k \end{bmatrix} \text{ and } \mathrm{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ ... \\ x_{ik} \end{bmatrix}, \; i = 1, 2, ..., n.$$

[9] In general, the form of the k-dimensional multivariate kernel density estimator for a random sample $\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_n$ is

$$\hat{f}_{\mathbf{H}}(\mathrm{x}) = n^{-1} \sum_{i=1}^{n} K_{\mathrm{H}}\Big(\mathrm{x} - \mathrm{x}_i\Big), \tag{5}$$

where

- $K_{\mathrm{H}}(\mathrm{x}) = |\mathbf{H}|^{-1/2} K(\mathbf{H}^{-1/2}\mathrm{x})$, and $K : \mathbb{R}^{\mathrm{k}} \to \mathbb{R}$, is a kernel function that takes k arguments.
- n is the number of k-dimensional vectors observed.
- H is a $k \times k$ bandwidth matrix, which is fixed, symmetric and positive definite. As in the univariate case, the level of smoothing of the kernel density function is vastly determined by the bandwidth matrix. Depending on the choice of bandwidth matrix, the kernel density function can either be too smooth and inaccurately represent the data, under-smoothed and representative of the data, or smooth and a good fit for the data set. There are two main cases of the forms that the bandwidth matrix can assume:

  1. The bandwidth matrix is symmetric and positive definite with the form

$$\mathrm{H} = \begin{bmatrix} h_1^2 & h_{12} & \cdots & h_{1k} \\ h_{12} & h_2^2 & \cdots & h_{2k} \\ \cdots & \cdots & \ddots & \cdots \\ h_{1k} & h_{2k} & \cdots & h_k^2 \end{bmatrix}.$$

2. The bandwidth matrix is diagonal, and positive definite with the form

$$diag\mathrm{H} = \begin{bmatrix} h_1^2 & 0 & \cdots & 0 \\ 0 & h_2^2 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & h_k^2 \end{bmatrix}.$$

If we let $h_1 = h_2 = \ldots = h_k$, the bandwidth reduces to

$$\mathrm{H} = h^2\mathrm{I}_k = \begin{bmatrix} h^2 & 0 & \cdots & 0 \\ 0 & h^2 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & h^2 \end{bmatrix},$$

where I_k denotes a $k \times k$ identity matrix. For this case of the bandwidth matrix, Equation 5 reduces to

$$\hat{f}_{\mathbf{H}}(\mathrm{x}) = \frac{1}{nh^k}\sum_{i=1}^{n} K\left(\frac{\mathrm{x} - \mathrm{x}_{\boldsymbol{i}}}{h}\right) = \frac{1}{nh^k}\sum_{i=1}^{n} K\left(\frac{x_1 - x_{i1}}{h}, \frac{x_2 - x_{i2}}{h}, ..., \frac{x_k - X_{ik}}{h}\right).$$

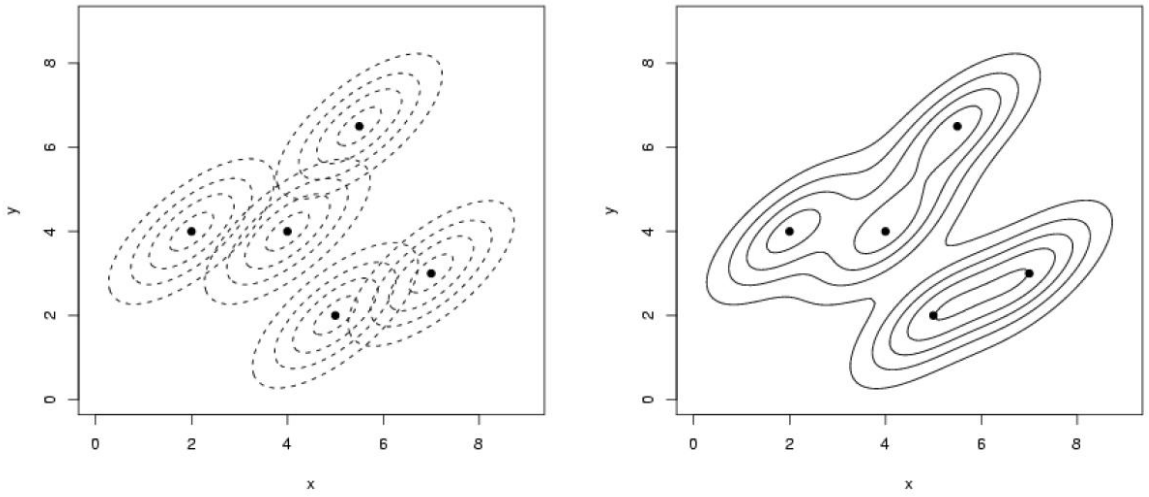
If we let $k = 1$, the we get the form of the univariate kernel density estimate shown in Equation 4.

In the rest of the report we will only consider the bivariate case, where $k = 2$, and the case where $\mathrm{H} = h^2\mathrm{I}_2$, of the bandwidth matrix. Equation 5 thus simplifies to

$$\hat{f}_{\mathbf{H}}(\mathrm{x}) = \frac{1}{nh^2}\sum_{i=1}^{n} K\left(\frac{\mathrm{x} - \mathrm{x}_{\boldsymbol{i}}}{h}\right) = \frac{1}{nh^2}\sum_{i=1}^{n} K\left(\frac{x_1 - x_{i1}}{h}, \frac{x_2 - x_{i2}}{h}\right).$$

Example

This example was adapted from [10]. We consider the bivariate case with bandwidth matrix $\mathrm{H} = \begin{bmatrix} h_1^2 & h_{12} \\ h_{12} & h_2^2 \end{bmatrix}$, where $h_1 = h_2$. We have a sample data set $\mathrm{x}_1 = (7, 3)$, $\mathrm{x}_2 = (2, 4)$, $\mathrm{x}_3 = (4, 4)$, $\mathrm{x}_4 = (5, 2)$ and $\mathrm{x}_5 = (5.5, 6.5)$ with a bandwidth matrix $\mathrm{H} = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$. Figure 5 taken from [10] depicts a plot of the points on a Cartesian plane with kernel functions fitted over each point, shown on the left and the kernel density estimate for the data set which is shown on the right.



(a) (b)

Figure 5: Bivariate kernel density estimate. (a), individual kernels. (b), kernel density estimate

The kernel function K still retains most of its properties when applied to kernel density estimation for the bivariate case, namely that it is a non-negative function, $K(\boldsymbol{t}) \geq 0$, such that

- $\int_{\mathbb{R}^2} K(\boldsymbol{t}) d\boldsymbol{t} = 1$
- $\int_{\mathbb{R}^2} \boldsymbol{t} K(\boldsymbol{t}) d\boldsymbol{t} = \mathbf{0}$

where $\boldsymbol{t}' = [t_1\, t_2]$, a 2×1 vector and $\mathbf{0}$, the zero vector.

5 Density.ppp

In the introduction of this report, a point pattern was defined as an arrangement of objects that are distributed in a region. A point may be represented as coordinates, (x, y), over a specified geographical region R, that can assume the form of any complex polygon or nonconvex shape. The method of kernel density estimation can be deployed to compute a kenel density estimate from the point pattern data. To this end, the `spatstat`[3, 2] package in `R` may be used, namely the `density.ppp` function. The `density.ppp` is a function in the `spatstat` package in `R` that computes a kernel smoothed intensity from spatial point data. It is a method for the generic command `density`. The `density.ppp` function can take a considerable number of parameters with some set defaults. The parameters may include additional non-compulsory arguments that can be excluded depending on validity and applicability to the data.

The density.ppp takes arguments and defaults shown in Equation 6.

$$\begin{aligned} \texttt{density.ppp}(\texttt{x}, \texttt{sigma} = \texttt{NULL}, ..., \texttt{weights} = \texttt{NULL}, \texttt{edge} = \texttt{TRUE}, \texttt{varcov} = \texttt{NULL}, \texttt{at} = \text{"}\texttt{pixels}\text{"}, \\ \texttt{leaveoneout} = \texttt{TRUE}, \texttt{adjust} = 1, \texttt{diggle} = \texttt{FALSE}, \texttt{se} = \texttt{FALSE}, \texttt{kernel} = \text{"}\texttt{gaussian}\text{"}, \\ \texttt{scalekernel} = \texttt{is.character}(\texttt{kernel}), \texttt{positive} = \texttt{FALSE}) \end{aligned} \tag{6}$$

In spatstat a point pattern is denoted by the object class "ppp" (planar point pattern). This is the first argument, x, that should be read into the density.ppp function. The function will use these collection of data points to calculate the estimate for the population intensity. Figure 6 is a plot of point pattern data simulated in R. This data will be used in the rest of the section to illustrate how the density.ppp can be used to find kernel estimates.

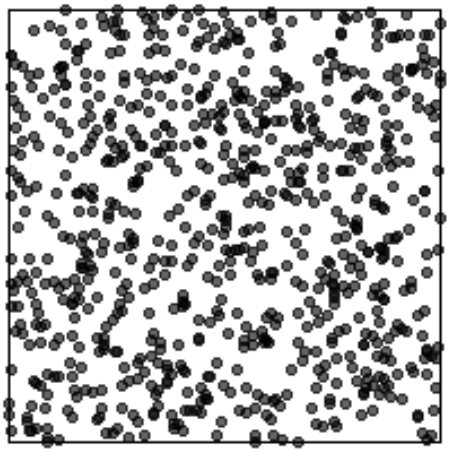


Figure 6: Point pattern plotted using simulated data in R

5.1 edge and diggle

The intensity estimates at the data points are computed using the following formulas,

$$\hat{\lambda}^{(1)}(\mathbf{x}) = \sum_{i=1}^{n} K(\mathbf{x} - \mathbf{x_i}), \tag{7}$$

$$\hat{\lambda}^{(2)}(\mathbf{x}) = \frac{1}{e(\mathbf{x})} \sum_{i=1}^{n} K(\mathbf{x} - \mathbf{x_i}), \tag{8}$$

$$\hat{\lambda}^{(3)}(\mathbf{x}) = \sum_{i=1}^{n} \frac{1}{e(\mathbf{x}_i)} K(\mathbf{x} - \mathbf{x_i}) \tag{9}$$

where $K(\boldsymbol{t})$ is the kernel function,

$$e(\mathbf{x}) = \int_R K(\mathbf{x} - \mathbf{v}) d\mathrm{v} \tag{10}$$

is the correction for the bias due to edge effects. Equations 7, 8 and 9 denote the kernel density estimates for the uncorrected, uniformly corrected and Diggle corrected kernel estimates respectively. When selecting a window R, the region of study, we create a boundary in which points in a pattern are observed. The observed points in the region R may be a subset of a larger region of points in a population. Points that fall outside this boundary should have no or minimal interdependence with observations in the window. Edge effects arise when there is some interaction between points inside the window of study and points that fall outside the region [7]. When calculating the smoothed kernel intensity, we account for edge effects by normalizing the density estimate with the function denoted in Equation 10. Points outside the boundary of the region of observation do not contribute to the sum of kernel intensity estimates. In the presence of edge effects, the uncorrected estimate in Equation 7 will decrease close to the boundary of the region of observation since the kernels over points close to the boundary have fewer contributions in the sum of the intensity estimate [2].

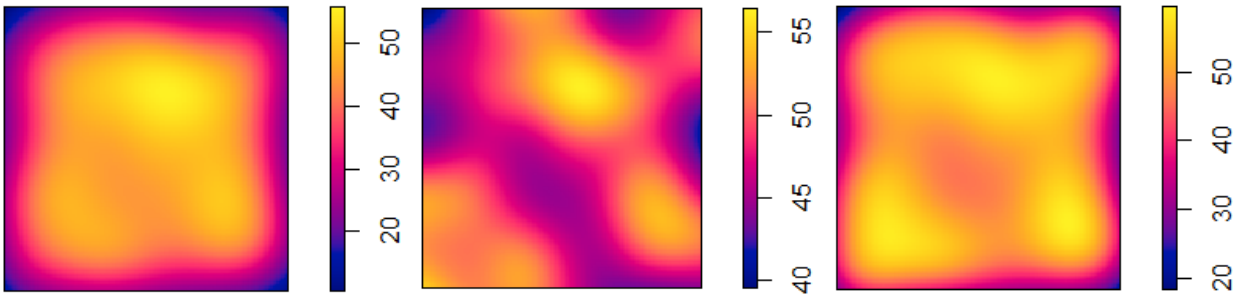
In the expressions above, Equation 7 is an instance of the kernel estimate that is not corrected for edge effects whereas Equations 8 and 9 correct for this. Equations 9 and 8, the Diggles and uniformly corrected estimates respectively, differ in that the Diggles correction attaches variable weights determined by Equation 10, where the function parameter is adjusted for each point, to each kernel fitted over a point and the sum of these weighted kernels are then used to compute the smoothed intensity estimate. The uniformly corrected estimate weights each kernel fitted over a point by a constant value also determined by the expression in Equation 10. The contribution to the sum of each point kernel is uniformly weighted.

Whether the kernel smoothed intensity is corrected for edge effects can be controlled by adjusting the `edge` parameter of the `density.ppp` function. If `edge` = `FALSE` the form of the expression in Equation 7 will be used to compute a kernel density estimate. If `edge` = `TRUE`, either one of the expression in Equations 8 and 9 will be utilized.

Which formula is used for the edge correction is determined by the argument in `diggle` parameter of the `density.ppp` function which can take on the Boolean values `TRUE` or `FALSE`. When the `diggle` parameter is set to its default value, `FALSE`, Equation 8 will be applied to compute a kernel smoothed intensity. Alternatively, if the parameter value is set to `TRUE`, Equation 9 will be used.

Figure 7 illustrates the different plots for the kernel density estimate when the `edge` and `diggle` options are adjusted. If `edge` = `FALSE` , we get the output in (a) which corresponds to the kernel density estimate calculated using the expression in Equation 7. When the logical value `TRUE` is assigned to the `edge` argument and the `diggle` parameter is not specified, the output in (b) corresponding to the function

for the kernel density estimate in Equation 8 will be given. Since the default for `diggle` is `FALSE`, when the logical value `FALSE` is assigned to `diggle` and `edge` = `TRUE` we would still get the same output given in (b). The output given in (c), is the case for `diggle` = `TRUE`, which corresponds to a smoothed kernel estimate that takes the form of the expression given in Equation 9.



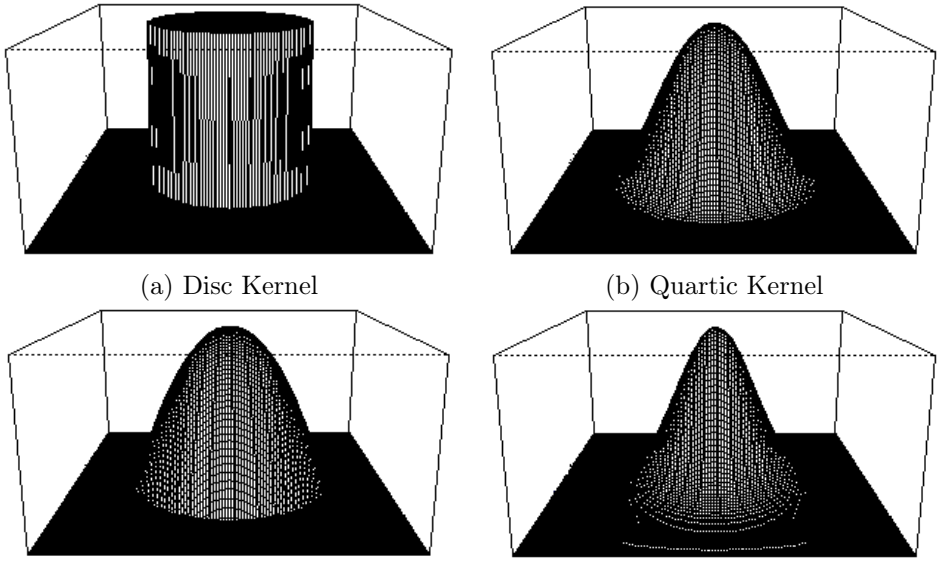
(a) `edge` = `FALSE` (b) `edge` = `TRUE` (c) `edge` = `TRUE`, `diggle` = `TRUE`

Figure 7: Kernel smoothed estimates for the `density.ppp` function of simulated data in `R` with variable options for `edge` and `diggle` for the Guassian kernel.

5.2 `kernel`

By default, the `density.ppp` function uses the isotropic Gaussian kernel to compute smoothed intensity estimates. The available kernel functions present in the `spatstat` package are epanechnikov, quartic, gaussian and disc. The choice of kernel function can be changed by specifying either one of the available kernel options (i.e. "`epanechnikov`","`quartic`","`gaussian`","`disc`") as a character string in the `kernel` argument of the `density.ppp` function.

Figure 23 shows different plots for density estimates when the available kernel options are changed. Depicted in Figure 8, is the graphical images of the shape of different kernel options.



(a) Disc Kernel (b) Quartic Kernel

(c) Epanechnikov Kernel (d) Gaussian Kernel

Figure 8: Graphical images of the shape of the different kernels

5.3 varcov

The varcov option is used to specify the covariance matrix of any Gaussian kernel. That is, a matrix of the form

$$\begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$$

can be assigned to the varcov parameter of the density.ppp function, where the diagonal entries denote the variance of the random variable and the off diagonal entries the covariances between random variables.

To illustrate the functionality of this parameter we examine several forms of covariance matrix. To this end, we plot a single point in a standard rectangular window and fit various kernel functions with different options for the varcov argument. We will only consider the plots for the gaussian,epanechnikov and the disc kernels. The results will follow similarly for other available kernels.

5.3.1 Diagonal positive definite matrix

Suppose we have a positive definite matrix that assumes the form

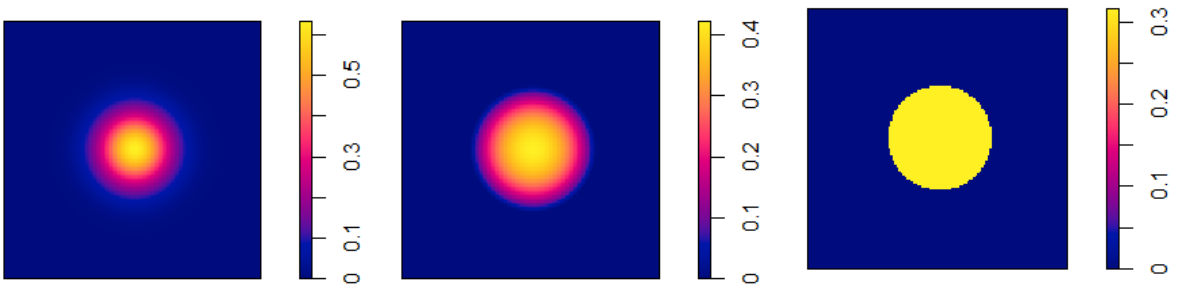
$$\begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}.$$

There are three main forms that this matrix can take.

Case:

- $\sigma_1 = \sigma_2$

We use a matrix $H_1 = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}$ that has this property. Figure 9 illustrates the shape that the guassian, epanechnikov and disc kernels will assume for this matrix when it is specified in the varcov argument. For all the listed kernels, any diagonal matrix with equal diagonal elements will produce isotropic shaped kernels.

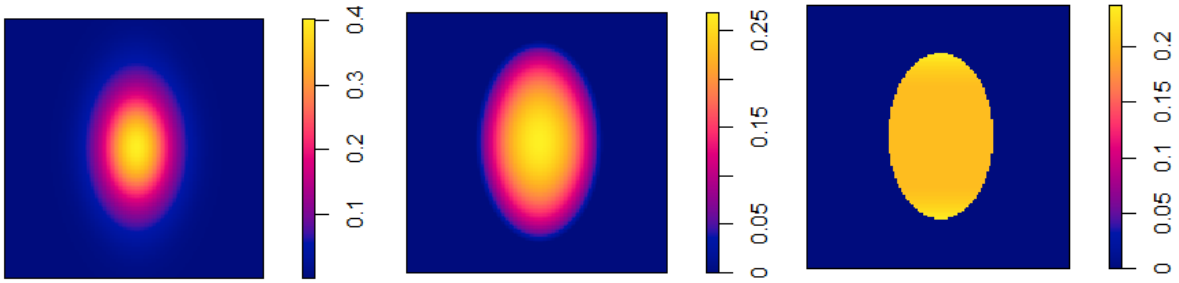


(a) Gaussian Kernel (b) Epanechnikov Kernel (c) Disc Kernel

Figure 9: Density plot for different kernels fitted over a point for matrix H_1

- $\sigma_1 < \sigma_2$

In the following example a matrix $H_2 = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.625 \end{bmatrix}$ was defined in the `varcov` parameter of the `density.ppp` function. Figure 10 depicts the shape of the kernels when the matrix assumes this form. For this instance of the matrix, the kernel functions have a vertical width larger than their horizontal counterpart.

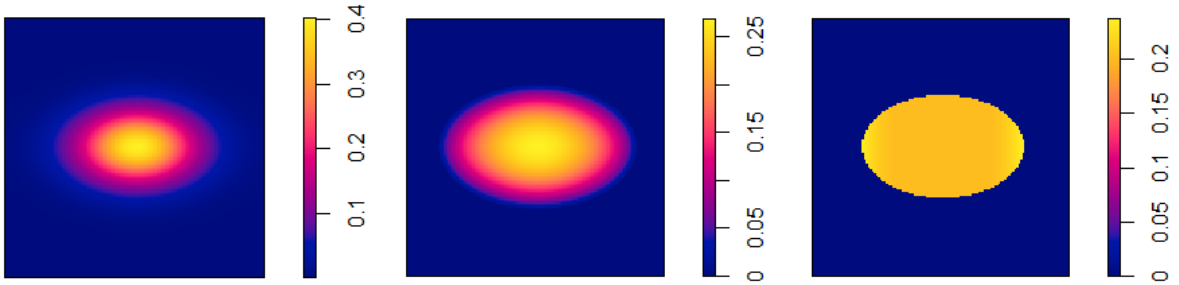


(a) Gaussian Kernel (b) Epanechnikov Kernel (c) Disc Kernel

Figure 10: Density plot for different kernels fitted over a point for matrix H_2

- $\sigma_1 > \sigma_2$

Similarly, as in the previous case, we specify a matrix that has one diagonal entry larger than the other for σ_1 and σ_2 respectively, namely $H_3 = \begin{bmatrix} 0.625 & 0 \\ 0 & 0.25 \end{bmatrix}$, in the `varcov` argument and produce the output illustrated in Figure 11. We again observe a larger width in the direction of the largest diagonal element specified, the horizontal direction.

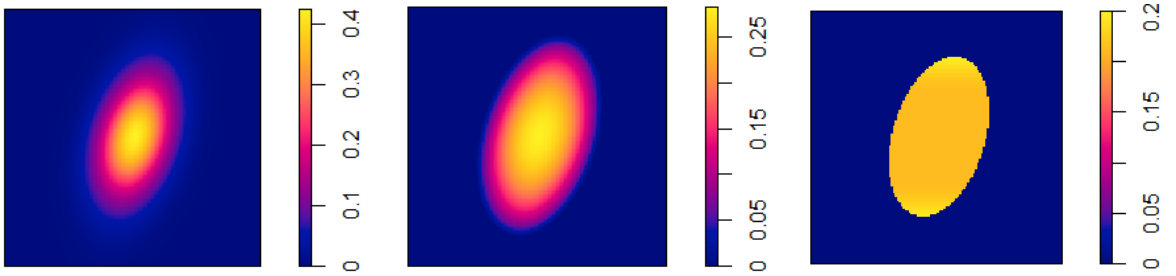


(a) Gaussian Kernel (b) Epanechnikov Kernel (c) Disc Kernel

Figure 11: Density plot for different kernels fitted over a point for bandwidth matrix H_3

5.3.2 Symmetric, positive definite matrix

To depict the kernel shape for a symmetric positive definite matrix which has either negative or positive off diagonal elements, we assign a matrix $H_4 = \begin{bmatrix} 0.25 & 0.125 \\ 0.125 & 0.625 \end{bmatrix}$ and $H_5 = \begin{bmatrix} 0.25 & -0.125 \\ -0.125 & 0.625 \end{bmatrix}$ for the cases $\sigma_{12} > 0$ and $\sigma <_{12} 0$ respectively to the `varcov` argument and plot a density for different kernel options illustrated in Figure 12 and Figure 13.



(a) Guassian Kernel (b) Epanechnikov Kernel (c) Disc Kernel

Figure 12: Density plot for different kernels fitted over a point for matrix H_4

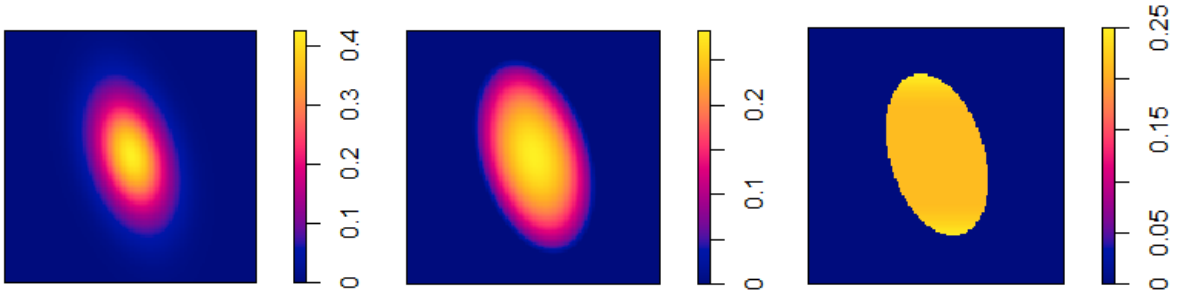


Figure 13: Density plot for different kernels fitted over a point for matrix H_5

5.3.3 Nonsymmetric covariance matrices

We chose several non-symmetric matrices with positive diagonal entries, $H_6 = \begin{bmatrix} 0.25 & -0.125 \\ 0.1 & 0.625 \end{bmatrix}$, $H_7 = \begin{bmatrix} 0.25 & 0.1 \\ -0.125 & 0.625 \end{bmatrix}$, and $H_8 = \begin{bmatrix} 0.25 & 0.3 \\ 0.1 & 0.625 \end{bmatrix}$ and create a density plot for different kernel options portrayed in Figure 14.

5.4 `sigma`

The bandwidth for the kernel can be specified in the `sigma` parameter option of the function and can be determined in one of several ways.

- A value can be directly assigned to the parameter by stating, `sigma` = `value`. The numerical value assigned to this parameter will be taken as the standard deviation of the Gaussian kernel. That is, the value will be used as both the bandwidth and the standard deviation of the Gaussian kernel. In general, the Gaussian kernel depends on two parameters namely the mean and variance. The value assigned to `sigma` accounts for the variance parameter for the Gaussian kernel. Since the kernel has the property of symmetry about zero, the mean parameter for the Gaussian kernel is given as zero. For the other available kernels (i.e. `epanechnikov`, `quartic`, `disc`), `sigma` is taken only as the bandwidth since no other function parameters need to be specified for these kernel functions.

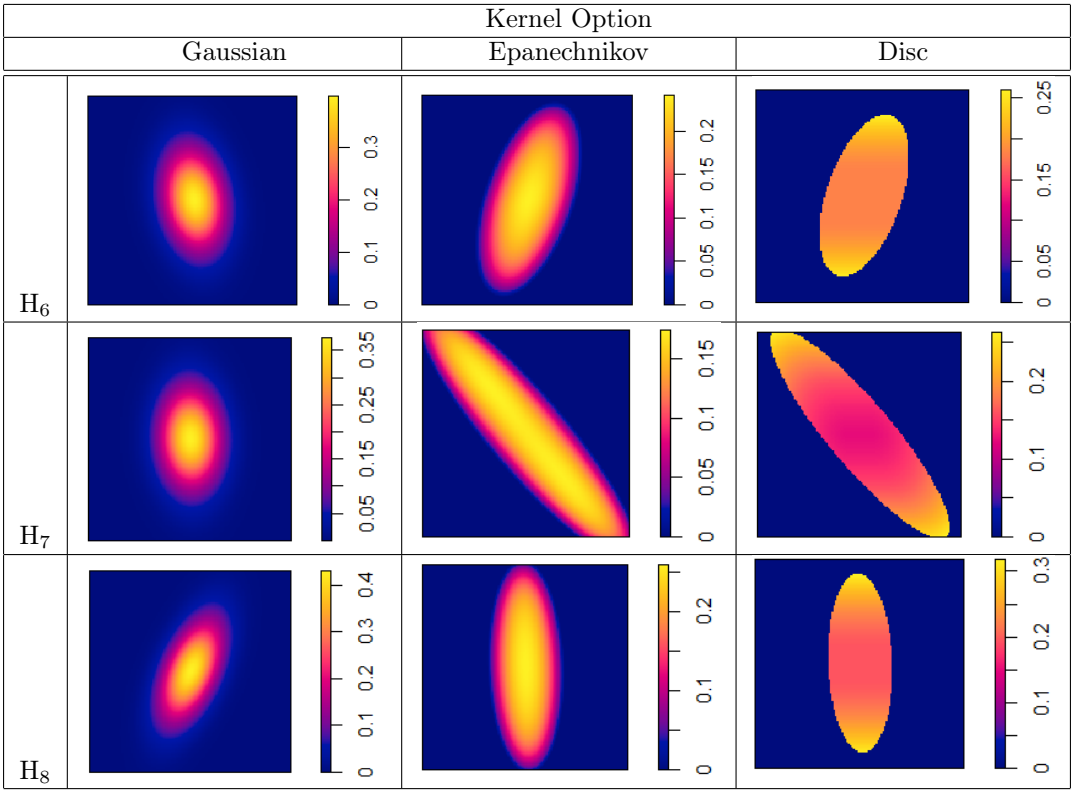


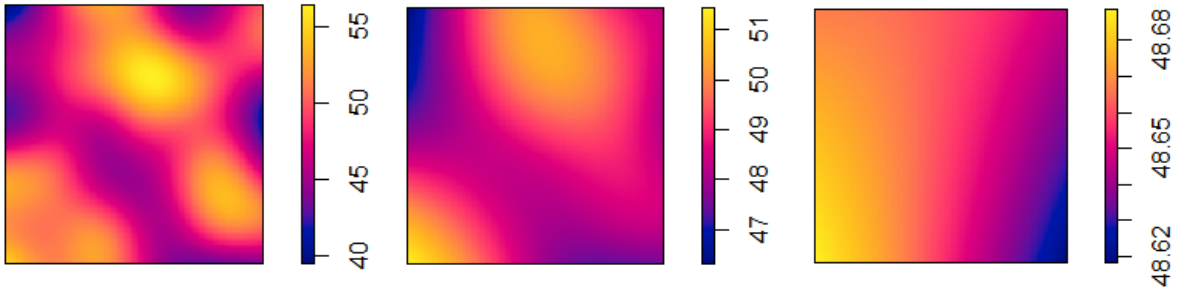
Figure 14: Density plot for different kernels fitted over a point for a non-symmetric matrix with positive diagonal entries

- For the default kernel, the Gaussian isotropic kernel, `sigma(x)` can be called in the parameter as a function and used to calculate a suitable bandwidth from the point pattern data described in the first function parameter, `x`, of the `density.ppp` function. The value of `sigma` can be calculated using various functions. These include `bw.diggle`, `bw.scott`, `bw.ppl` and `bw.frac` for Diggle and Berman's mean square error cross-validation method, likelihood cross validation method, Scott's rule of thumb and a fast rule of thumb based on the shape of the window respectively. Generally, the `sigma(x)` function calculates the standard deviation of an `R` object `x` , stated in the function parameter when it is called.

- Another available option for the sigma parameter would be to use a vector of length 2 that gives that standard deviation of two independent Gaussian coordinates.

- When `sigma` is not specified, it is calculated as one eighth of the shortest side length of the enclosing rectangle for the point pattern. `sigma` and `varcov` are not compatible since either one of these options has to be used to specify the variance or covariance matrix of the Gaussian kernel.

Figure 15 illustrates the different plots for the kernel smoothed intensity when the value of `sigma` is adjusted.



(a) `sigma` = 0.5 (b) `sigma` = 1 (c) `sigma` = 5

Figure 15: Plot of kernel smoothed intensity estimates for the `density.ppp` function for simulated data in `R` with different values for `sigma` for the default Gaussian kernel

5.5 `adjust`

The `adjust` argument is used in conjunction with the `sigma` option to control for the change in bandwidth. The value assigned to `sigma` will be multiplied by the factor `adjust`. The default for the `adjust` argument is the numeric value one.

5.6 `leaveoneout`

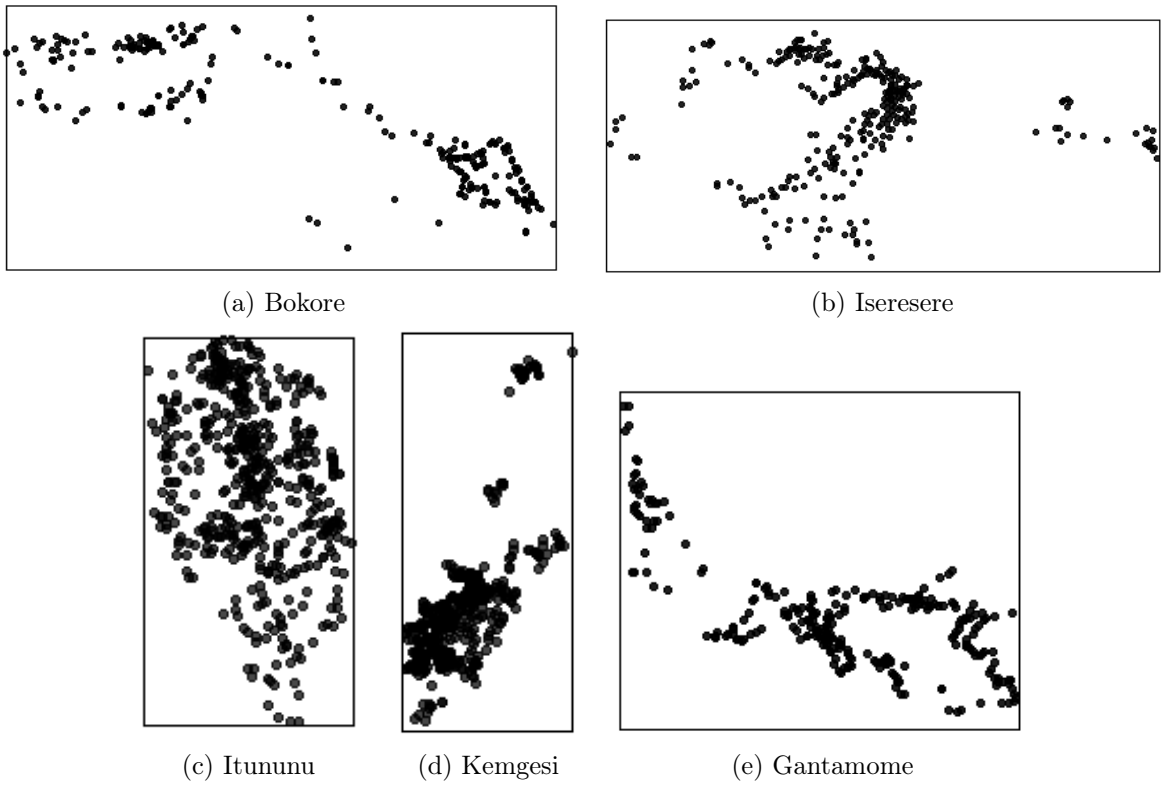
The `leaveoneout` argument in the function parameter of `density.ppp` takes on the logical values `TRUE` or `FALSE`. If `leaveoneout` is set to `TRUE`, the sum in the equation of the kernel intensity is taken over all points not equal to a coordinate in the window of observation, that is for Equations 7, 8 and 9 the sum is calculated over all points $\mathbf{x}_i$ not equal to $\mathbf{x}$, thus the intensity value at a point is the sum of kernel contributions from other data points. When `leaveoneout` = `FALSE`, the sum is taken over all $\mathbf{x}_i$ including those equal to $\mathbf{x}$.

6 Application

6.1 Introduction

The data used to illustrate the functionality of the `density.ppp` function was collected in Mara province, situated in Northern Tanzania[4,5]. The census data comprises of data for 34253 households spread across 78 villages. A subset of the data available for a total of 8343 households and 18 villages was extracted from the original census. For the purposes of this report a total of 5 villages, namely Bokore, Gantamome, Iseresere, Itununu and Kemgesi with households that number 269, 265, 295, 534 and 519 respectively will be utilized. Figure 16 depicts a point pattern plot of the geographic location of households for each of the villages listed with variable rectangular windows.

[4] http://www.gla.ac.uk/researchinstitutes/bahcm/staff/katiehampson/
[5] http://www.katiehampson.com/#intro



(a) Bokore (b) Iseresere

(c) Itununu (d) Kemgesi (e) Gantamome

Figure 16: A point pattern plot of the geographic locations of households for five villages in Mara province, Northern Tanzania.

6.2 Illustration

In Section 4 the form of the `density.ppp` function in `R` that computes a kernel smoothed intensity from spatial point data was given as

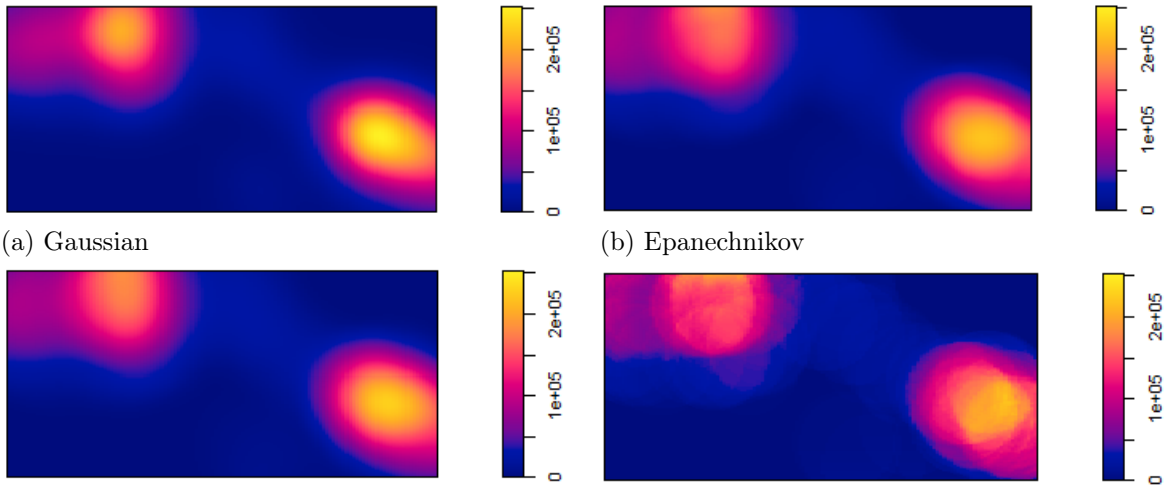
$$\begin{aligned}\texttt{density.ppp}(\texttt{x}, \texttt{sigma} = \texttt{NULL}, ..., \texttt{weights} = \texttt{NULL}, \texttt{edge} = \texttt{TRUE}, \texttt{varcov} = \texttt{NULL}, \texttt{at} = \text{"}\texttt{pixels}\text{"},\\ \texttt{leaveoneout} = \texttt{TRUE}, \texttt{adjust} = 1, \texttt{diggle} = \texttt{FALSE}, \texttt{se} = \texttt{FALSE}, \texttt{kernel} = \text{"}\texttt{gaussian}\text{"}, \quad ,\\ \texttt{scalekernel} = \texttt{is.character}(\texttt{kernel}), \texttt{positive} = \texttt{FALSE})\end{aligned}$$

(i.e. Equation 6). Only objects of type "ppp" can be passed in the argument of this function. The intensity of points is estimated for each village. The latitudinal and longitudinal coordinates with standard rectangular windows denote the point pattern that are passed into the first argument of the `density.ppp` function. The bandwidth is calculated as one eighth of the shortest side length of the enclosing rectangle for the point pattern since no other argument is passed to the function. The default Gaussian kernel is used to compute intensities and the numerical value assigned to the bandwidth is taken as the standard deviation of the Gaussian kernel. Figure 24 in the Appendix shows estimated smoothed intensities and the perspective plots (for the defaults of the `density.ppp` function) of a point pattern of the geographic locations of households for the five villages.

6.3 Standard rectangular window

6.3.1 Kernel

In isolated regions with a low concentration of points, the kernel smoothed estimates barely register any intensity for the defaults of the `density.ppp` function. This is observable in the kernel smoothed estimate plots for the Bokore, Iseresere and the Kemgesi villages listed in the appendix. To alleviate this problem, we adjust the different options in the `density.ppp` function beginning first with the kernel option. In Section 4.1 we noted that the available kernel functions present in the `spatstat` package are the default `gaussian` kernel, and the `epanechnikov`, `quartic` and `disc` kernel functions. Each one is distinct in shape and appearance as illustrated in Figure 8. Figure 25 and 26 in the appendix depict the kernel smoothed intensity estimates of the point pattern of the geographic locations of households for each of the five villages in Mara province, Northern Tanzania for variable kernel options. Figure 17 is an excerpt taken from these figures for Bokore village. For the `disc` kernel option, the kernel smoothed intensity estimate is less smooth. There are noticeable disparities between the kernel smoothed intensity estimates fitted with the `disc` kernel function and that of other kernel options. Intensity estimates fitted with the `quartic` and `epanechnikov` kernels are insignificantly distinguishable from one another.



(a) Gaussian (b) Epanechnikov

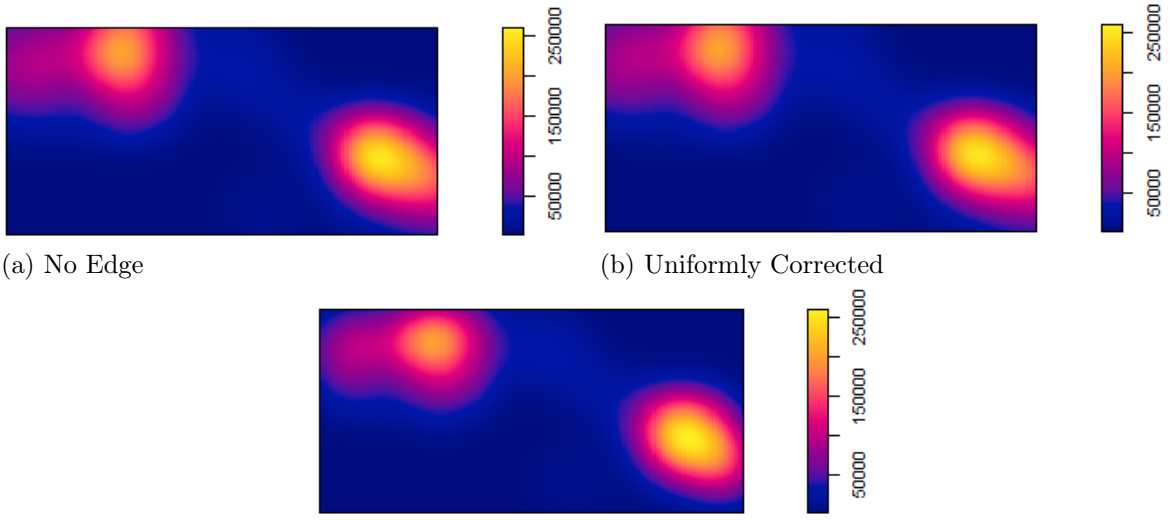
(c) Quartic (d) Disc

Figure 17: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for Bokore village for variable kernel options

6.3.2 Edge Effects

The `density.ppp` function is adjusted for edge effects in the `edge` parameter. To control for the type of edge correction, either Diggle's correction or the uniform edge correction, we specify `TRUE` or `FALSE` in the `diggle` argument of the `density.ppp` function for the respective correction options. Figure 18 depicts the plots for the kernel density estimates adjusted for edge effects for the Bokore village. The plots for the other villages are given in the appendix in Figures 29 and 30.There are no visible contrasts between

the estimated intensity plot of the the uniformly corrected density and that of the densities without the calibration for edge effects. A perceivable difference in the estimated smoothed intensity plot for the Diggle edge correction is the decrease of intensity on the boundary created by the window. At points near the window, the intensity plots seem to curve away from the boundary.



(a) No Edge (b) Uniformly Corrected

(c) Diggle's correction

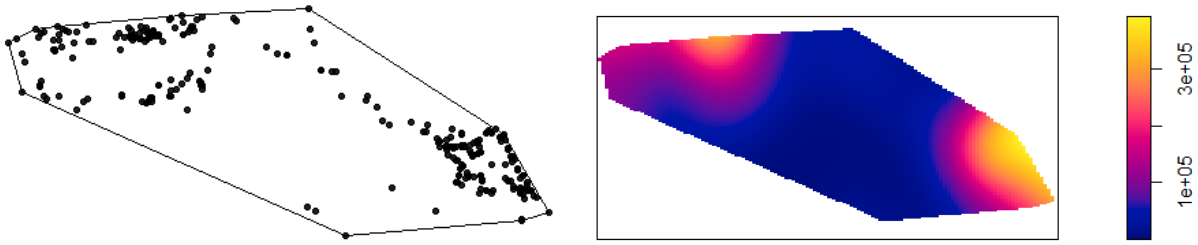
Figure 18: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for Bokore village in Mara province, Northern Tanzania for different edge corrections

6.4 Convex hull

For a set B whose elements comprise of points, the convex hull is defined as the smallest convex shape that contains all the points in the set [6]. For a point pattern process, this would be the smallest convex window that envelopes all the points. To create a convex window as an object of type `owin` in `R` one could call the function `convexhull`. In Figures 31 and 32 the five villages are fitted with a convex hull and the kernel smoothed intensity estimates computed for each via the `density.ppp` function for the default Gaussian kernel. The kernel smoothed intensity estimates for the convex hull registers a higher intensity for regions near the boundary of the window that have a low concentration of points than that initially registered by the standard rectangular window. This can be observed in the estimated density plots illustrated in Figure 19 and additional output in Figures 31 and 32 of the appendix.

6.4.1 Kernel

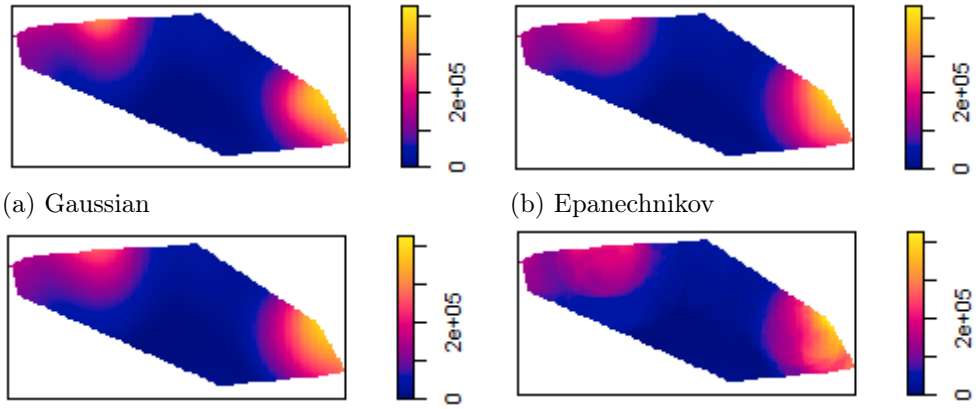
Using the available kernel options, a kernel intensity estimate is computed via the `density.ppp` function for the point patterns contained in a convex window for each village. As in the case of the standard rectangular window, dissimilarities exist in the estimate plots for kernel smoothed intensities fitted with the `disc` kernel function and that of other kernel options as illustrated in Figures 20. Kernel estimate



(a) Point Pattern (b) Kernel Density Estimate

Figure 19: Kernel smoothed intensity estimate of a point pattern of the geographic locations of households for Bokore village in Mara province, Northern Tanzania with convex window for the Gaussian kernel

intensities fitted with a `disc` kernel tend to be less smooth and register lower intensity values when compared to other kernel options. For kernel smoothed intensity estimates fitted with the `gaussian` kernel option, the density estimates register higher intensity values.



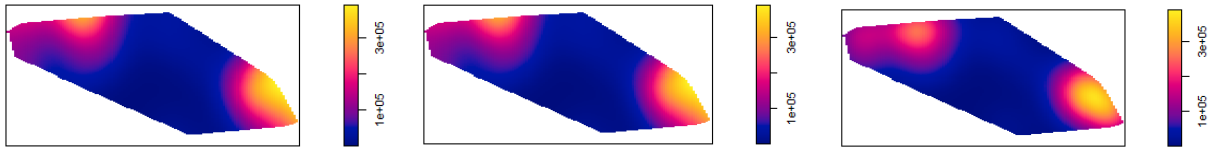
(a) Gaussian (b) Epanechnikov

(b) Quartic (d) Disc

Figure 20: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for Bokore village in Mara province, Northern Tanzania for variable kernel options for a convex hull window

6.4.2 Edge Effects

Figure 21 below and Figures 35 and 36 in the appendix are plots of the kernel smoothed intensity estimates for a convex window adjusted for edge effects for the five villages in the Mara province. There are no visibly significant differences between the estimated intensity plot of the the uniformly corrected density and that of the densities without the correction for edge effects, which was also observable in the case of a standard rectangular window. A perceivable difference in the estimated smoothed intensity plot for the Diggle edge correction is the decrease of intensity on the boundary created by the window. At points near the window, the intensity plots seem to curve away from the boundary.



(a) No correction (b) Uniformly Corrected (c) Diggle's Correction

Figure 21: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for Bokore village in Mara province, Northern Tanzania for different edge corrections for a convex hull window

6.5 Using SAS for kernel density estimation

The method of kernel density estimation can be performed in SAS by initializing the KDE procedure in the PROC statement of a SAS program. The PROC KDE procedure can be applied to univariate and bivariate kernel density estimation. Several statistics which include estimates of the percentiles of the hypothesized probability density function can be enumerated by the procedure. A Gaussian density is implemented as the kernel for the process of kernel density estimation and the variance used as a smoothing parameter. The syntax for the procedure is as follows,

```
PROC KDE < options >;

BIVAR variable − list < /options >;

UNIVAR variable − list < /options >;

BY variables;

FREQ variable;

WEIGHT variable;
```

To illustrate the basic features of PROC KDE, we use the Bokore village to get a kernel density estimate. The longitudinal and corresponding latitudinal values are entered into the DATA step of a SAS program with the name Bokore and the PROC KDE procedure initialized. Illustrated in Figure 22 is the output given for the following lines of code[6],

```
ods graphics on;

proc kde data = Bokore;

bivar latitude longitude/plots = (contour surface);
```

[6]The code and output for this section was generated using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2002-2012 by SAS Institute Inc. SAS and all other SAS Institute Inc, product or service names are registered trademarks or trademarks of SAS Institute Inc., Cary, NC, USA.

```
run;

ods graphics off;
```

In the table titled "Inputs", information about the data set used is listed. This includes the working data set, the number of observations used, the variables contained in the data set and the bandwidth method used to determine the level of smoothing. Contour and surface plots of the density estimates are also displayed. Although a suitable estimation is provided for the kernel density, there is a restriction on the options available in SAS (i.e. the option of adjusting a spatial window or choosing a kernel is not available).

7 Conclusion

In this research report, the method of kernel density estimation for univariate and bivariate data was explored. In the case of bivariate data, the kernel smoothed intesity estimates for a spatial point pattern process were computed via the `density.ppp` function in the `spatstat` package in `R`. The `density.ppp` function and each of its parameters were inspected and their effect on the kernel density estimate analyzed. This ranged from adjusting the kernel options, edge effects and bandwidth selection. We investigated the effect that the window in which objects were studied had on the smoothed intensity estimates and gave particular focus to standard rectangular windows and convex shapes. Using data collected in Mara province, situated in Northern Tanzania, we tested the effect of calibrating the `density.ppp` function for different kernel choices and edge effects for both the standard rectangular window and the convex hull window. There were significant differences observed in the plots of intensity estimates fitted with a `disc` kernel and those of other kernel options which was allotted to the shape of the `disc` kernel function, with plots being less smooth. It was observed that the choice of window also had an influence on the the kernel density estimate and that the convex hull window had intensity plots more representative of the point pattern data. In both instances of the standard rectangular window and the convex hull window the function performance was the same when adjusted for edge effects.

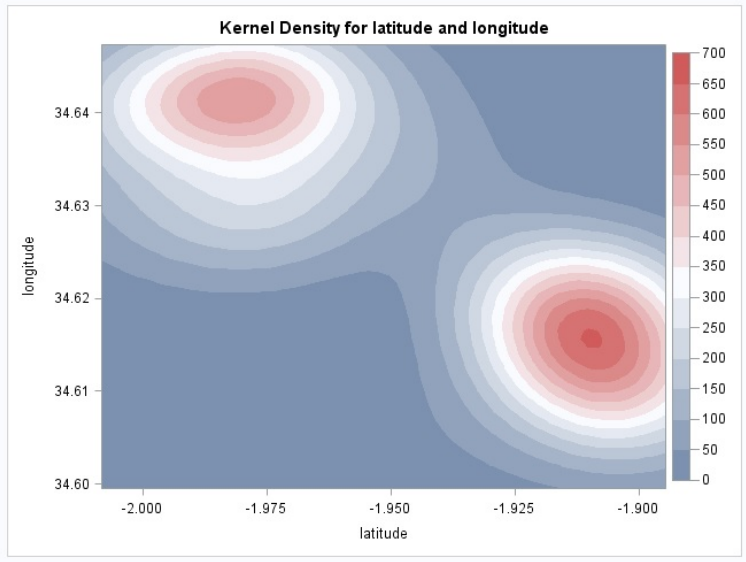
Further study on the improvement on methods for spatial data analysis are being done. Even though it was not covered in its entirety in this report, the selection of the bandwidth is integral to the process of estimating the kernel density estimate. Cronie and van Lieshout [5] have proposed a new method for bandwidth selection that is based on the Campbell formula applied to the reciprocal of the intensity function. The proposed method is unrestricted in that it does not require a specific class of point process models, it is non-parametric and does not require prior knowledge of the densities.

The SAS System

The KDE Procedure

Inputs	
Data Set	WORK.BOKORE
Number of Observations Used	269
Variable 1	latitude
Variable 2	longitude
Bandwidth Method	Simple Normal Reference

Controls		
	latitude	longitude
Grid Points	60	60
Lower Grid Limit	-2.008	34.6
Upper Grid Limit	-1.895	34.647
Bandwidth Multiplier	1	1



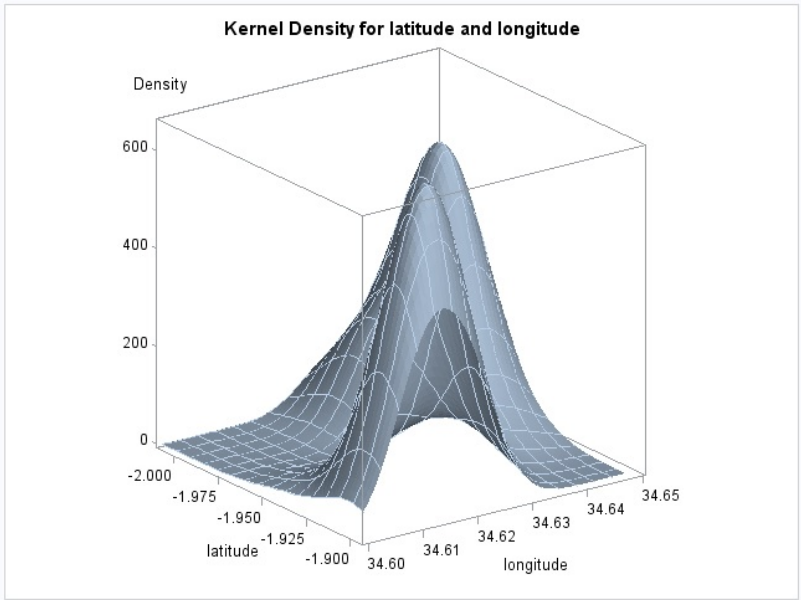


Figure 22: SAS output for the PROC KDE procedure using the Bokore village data set.

References

[1] Adrian Baddeley. Analysing spatial point patterns in `R`. Technical Report Version 4.1, CSIRO and Universtiy of Western Australia, 2010.

[2] Adrian Baddeley, Ege Rubak, and Rolf Turner. *Spatial Point Patterns: Methodology and Applications with* `R`. CRC Press, 2015.

[3] Adrian Baddeley and Rolf Turner. `spatstat`: an `R` package for analyzing spatial point patterns. *Journal of Statistical Software*, 12(6):1–42, 2005.

[4] Leo Breiman, William Meisel, and Edward Purcell. Variable kernel estimates of multivariate densities. *Technometrics*, 19(2):135–144, 1977.

[5] Ottmar Cronie and MNM van Lieshout. Bandwidth selection for kernel estimators of the spatial intensity function. *arXiv preprint arXiv:1611.10221*, 2016.

[6] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Cheong Schwarzkopf. Convex hulls. In *Computational Geometry*, pages 235–250. Springer, 2000.

[7] Peter J Diggle. *Statistical Analysis of Spatial and Spatio-temporal Point Patterns*. CRC Press, 2013.

[8] Peter J Diggle, Julian Besag, and J Timothy Gleaves. Statistical analysis of spatial point patterns by means of distance methods. *Biometrics*, pages 659–667, 1976.

[9] Tarn Duong. *Bandwidth Selectors for Multivariate Kernel Density Estimation*. PhD thesis, 2004.

[10] Tarn Duong and Martin Hazelton. Plug-in bandwidth matrices for bivariate kernel density estimation. *Journal of Nonparametric Statistics*, 15(1):17–30, 2003.

[11] Anthony C Gatrell, Trevor C Bailey, Peter J Diggle, and Barry S Rowlingson. Spatial point pattern analysis and its application in geographical epidemiology. *Transactions of the Institute of British Geographers*, 21(1):256–274, 1996.

[12] Brian D Ripley. *Spatial Statistics*, volume 575. John Wiley & Sons, 2005.

[13] Bernard W Silverman. *Density Estimation for Statistics and Data Analysis*, volume 26. CRC Press, 1986.

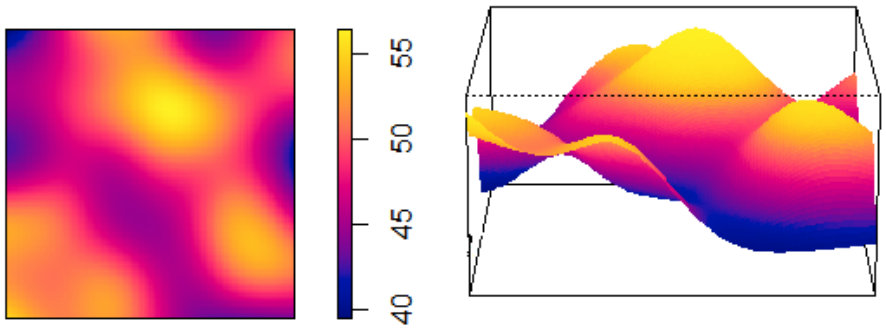
[14] Donald F Specht. Probabilistic neural networks. *Neural networks*, 3(1):109–118, 1990.

[15] Graham Upton and Bernard Fingleton. *Spatial Data Analysis by Example. Volume 1: Point Pattern and Quantitative Data.* John Wiley & Sons Ltd., 1985.

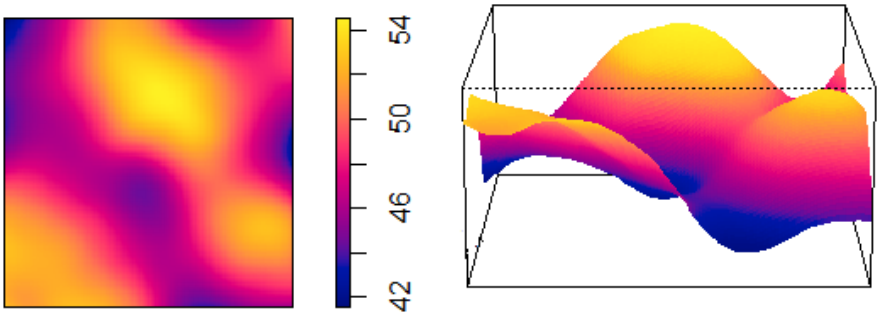
[16] Thorsten Wiegand and Kirk A Moloney. Rings, circles, and null-models for point pattern analysis in ecology. *Oikos*, 104(2):209–229, 2004.

[17] Zhixiao Xie and Jun Yan. Kernel density estimation of traffic accidents in a network space. *Computers, Environment and Urban Systems*, 32(5):396–406, 2008.

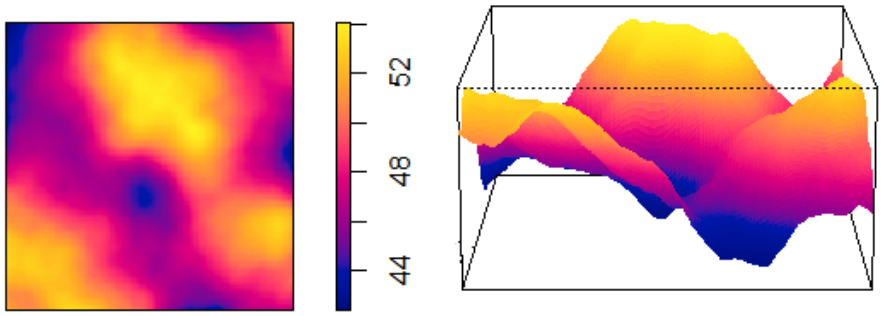
Appendix



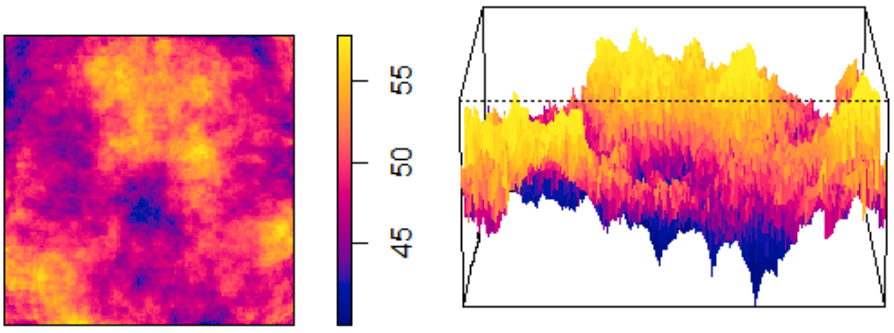
(a) `kernel` = "`gaussian`"



(b) `kernel` = "`quartic`"

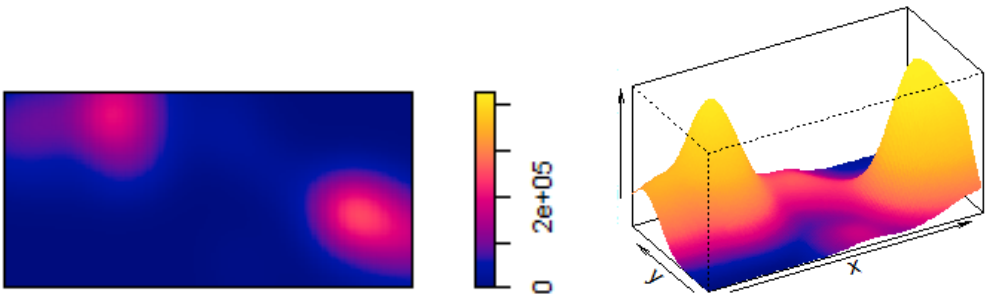


(c) `kernel` = "`epanechnikov`"

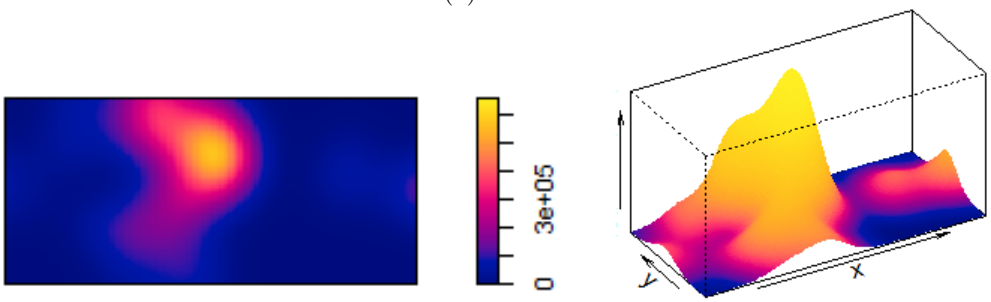


(d) `kernel` = "`disc`"

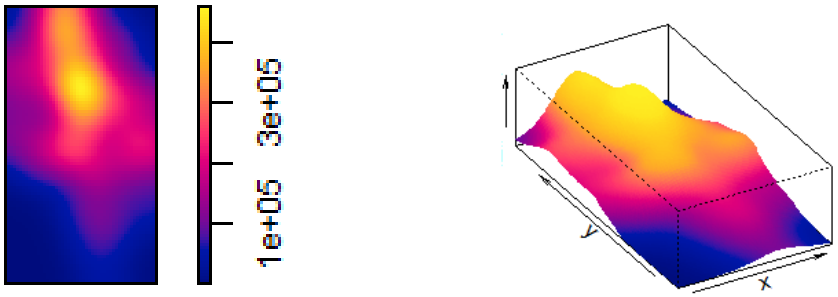
Figure 23: Kernel smoothed estimates for the `density.ppp` function of simulated data in `R` with variable options for `kernel`.



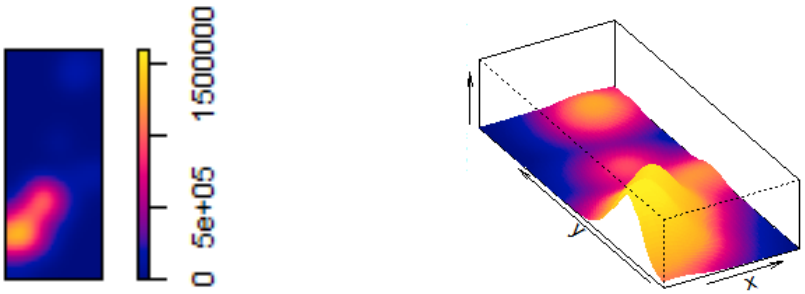
(a) Bokore



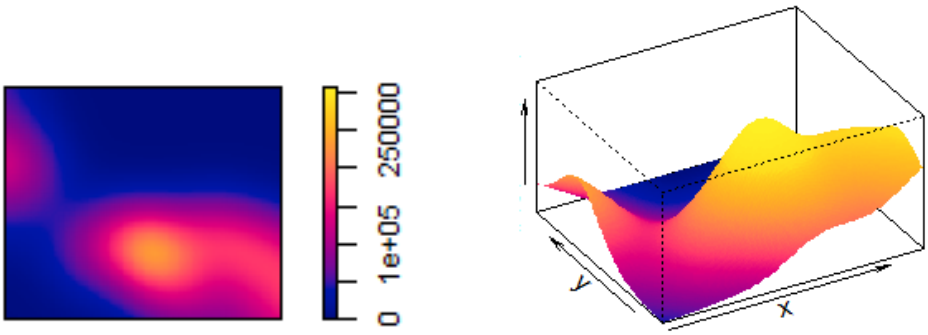
(b) Iseresere



(c) Itununu



(d) Kemgesi



(e) Gantamome

Figure 24: Kernel smoothed intensity and perspective plots of a point pattern of the geographic locations of households for five villages in Mara province, Northern Tanzania.

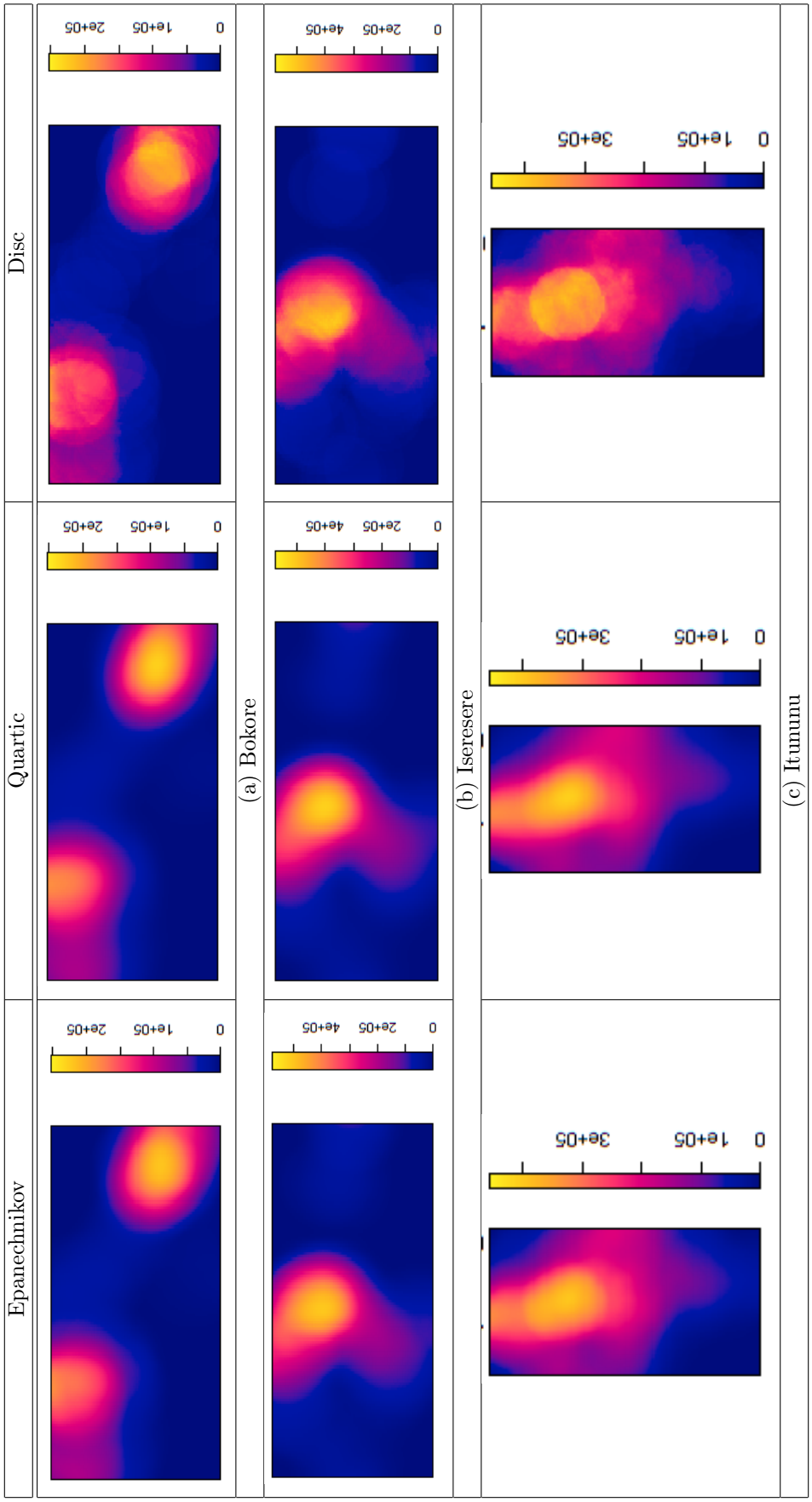


Figure 25: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for five villages in Mara province, Northern Tanzania for variable kernel options

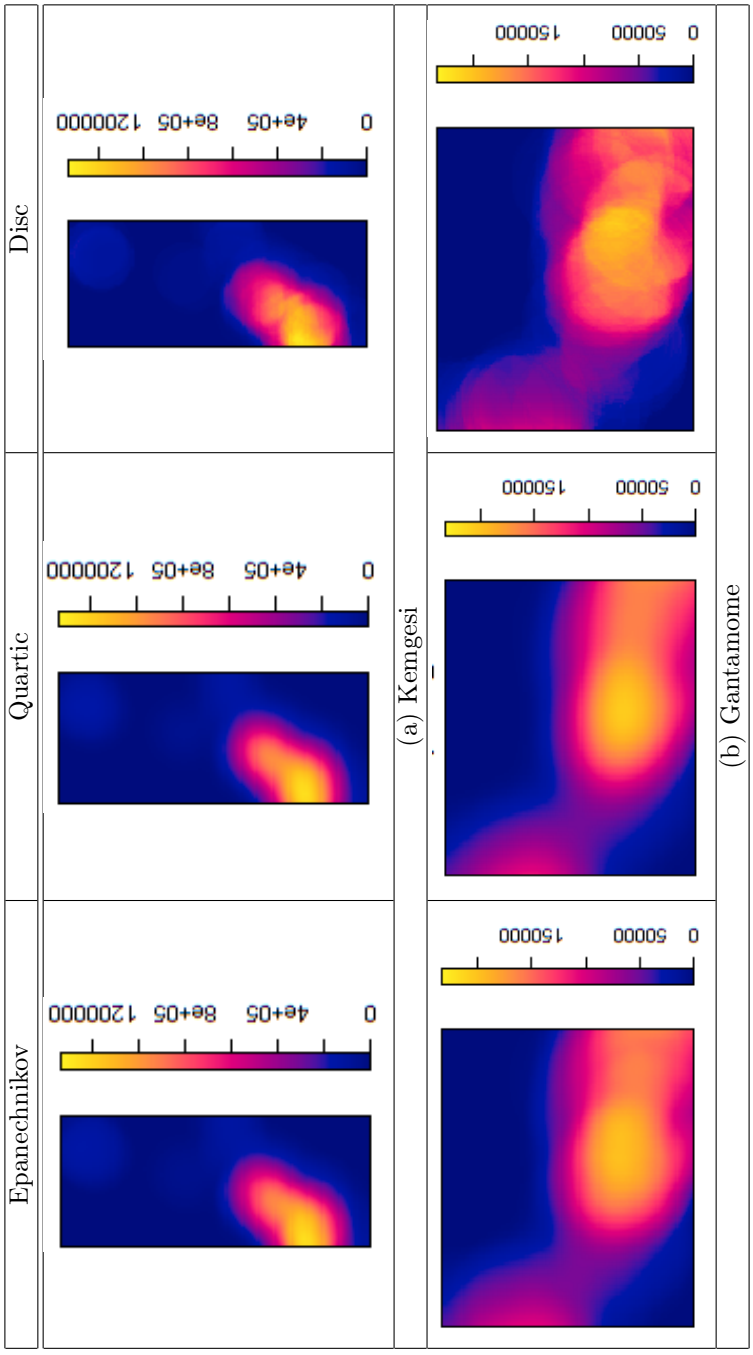


Figure 26: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for five villages in Mara province, Northern Tanzania for variable kernel options

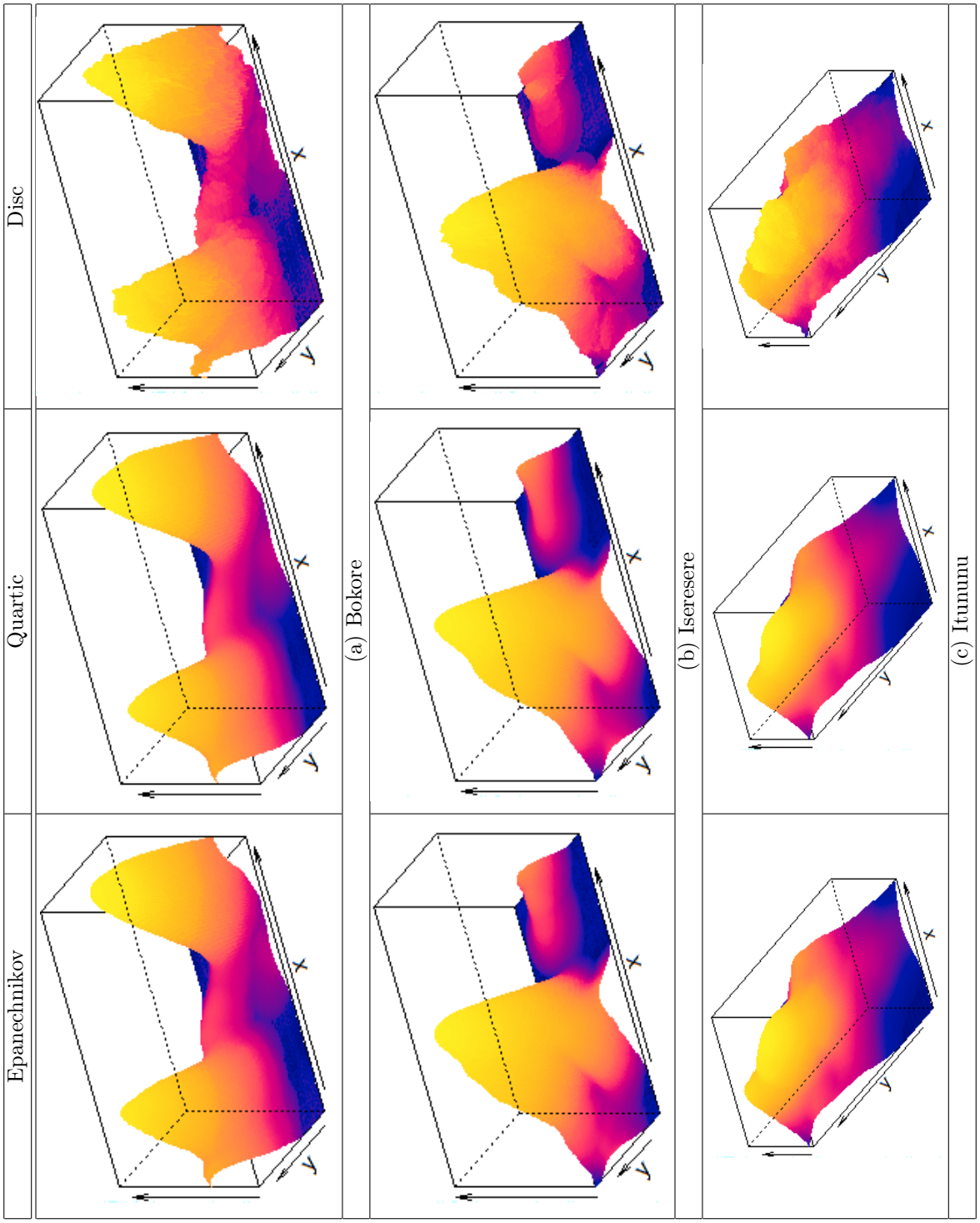


Figure 27: Perspective plot of kernel smoothed intensity estimates of a point pattern of the geographic locations of households for five villages in Mara province, Northern Tanzania for variable kernel options

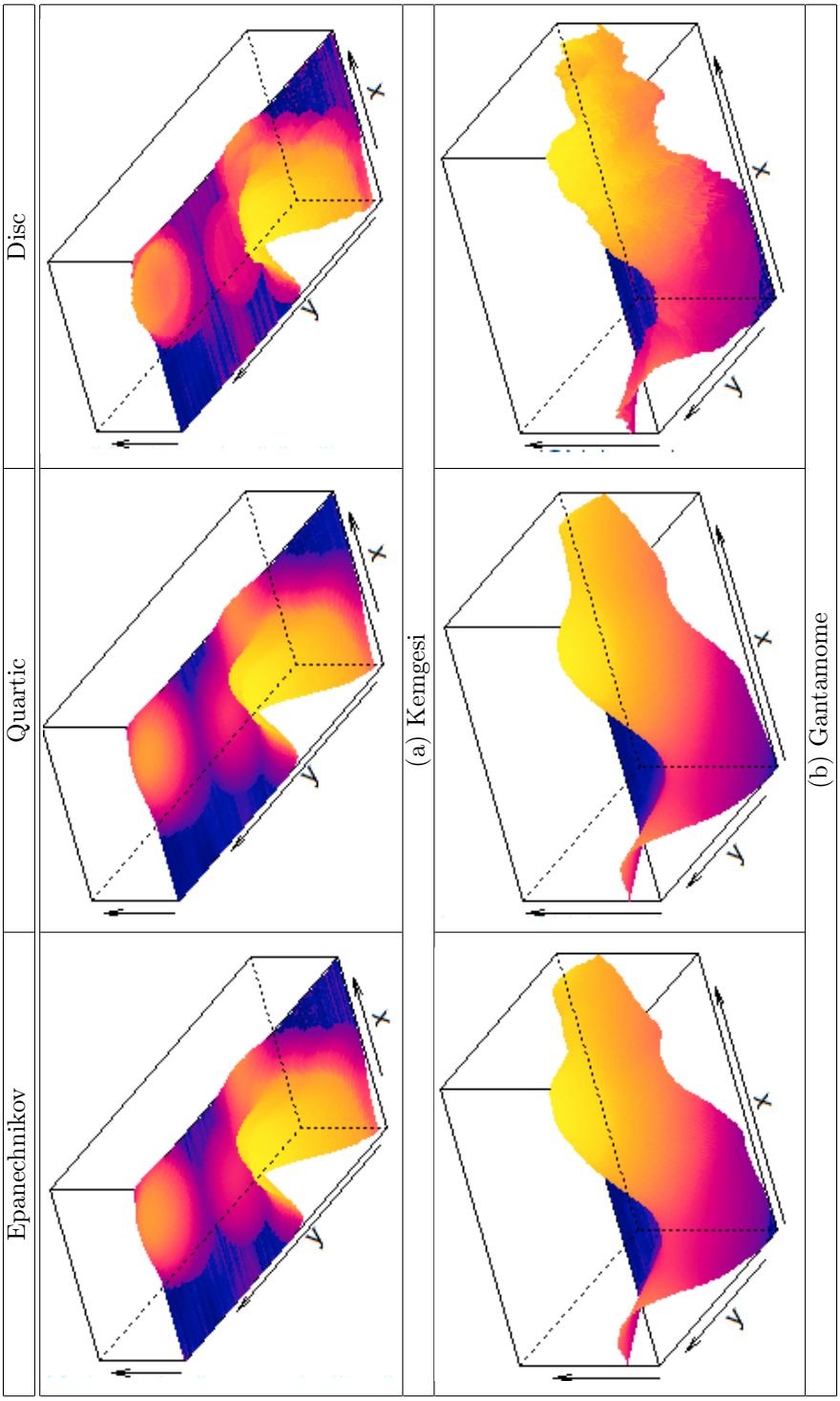


Figure 28: Perspective plot of kernel smoothed intensity estimates of a point pattern of the geographic locations of households for five villages in Mara province, Northern Tanzania for variable kernel options

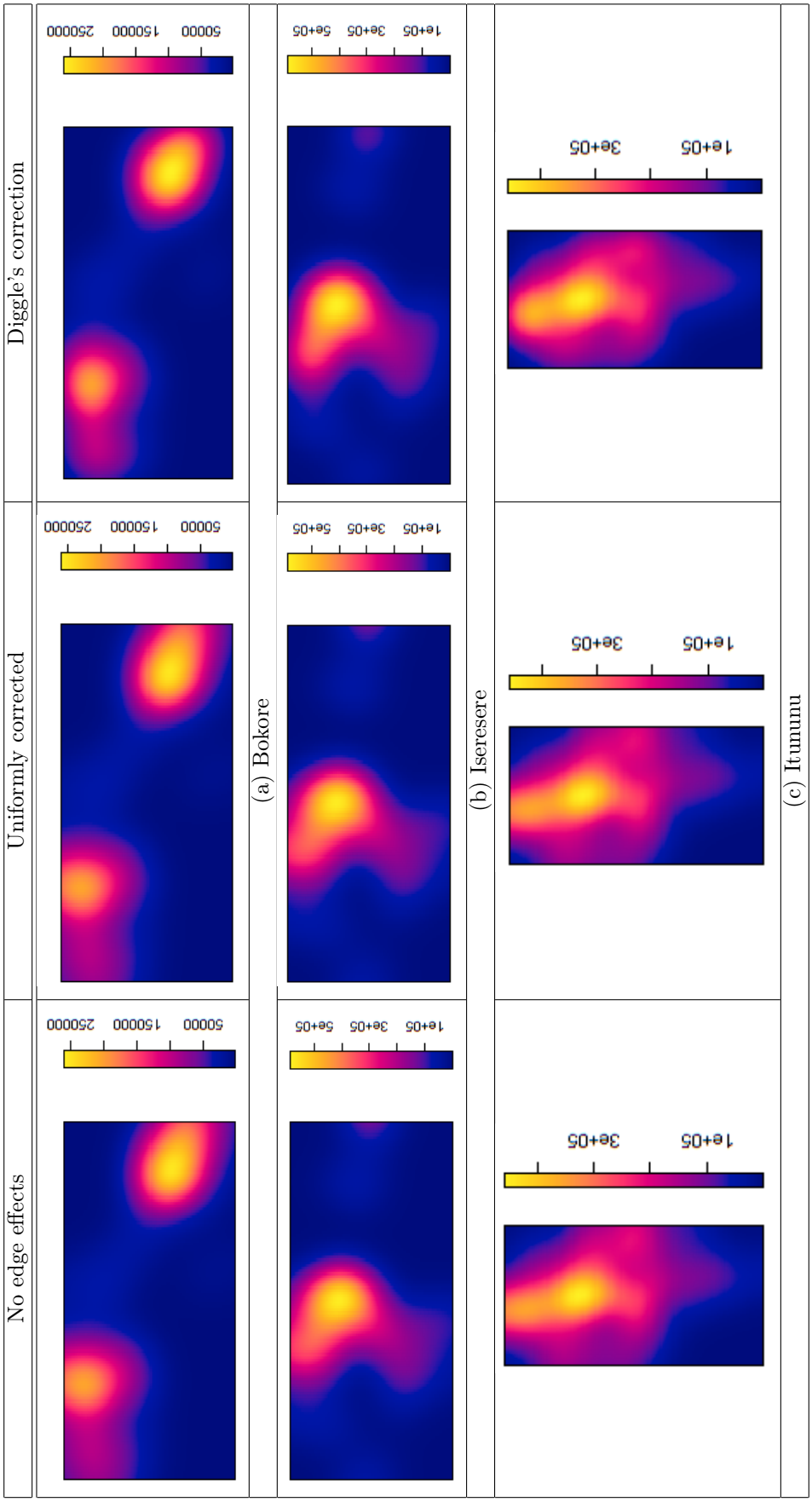


Figure 29: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for five villages in Mara province, Northern Tanzania for different edge corrections

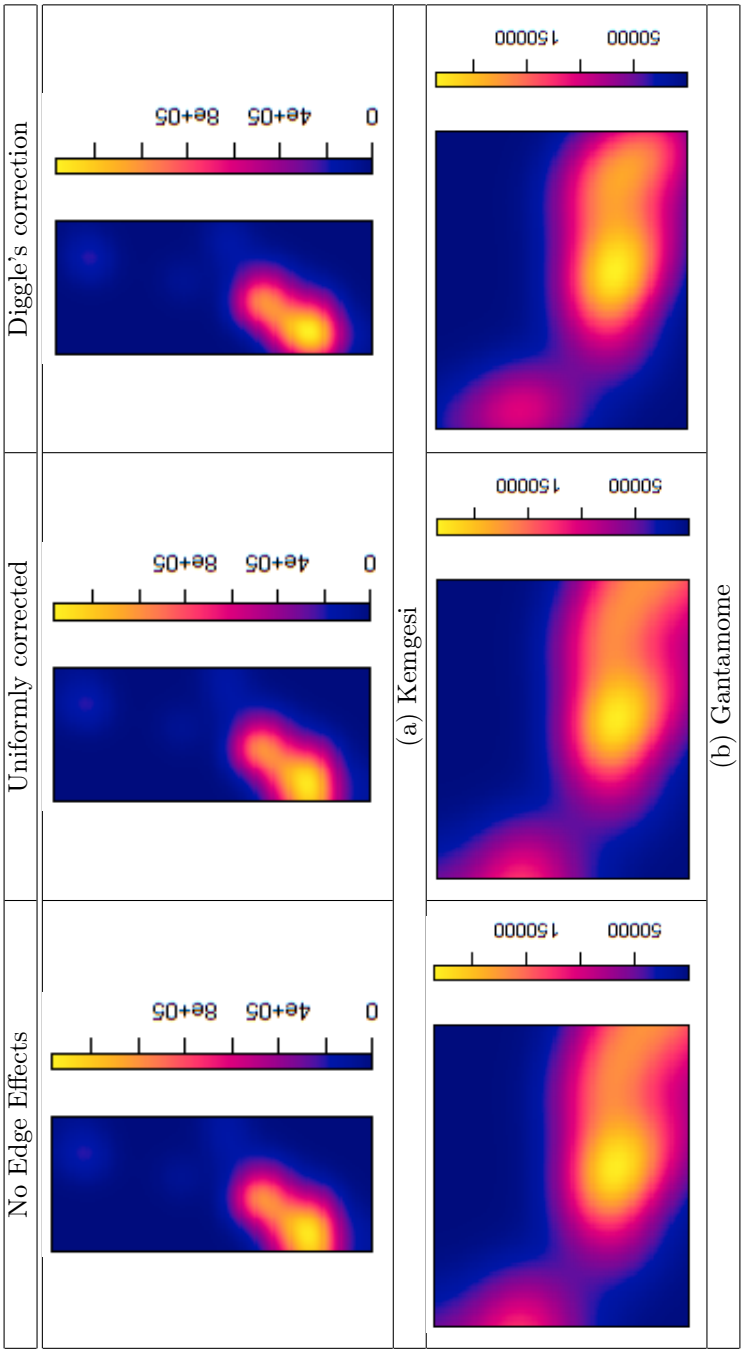


Figure 30: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for five villages in Mara province, Northern Tanzania for different edge corrections

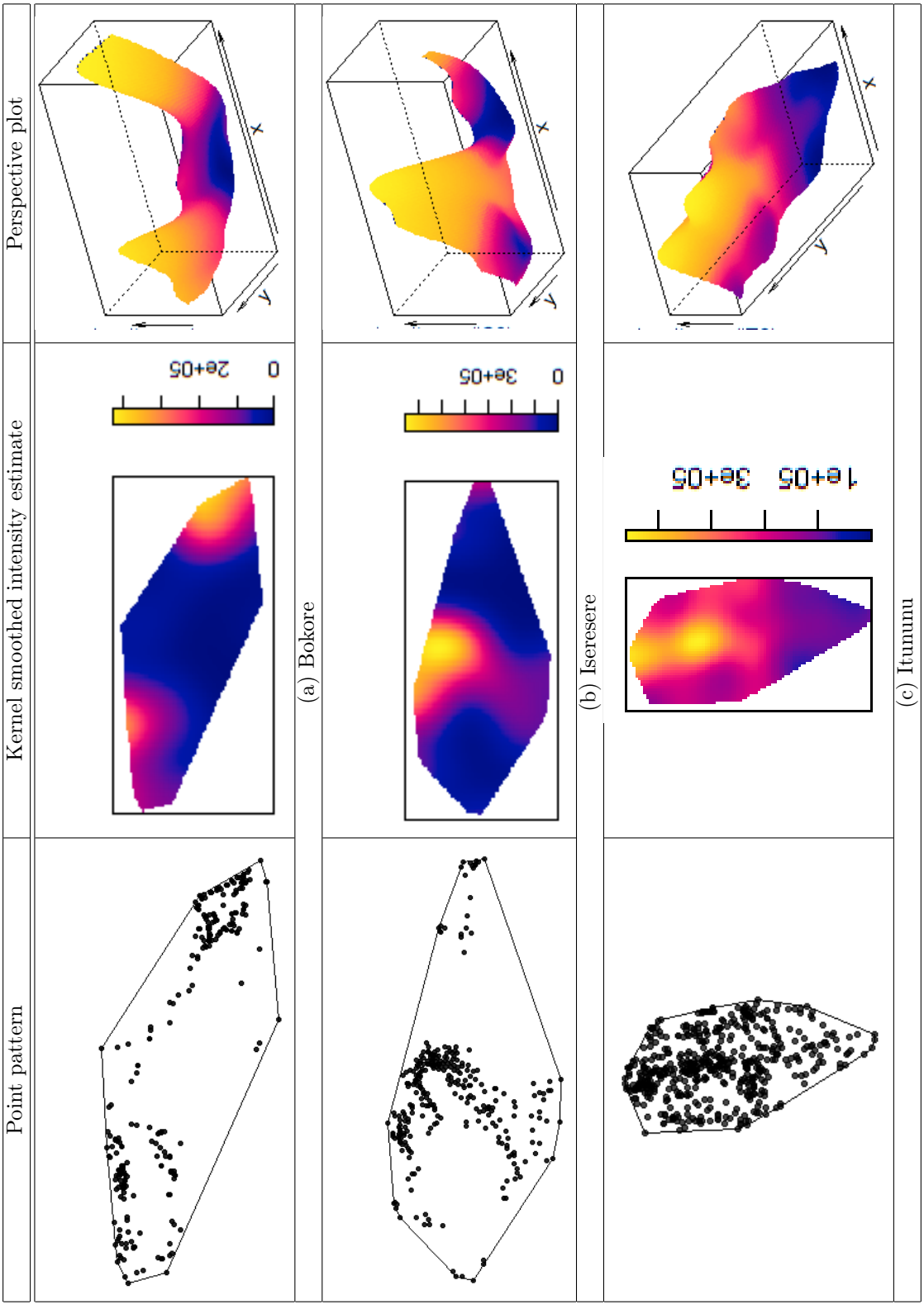


Figure 31: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for villages in Mara province, Northern Tanzania with convex window for the Gaussian kernel

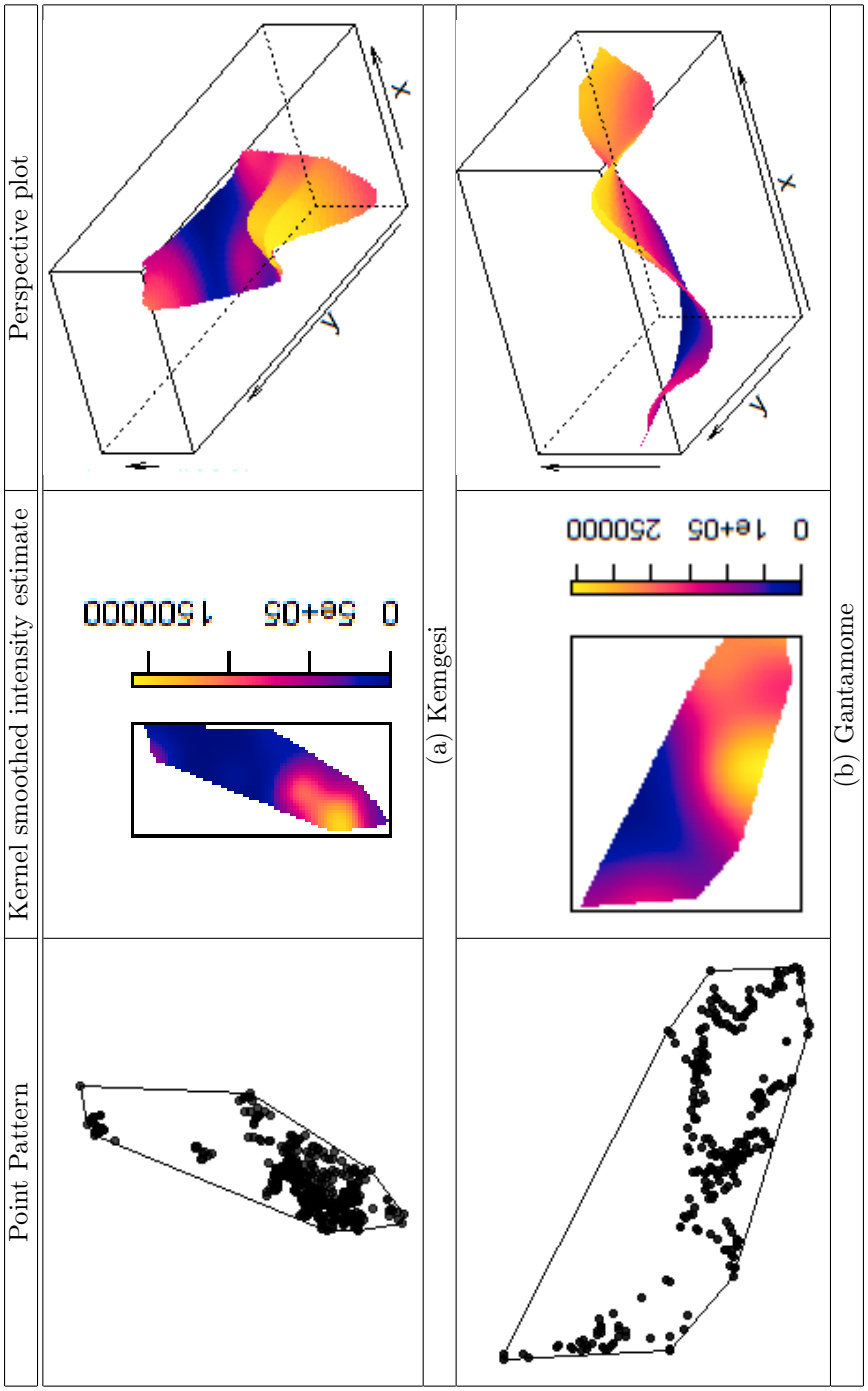


Figure 32: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for villages in Mara province, Northern Tanzania with convex window and Gaussian kernel

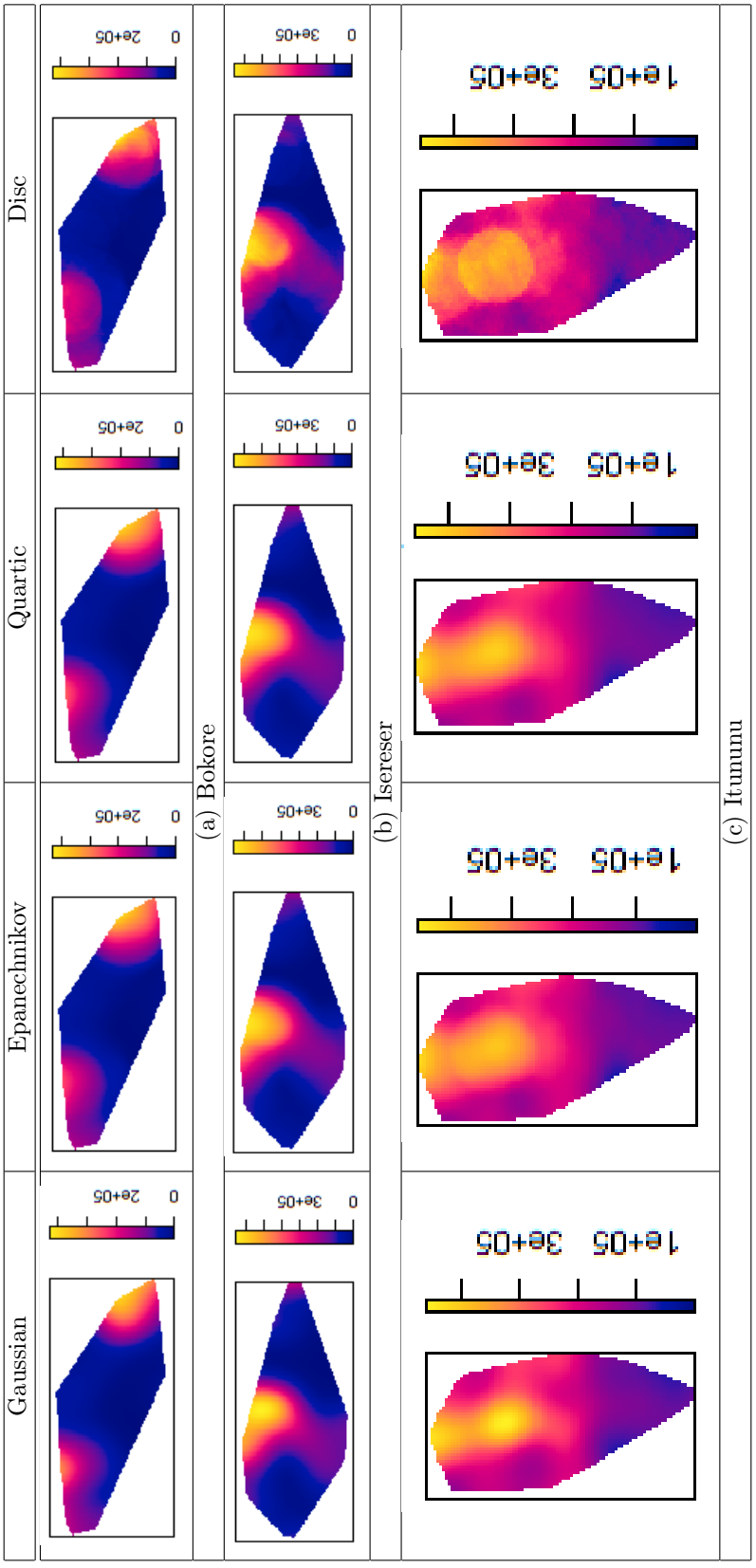


Figure 33: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for five villages in Mara province, Northern Tanzania for variable kernel options for a convex hull window

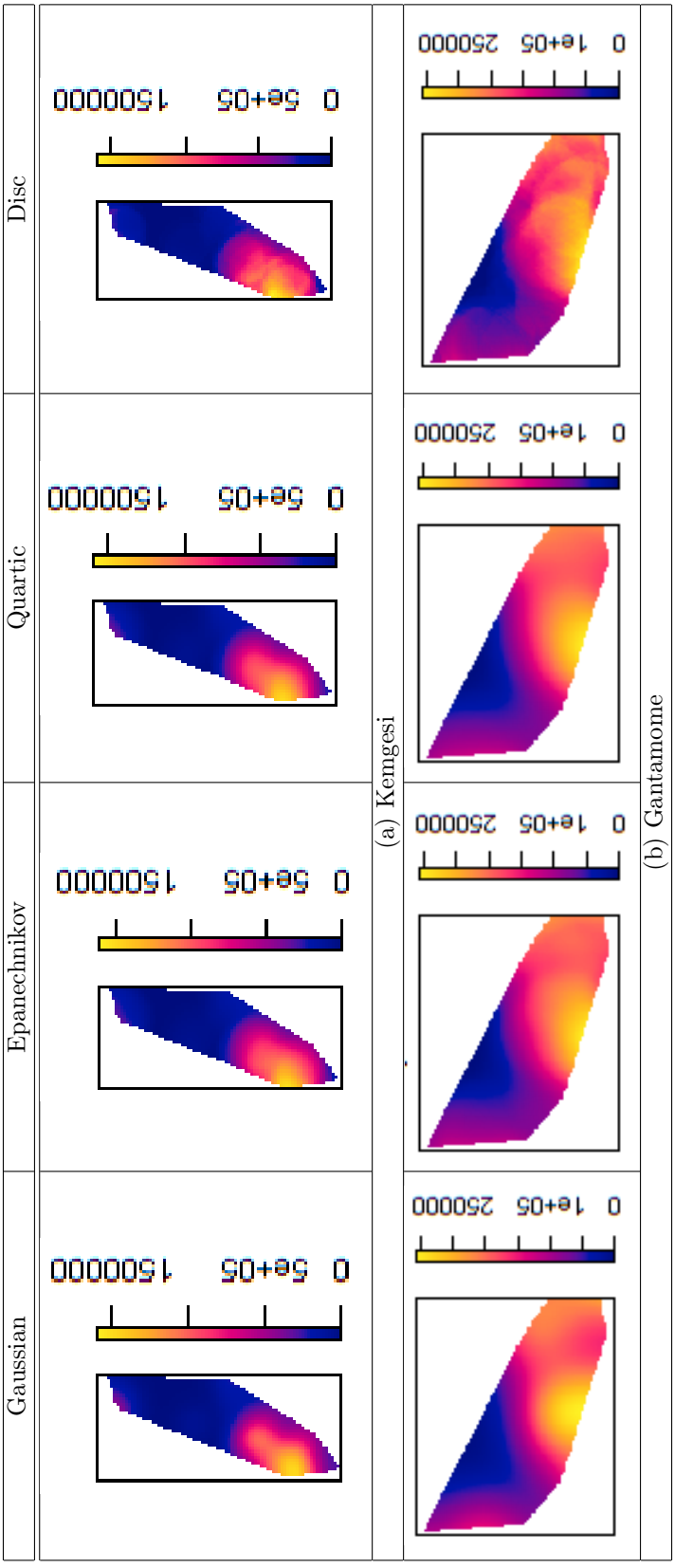


Figure 34: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for five villages in Mara province, Northern Tanzania for variable kernel options for a convex hull window

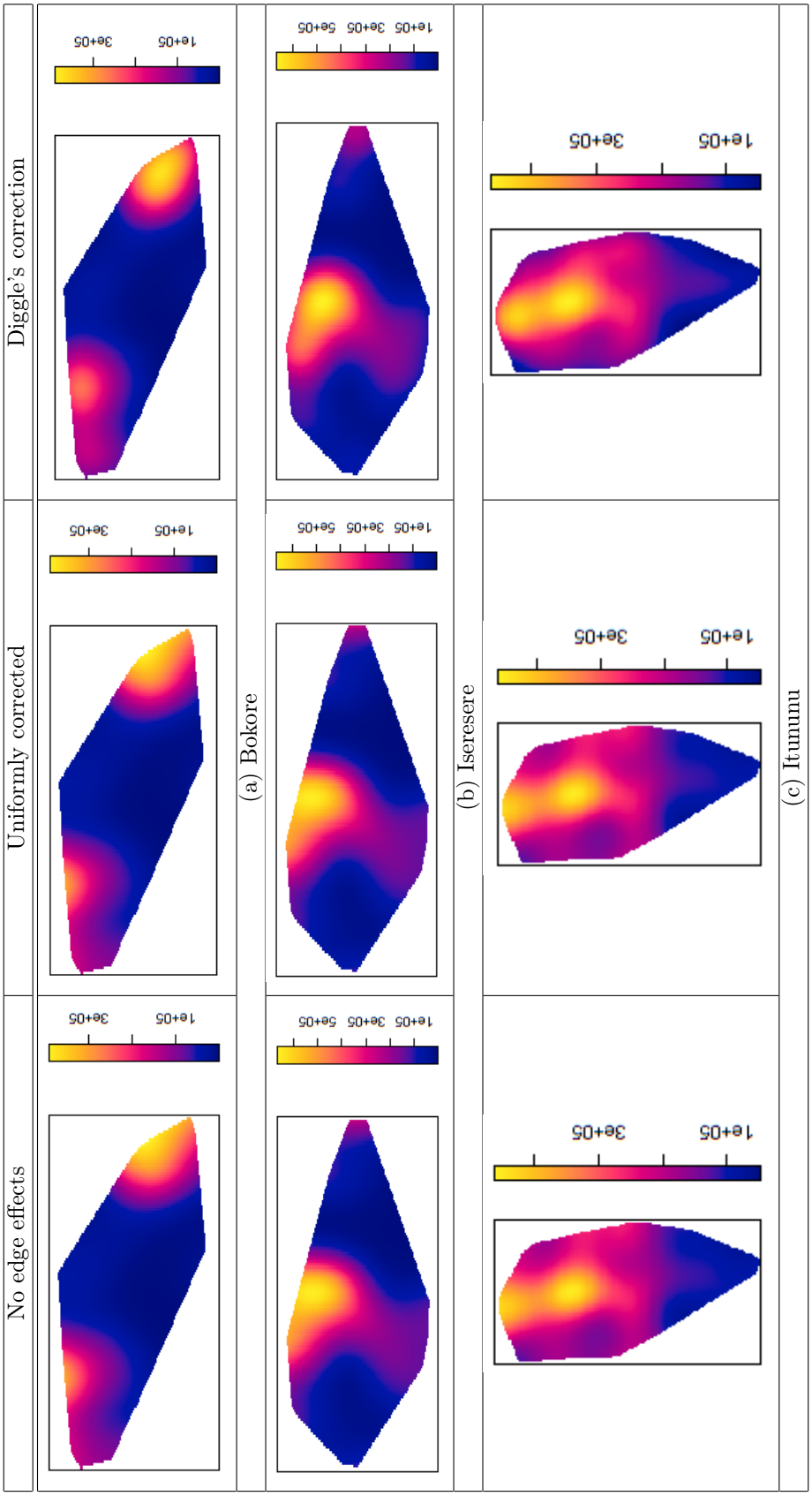


Figure 35: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for five villages in Mara province, Northern Tanzania for different edge corrections for a convex hull window

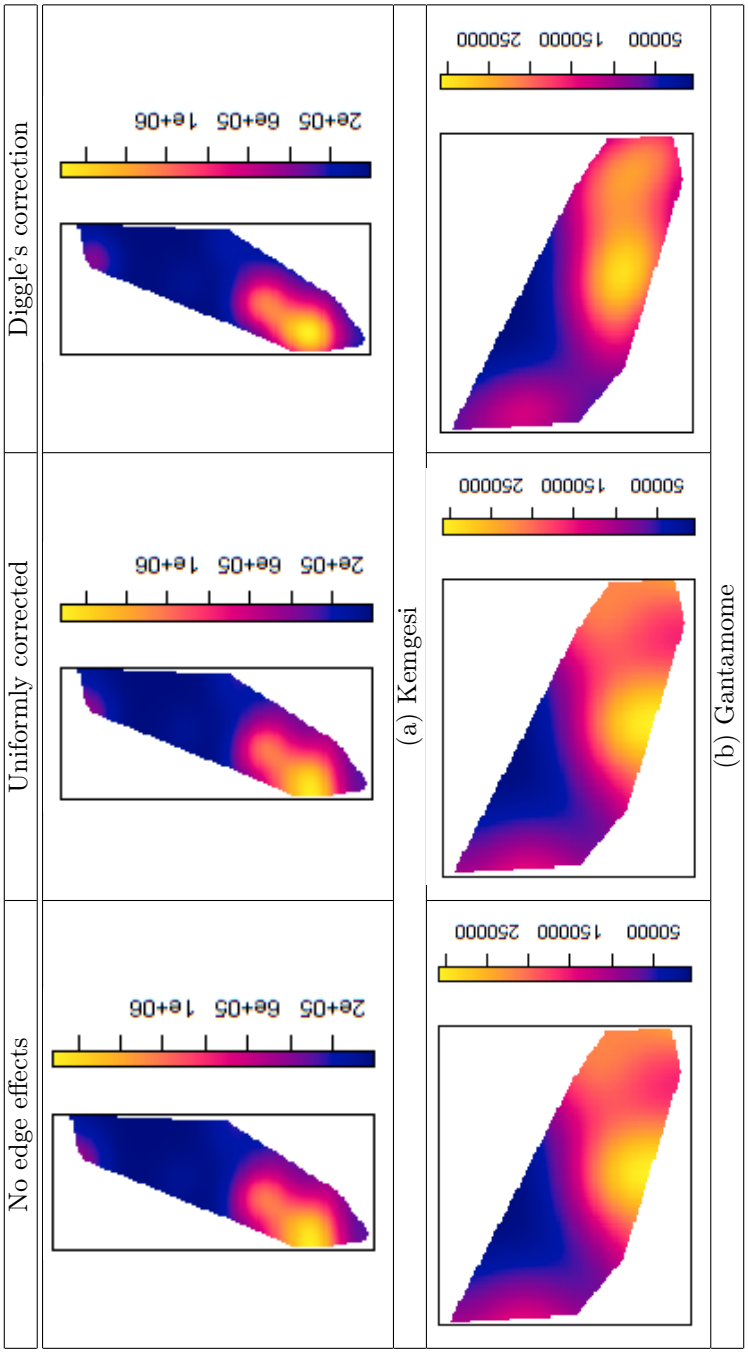


Figure 36: Kernel smoothed intensity estimates of a point pattern of the geographic locations of households for five villages in Mara province, Northern Tanzania for different edge corrections for a convex hull window

Panel data regression models: fixed effects approach

Themba Masilela 13026012

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr J Kleyn

Department of Statistics, University of Pretoria

October 30, 2017

Abstract

This paper will be discussing the estimation of the panel data regression models using the fixed effects (FE) approach. Under the approach, the intercept is treated as a constant across both space and time. The regression models are developed on the bases of the assumptions made with regard to the intercept, the slope coefficients and the error term. Thus the discussion of the four models formed under the following assumptions: In the first model all the coefficients remain constant across space and time. The second model assumes that the intercept varies across space but remains unchanged over time and the slope coefficients remain unchanged across both time and individuals. This model is also known as the Least squares dummy variable (LSDV) regression model. In the third model, the slope coefficients remain constant over space and time while the intercept varies across space and time. In the last model discussed, all coefficients vary across space while staying constant through time. In each of the models it is assumed that the error term captures differences over space and time where it follows a normal distribution with zero mean and a constant variance. The last three models assume the "individuality" of coefficients across space and time through the use of dummy variables, this is called the least squares dummy variable (LSDV) method. Hence the dummy variable models discussed are used to detect individuality or rather differences among cross-sectional units. A practical example is considered for the illustration of the LSDV method which uses ordinary least squares (OLS) estimation method.

Declaration

I, *Themba Masilela,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____

Themba Masilela

_____

Judy Kleyn

_____

Date

Acknowledgements

First and foremost I give thanks to the Lord Jesus Christ, who gave his life for me to live: My enabler, strength and eternal supply. It is with much gratitude that I acknowledge the Nedbank trust for the more than generous support over the past five years. The bursary initiative is a blessing and without it, my journey would have been much different for the worst. I am grateful to the trust for investing in my dream and giving me the chance to enable myself to live up to my potential. I could never forget all the expertise and effort made by the prestigious University Of Pretoria academic staff to ensure excellence in the transfer of quality knowledge and skill. I am eternally grateful for the guidance, effort and patience extended towards me by my supervisor Dr Judy Kleyn and not to mention the unprecedented advice from Dr Inger Fabris-Rotelli. I am thankful to my family for the support and unwavering belief in me, for fueling my desire for achievement. Last but not least thanks to my dearest friend, Tshidiso Thebe for having a heart of gold and encouraging me with much positivity in the face of seemingly impossible milestones; those encouraging words we shared surely go a long way.

Contents

List of Figures

List of Tables

1 Introduction

Cheng [3], defines a panel or longitudinal data set as having a typical structure of N time-series data observations obtained from the same individuals over a period of time. This is different from pooled cross-sectional data, discussed in [12], where cross-sectional data on N random objects within a population are observed with no restriction of observing the same set of objects over the period of time. Pooled cross-sectional data sets are mostly as a result of censoring due to security purposes; individuals may not want to be identified when completing surveys. Thus the nature of panel data has the advantage of allowing for efficient study of characteristic traits (heterogeneity) of specific objects within a population through multiple data observations on those same objects. [1] mentions the popularity of Panel data in economic research, say for example conducting a study of profitability among companies operating in the same sector or across different sectors. An assumption, that could be tested, in the former case would be assuming homoscedasticity and the latter heteroscedasticity as it is more appropriate to assume that within a sector the variation remains constant and different sectors will exhibit different variation. Panel data is also useful in medical research, for example: in a study of gene defects a selected group of affected individuals may be observed through out time and the effects of the defect on the individuals may be studied. Panel data is rich in fields of application, it is commonly used in economic research.

A pure time series data set is an observation of a single object in a population of interest (say an industry) throughout a period of time. Conversely a cross-section data set observes many objects in the population at a single point in time. [5] mentions the inherent advantages of panel data over pure cross-sectional data sets and pure time series data sets. Panel data sets allow the capturing of the omitted variables called unobserved heterogeneity which may pass unobserved in pure time series or pure cross-sectional data analysis. [5] argues that this is the case since panel data observes the same objects over time, hence the individuality of each object (heterogeneity) may be captured through allowing for object specific variables. [9] and [10] mention that the consequence of failure to observe heterogeneity by pure cross-section and pure time series data leads to the risk of obtaining biased estimators for the parameters. [5] further emphasises the difficulties encountered in estimating unbiased estimators of the parameters from pure cross sectional analysis as well as estimating the parameters from pure time series analysis through a discussion. Biased estimators may result in erroneous predictions which might have economic consequences.

Baltagi [2], further remarks that, due to the combination of both time-series and cross-sectional data, panel data tends to provide "*more informative data, more variability, less collinearity among the vari-*

ables, more degrees of freedom and more efficiency". It is further seen that most of these are desired qualities for elementary analysis of regression models according to [11]. It is worth noting that the object specific variables used in panel data come at a cost. This is because for every object specific variable added, a corresponding parameter is included in the model. This has a consequence of decreased degrees of freedom, since one loses a degree of freedom for every parameter added, this has the implication of having less data available for meaningful statistical analysis this is more detrimental in panels having few observations for the cross-sectional units.

Gujarati [5], observes the fact that due to panel data following the same cross sectional units through time, panel data is better suited for the study of *dynamics of change*. Say that one is interested in investigating the spells of unemployment. By having observations of the same objects both before and after unemployment one is able to detect the effect that unemployment had on the affected objects. The same principle may be used to study level of skills, job turnover and labour mobility.

More complicated behavioural models such as economies of scale and technological advancement may be better handled through panel data rather than pure cross-sectional or time series data. Moreover [5] mentions that panel data can handle the bias that may emerge as a consequence of grouping of the data into broad aggregates.

Some of the well known panel data sets used in economic research are the Panel study of income dynamics (PSID) where, each year, 5000 families are observed and data is collected under the supervision of the Institute of Social Research at the University of Michigan. The data collected is on various socioeconomic issues as well as demographic variables. Another panel data set called the Survey of Income and Program Participation (SIPP) is one conducted by the Bureau of the Census Department of Commerce. Four times a year data is gathered from respondents about their economic contribution. Other panel data is the German Socio-Economic Panel (GESOEP) and the National Longitudinal Survey of Youth (NLSY). Several other such surveys are conducted by government and non-government agencies in a number of countries.

As is the case that some participants in the surveys may default from the surveys through death and/or other reasons, [7] mentions that panel data sets can be classified according to the number of observations, on each cross-sectional unit, as either balanced or unbalanced. A panel data set is defined as balanced if all N cross-sectional units have T observations, i.e. every object has an observation at

each point $t = 1, \ldots, T$ of sampling. When the cross-sectional units have varying observations, say T_i where $i = 1, \ldots, N$, the panel data set is termed unbalanced. Unbalanced panel data sets may arise as a result of "missing" data also known as censored data which may be due to the unavailability of some cross-sectional units during the time of sampling or the discontinuation of observations made on some cross-sectional units in the particular study. This paper will specifically study a balanced panel data set.

The very first panel data conference was held in the year 1977 in Paris at a seminal conference hosted at INSEE. the organization of the conference is attributed to Pascal Mazodier, Jacques Mairesse and Alain Trognon. The conference led to the publication of two volumes of *Annales de l'INSEE* edited by Trognon and Mazodier (1978). The increased use of panel data in research studies has since evolved with innovative solutions developed to accommodate non-traditional approaches. However this paper will discuss, in particular, the linear fixed effects model. Other topics include non-linear fixed effects models, discreet data fixed effects models, truncated and censored fixed effects models as well as incomplete fixed panel data models as outlined in [3].

Gujarati [5], introduces the basic panel data regression model with the following equation:

$$Y_{it} = \beta_0 + \beta_1 X_{1it} + \beta_2 X_{2it} + \epsilon_{it}$$

where $i = 1, 2, \ldots, N$ are representative of N cross-sectional units observed over $t = 1, 2, \ldots T$ time periods and β_0 is the intercept, with β_1 and β_2 the slope coefficients of the object specific variables or rather the exogenous random variables X_{1it} and X_{2it}. The error term ϵ_{it} captures differences over space and time while Y_{it} is the response variable. It is worth noting that the number of exogenous variables need not be two, the variables may be as small as one or as large as 20 and more. It all depends on the researchers discretion on the basis of necessity of number of input variables deemed significant and most of all independent. [5] warns that the researcher must take precaution of the number of variables used in the model; too many variables may lead to multicollinearity among variables. For instance, the object specific variables may overlap. For example suppose that among many variables in a model, education, skill level and poverty are included. since all 3 variables are related because people in poverty usually have little or no education not to mention the quality of the education they have access to which directly affects their skills level. Therefore precise estimation for each distinct variable may be difficult in this case. Too many variables may also result in low degrees of freedom especially in the case of a small number of observations for each cross-sectional unit.

Arrelano [1] observes that the interest in panel data has two core motives. The first motive pertains to the desire to exploit panel data with the aim to control the unobserved time-invariant heterogeneity in each cross-sectional unit which may be undetected by pure time series data. The second motive is attributed to using panel data as a means to disentangle components of variance and estimate the transition probabilities among states with the aim to study the dynamics of cross-sectional units. [1] further mentions that the two motives can be associated with two strands of panel data analysis namely: fixed effects and random effects, the two respective approaches are useful when the researcher wishes to estimate the panel data regression model. The theory and practical sections of the paper will mostly cover the fixed effects approach to panel data regression analysis.

Gujarati [5], distinguishes between the two approaches on the basis of the assumptions made about the intercepts. In the random effects approach, the intercepts are given by $\beta_i = \beta_0 + \mu_i$ for $i = 1, 2, \ldots N$ where error term μ_i captures differences over the individual cross-sectional units. On the other hand, the intercept is given by β_0 which is non-stochastic over all cross-sectional units, this is known as the fixed effects approach. The focus of this paper will be on the fixed effects approach. [5] suggests that the approach to use relies upon the assumption made with regards to correlation between the exogenous variables and the error term. The fixed effects approach is chosen if there is an assumption of correlation between the X's and the ϵ. Alternatively the random effects approach is chosen if there is an assumption of no correlation.

Gujarati [5], examines three estimation methods which may be used under the fixed effects method. Suppose that the basic model is given by:

$$Y_{it} = \beta_i + \beta_1 X_{1it} + \beta_2 X_{2it} + \epsilon_{it}.$$

Since it is assumed that the error term and the exogenous variables are correlated under this method, i.e $cov(X_{it}, \epsilon_{it}) \neq 0$, OLS estimation may not be used. A simple solution to the predicament, as collaborated by [8], is the *first difference estimator* where the coefficient slope parameters are estimated through the resultant regression model:

$$\triangle Y_i = \beta_1 \triangle X_{1i} + \beta_2 \triangle X_{2i} + \nu_i$$

where $\triangle Y_i = Y_{it} - Y_{i(t-1)}$, $\triangle X_{1i} = X_{1it} - X_{1i(t-1)}$, $\triangle X_{2i} = X_{2it} - X_{2i(t-1)}$ and $\nu_i = \epsilon_{it} - \epsilon_{i(t-1)}$ i.e. [5] derives this model through taking the difference of two successive basic models on the same cross-sectional unit taking into account successive time periods. Note that in the basic model, the β_i values are not

observed thus the first difference method enables unbiased estimation of the coefficient slope parameters by getting rid of the unobservable effect. [5] argues that OLS estimation may then be employed since even though the X's and the error terms are correlated, there is no priori reason that their differences are also correlated. Another alternative identified in [5] for dealing with the unobserved effect β_i is called the within group estimator. The regression model under this method is given by:

$$Y_{it} - \overline{Y}_i = \beta_1(X_{1it} - \overline{X}_{1i}) + \beta_2(X_{2it} - \overline{X}_{2i}) + \epsilon_i.$$

The model is extensively illustrated in [6] with collaboration done in [10] where the equation follows by considering the distances of each observation from its group mean. Similarly OLS estimation is used because there is no reason why the distances of each observation from its mean should be correlated. And finally, [5] introduces the least squares dummy variable (LSDV) estimator which will be the estimation technique that will be discussed in this paper.

Gujarati [5], introduces Some of the models that result as a consequence of the assumptions of homogeneity or heterogeneity of the parameters to be estimated in the model. Although the assumptions considered in this paper yield four models to be studied. Suppose that it is assumed that all the coefficients are constant, then the model is given by:

$$Y_j = \beta_1 + \beta_2 X_{2j} + \beta_3 X_{3j} + \epsilon_j \quad \text{for } j = 1, 2, \ldots, 80 \tag{1}$$

This is known as the pooled regression model and is defined as **model 1**. Suppose that it is assumed that the slope coefficient is different across all individuals but the slope coefficients remain constant then **model 2** follows as:

$$Y_{it} = \beta_{1i} + \beta_2 X_{2it} + \beta_3 X_{3it} + \epsilon_{it} \quad \text{for } i = 1, 2, 3, 4 \text{ and } t = 1, 2, \ldots, 20. \tag{2}$$

This model is known as the least square dummy variable model. It also follows that if it is assumed that the slope coefficient changes over both time and the individual cross-sectional units but the slope coefficients remain constant then **model 3** is given by:

$$Y_{it} = \beta_{1it} + \beta_2 X_{2it} + \beta_3 X_{3it} + \epsilon_{it} \quad \text{for } i = 1, 2, 3, 4 \text{ and } t = 1, 2, \ldots, 20. \tag{3}$$

Finally suppose that the assumption is that all coefficients vary across individuals then **model 4** follows

as:

$$Y_{it} = \beta_{1i} + \beta_{2i}X_{2it} + \beta_{3i}X_{3it} + \epsilon_{it} \quad \text{for } i = 1, 2, 3, 4 \text{ and } t = 1, 2, \ldots, 20. \tag{4}$$

The theoretical and illustrative use of the LSDV method on the last 3 models mentioned above will be considered in sections 2 and 3 respectively and finally section 4 contains the conclusion, the limitations of the technique and possible solutions which can be considered to rectify the problems detected in this modeling approach.

Moreover, the theory will have a specific focus on the testing procedures with respect to the assumptions of heterogeneity and/or homogeneity in the intercept coefficients, i.e. aim to show how the differential intercept coefficients are used to detect differences between different cross sections as well as showing how the differential slope coefficients are used to detect the differences between different slopes. As such, the significance of the coefficients used in the models will also be considered in the applications section. The estimation equations are then produced based on the statistical inferences made on the coefficients.

Furthermore, [5] remarks that it is worth noting that in using the LSDV model, caution must be taken when including too many dummy variables (as is done in model 4). The introduction of many dummy variables, as in the case of subject specific variables, could also lead to multicollinearity being present among the dummy variables, the consequence of multicollinearity is that precise estimation of one or more parameters may be difficult due to the overlap of the explanatory variables, thus leading to increased estimator variance. [5] further mentions that the implication is wider confidence intervals which may lead to more readily failing to reject the hypothesis of insignificance.

Given that multicollinearity is to be considered, the natural question that follows is how can one test for multicollinearity? [5] introduces the **variance-inflating factor (VIF)**, which is defined as a measure that shows how the variance of an estimator is inflated by the presence of multicollinearity. [4] further collaborates that in the cases of high collinearity, one may find that one or more of the partial coefficients are individually statistically insignificant on the basis of the t-test. [5] further mentions that in such cases, the R^2 value which is the overall measure of goodnes fit is very high. Hence one may reject the null hypothesis of insignificance under the F-test. Since one may assume that the model with a high coefficient of determination is a better fit for data being analysed. More-over [5] mentions the argument by Kmenta that multicollinearity is a question of degree and not of kind, as such one can measure its degree in any particular sample. VIF is the measure that will be used in this paper.

What is of interest is that as the R^2 value increases towards unity i.e. as the collinearity of the regressors increases, the *VIF* increases as well. A rule of thumb to be used is that if the *VIF* of a variable exceeds 10, which is the case if R^2 is in excess of 90%, then the conclusion is that the variable in question is highly collinear and thus its parameter estimate may be questionable due to the large variance of the OLS estimator. The following is a graphical explanation of multicollinearity illustrated in [5]:

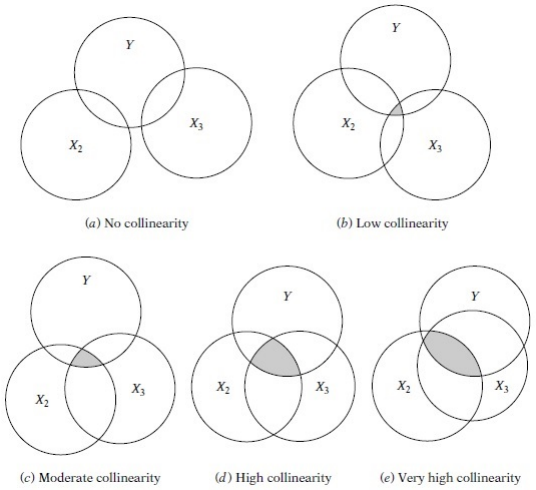


Figure 1: Collinearity levels

Another aspect considered by [5] is the issue of autocorrelation in the data. Since the LSDV model uses OLS estimation it is worthwhile to note the consequences of estimation in the presence of autocorrelation. [5] states that the OLS estimators are still linear unbiased and consistent but they are no longer efficient i.e. they no longer have minimum variance. [5] further remarks that in this case the confidence intervals for the inefficient parameters may be wider than those derived from the generalised least squares procedure (GLS). Which means that it is very likely to fail to reject the hypothesis of insignificance i.e. declare a parameter estimate as not statistically different from zero while in fact it may be statistically different under the GLS estimation method. [5] produces the following to illustrate the confidence intervals under the two methods given the presence of autocorrelation:

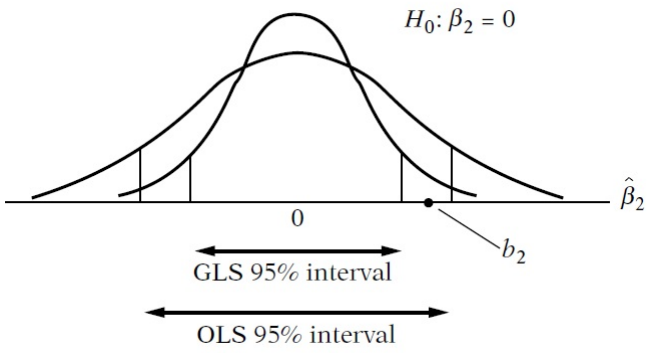


Figure 2: Confidence intervals under GLS and OLS

[5] further mentions that a way to detect autocorrelation is through the Durbin-Watson d test. The decision is based on the following image produced by [5]:

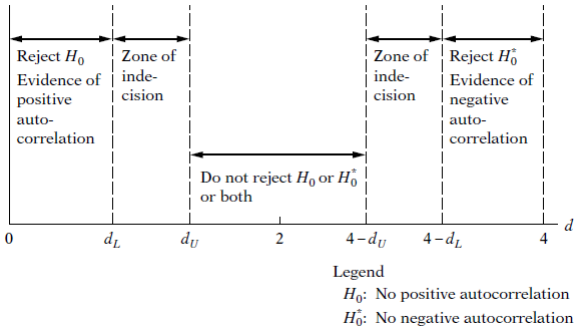


Figure 3: Durbin-Watson Criteria

where d_L and d_U are the critical d-values. It is seen that a d-value very close to 2 implies no autocorrelation . The d-statistic is also used to detect possible model miss-specification.

2 Theory

In this section the basic theory will be discussed in the context of the practical example to be considered in section 3. For illustrative purposes, data obtained from [5] is used which was reproduced from Vinod [1981]. The data is from a famous study by Y. Grunfeld. Grundfeld had his interests in investigating the dependency of a company's real gross investment (Y) on the basis of the real capital stock (X_3) as well as the real value of the company (X_2). The initial investigation was based on several firms, but for the purpose of illustration of the FE approach only the following firms are considered: the first firm being General Electric (GE), the second firm considered is General Motors (GM), the third firm is the U.S. Steel (US) and finally the last firm considered is Westinghouse (WH). The following tables show the data captured for the three variables for the years 1935 to 1954 for each company *Comp*.

Time	Year	Comp	Y	X2	X3
1	1935	1	33,1	1170,6	97,8
2	1936	1	45	2015,8	104,4
3	1937	1	44,6	2803,3	118
4	1938	1	48,1	2039,7	156,2
5	1939	1	74,4	2256,2	172,6
6	1940	1	113	2132,2	186,6
7	1941	1	91,9	1834,1	220,9
8	1942	1	61,3	1588	287,8
9	1943	1	56,8	1749,4	319,9
10	1944	1	93,6	1687,2	321,3
11	1945	1	159,9	2007.7	319,6
12	1946	1	147,2	2208,3	346
13	1947	1	146,3	1656,7	456,4
14	1948	1	98,3	1604,4	543,4
15	1949	1	93,5	1431,8	618,3
16	1950	1	135,2	1610,5	647,4
17	1951	1	157,3	1819,4	671,3
18	1952	1	179,5	2079,7	726,1
19	1953	1	189,6	2371,6	800,3
20	1954	1	317,6	2759,9	888,9

Table 1: Data for General Electric

Time	Year	Company	Y	X2	X3
1	1935	2	317,6	3078,5	2,8
2	1936	2	391,8	4661,7	52,6
3	1937	2	410,6	5387,1	156,9
4	1938	2	257,7	2792,2	209,2
5	1939	2	330,8	4313,2	203,4
6	1940	2	461,2	4643,9	207,2
7	1941	2	512	4551,2	255,2
8	1942	2	448	3244,1	303,7
9	1943	2	499,6	4053,7	264,1
10	1944	2	547,5	4379,3	201,6
11	1945	2	561,2	4840,9	265
12	1946	2	688,1	4900	402,2
13	1947	2	568,9	3526,5	761,5
14	1948	2	529,2	3245,7	922,4
15	1949	2	555,1	3700,2	1020,1
16	1950	2	642,9	3755,6	1099
17	1951	2	755,9	4833	1207,7
18	1952	2	891,2	4924,9	1430,5
19	1953	2	1304,4	6241,7	1777,3
20	1954	2	1486,7	5593,6	2226,3

Table 2: Data for General Motors

Time	Year	Company	Y	X2	X3
1	1935	3	209.9	1362.4	53.8
2	1936	3	355.3	1807.1	50.5
3	1937	3	469.9	2673.3	118.1
4	1938	3	262.3	1801.9	260.2
5	1939	3	230.4	1957.3	312.7
6	1940	3	361.6	2202.9	254.2
7	1941	3	472.8	2380.5	261.4
8	1942	3	445.6	2168.6	298.7
9	1943	3	361.6	1985.1	301.8
10	1944	3	288.2	1813.9	279.1
11	1945	3	258.7	1850.2	213.8
12	1946	3	420.3	2067.7	232.6
13	1947	3	420.5	1796.7	264.8
14	1948	3	494.5	1625.8	306.9
15	1949	3	405.1	1667	351.1
16	1950	3	418.8	1677.4	357.8
17	1951	3	588.8	2289.5	341.1
18	1952	3	645.2	2159.4	444.2
19	1953	3	641	2031.3	623.6
20	1954	3	459.3	2115.5	669.7

Table 3: Data for U.S. Steel

Time	Year	Company	Y	X2	X3
1	1935	4	12.93	191.5	1.8
2	1936	4	25.9	516	0.8
3	1937	4	35.05	729	7.4
4	1938	4	22.89	560.4	18.1
5	1939	4	18.84	519.9	23.5
6	1940	4	28.57	628.5	26.5
7	1941	4	48.51	537.1	36.2
8	1942	4	43.34	561.2	60.8
9	1943	4	37.02	617.2	84.4
10	1944	4	37.81	626.7	91.2
11	1945	4	39.27	727.2	92.4
12	1946	4	53.46	760.5	86
13	1947	4	55.56	581.4	111.1
14	1948	4	49.59	662.3	130.6
15	1949	4	32.04	583.8	141.8
16	1950	4	32.24	635.2	136.7
17	1951	4	54.38	732.8	129.7
18	1952	4	71.78	864.2	145.5
19	1953	4	90.08	1193.5	174.8
20	1954	4	68.6	1188.9	213.5

Table 4: Data for Westinghouse

The basic model (also known as the pooled model) studied in panel data regression has the form:

$$Y_{it} = \beta_1 + \beta_2 X_{2it} + \beta_3 X_{3it} + \epsilon_{it} \quad \text{for } i = 1, 2, \ldots, N, \text{ and } t = 1, 2, \ldots, T, \tag{5}$$

where i and t are the space and time dimensions respectively, i.e there are N cross-sectional units observed over T time periods. The observations from the response variable, Y_{it}, as well as the exogenous variables, X_{2it} and X_{3it}, are used for statistical inference on the parameters (β_1, β_2, and β_3) , with the ultimate goal of determining the estimate model:

$$\hat{Y}_{it} = \hat{\beta}_1 + \hat{\beta}_2 X_{2it} + \hat{\beta}_3 X_{3it}, \quad \text{for } i = 1, 2, \ldots, N, \text{ and } t = 1, 2, \ldots, T,$$

[5] mentions that the variable ϵ_{it} is the error term i.e. the difference over space and time, where:

$$E(\epsilon_{it}) = 0 \text{ and } Var(\epsilon_{it}) = \sigma_\epsilon^2, \quad \text{for } i = 1, 2, \ldots, N, \text{ and } t = 1, 2, \ldots, T,$$

[5] further states that the study of fixed effects approach assumes that the intercept is a non-stochastic variable. Furthermore an advantage of panel data over cross-sectional and time series data is the fact that the least squares dummy variable method, considered in the fixed effects approach, takes into consideration the heterogeneity of the cross-sectional units through the use of individual specific dummy variables. The models to be considered are a result of the assumptions made with regards to the heterogeneity and/or homogeneity of the coefficients corresponding to the individual specific variables.

In this literature it is the assumption that $i = 1$ refers to GE, $i = 2$ refers to GM, $i = 3$ refers to US and finally $i = 4$ refers to WH. A way in which the heterogeneity is taken into account, using the fixed effects approach is by making assumptions about the variation of the coefficients over space and time.

2.1 Model 1: all coefficients constant

Consider the assumption that all coefficients remain constant. With consideration to the data above, there are 4 companies with 20 observations each, therefore there is a total of 80 observations. The basic model (1) may be transformed through stacking the columns of the 4 company observations thus pooling the data to obtain the linear model:

$$Y_j = \beta_1 + \beta_2 X_{2j} + \beta_3 X_{3j} + \epsilon_j \quad \text{for } j = 1, 2, \ldots, 80$$

This is the ordinary least squares (OLS) model. It then follows that the ordinary least squares model is given by the equation:

$$\hat{Y}_j = \hat{\beta}_1 + \hat{\beta}_2 X_{2j} + \hat{\beta}_3 X_{3j}$$

for $j = 1, 2, \ldots 80$ using the unbiased OLS estimators of parameters β_1, β_2, and β_3, [5]. This model is highly restrictive due to the assumption that all the companies have the same coefficients, it is also unrealistic since different companies could have unique individual characteristics that distinguish one company from the other for example experience, size of assets, management style, expertise, technological advancement, level of investment etc. That is, there is heterogeneity among the companies.

2.2 Model 2: Intercept coefficient varies over companies

Gujarati [5] states that one way to take such heterogeneity into account, is by assuming that each company has a unique intercept. It then follows that (1) takes the form:

$$Y_{it} = \beta_{1i} + \beta_2 X_{2it} + \beta_3 X_{3it} + \epsilon_{it} \tag{6}$$

for $i = 1, 2, \ldots, 4$ and $t = 1, 2, \ldots, 20$. The subscript i in the intercept coefficient implies that the intercept varies over each cross-sectional unit $i = 1, 2, 3, 4$ i.e over each of the companies: GE, GM, US and WH respectively. This is indicative of the "individuality" of each company. Gujarati [5] states that how the intercept is actually allowed to vary across the 4 companies is by the use of the **differential intercept dummies** hence model 2 is written as:

$$Y_{it} = \alpha_1 + \alpha_2 D_{2i} + \alpha_3 D_{3i} + \alpha_4 D_{4i} + \beta_2 X_{2it} + \beta_3 X_{3it} + \epsilon_{it} \tag{7}$$

for $i = 1, 2, \ldots$, and $t = 1, 2, \ldots, 20$ where α_1 is the *comparison intercept*, D_{2i}, D_{3i} and D_{4i} are defined as the *dummy variables* and α_2, α_3 and α_4 are defined as the *differential intercept coefficients* for GM, US and WH respectively. This is known as the least squares dummy variable (LSDV) model. It is assumed that:

if $i = 1$, which implies that GE is considered. Then $D_{21} = 0$, $D_{31} = 0$ and $D_{41} = 0$

if $i = 2$, which implies that GM is considered. Then $D_{22} = 1$, $D_{32} = 0$ and $D_{42} = 0$

if $i = 3$, which implies that US is considered. Then $D_{23} = 0$, $D_{33} = 1$ and $D_{43} = 0$

if $i = 4$, which implies that WH is considered. Then $D_{24} = 0$, $D_{34} = 0$ and $D_{44} = 1$

One may have an inquisition as to why there are only three dummy variables even though there are four companies. [5] states that the reason why there are only 3 dummy variables rather than 4 is to avoid the dummy variable trap. This is because the dummy variable trap implies perfect collinearity which results from introducing 4 dummy variables for the 4 companies. This is expected since perfect collinearity contradicts the assumptions for unbiased least squares estimation of the *classical linear regression model*.

From the assumptions, it follows that for GE, i.e. $i = 1$, then the following linear model is obtained:

$$E(Y_{1t}|D_{21} = 0,\, D_{31} = 0,\, D_{41} = 0,\, X_{21t},\, X_{31t}) = \alpha_1 + \beta_2 X_{21t} + \beta_3 X_{31t}$$

for all $t = 1, 2, \ldots, 20$. Hence the intercept for GE is given by the *comparison intercept*, α_1. [5] observes that it then naturally follows that GE is the *comparison* cross-sectional unit.

Now suppose that the intercept for GM is of interest, then $i = 2$, hence it follows by the dummy variable assumptions that for all $t = 1, 2, \ldots, 20$, the linear model for GM is given by:

$$E(Y_{2t}|D_{22} = 1,\, D_{32} = 0,\, D_{42} = 0,\, X_{22t},\, X_{32t}) = (\alpha_1 + \alpha_2) + \beta_2 X_{22t} + \beta_3 X_{32t}$$

Consequently the intercept is given by $(\alpha_1 + \alpha_2)$. If the coefficient α_2 is statistically significant, then the intercept for GM differ significantly from that of GE.

If $i = 3$, then US is considered. It follows from the assumptions that, for US, the linear model obtained is then given by:

$$E(Y_{3t}|D_{23} = 0,\, D_{33} = 1,\, D_{43} = 0,\, X_{23t},\, X_{33t}) = (\alpha_1 + \alpha_3) + \beta_2 X_{23t} + \beta_3 X_{33t}$$

for all $t = 1, 2, \ldots, 20$. It then follows that the intercept coefficient for US is given by $(\alpha_1 + \alpha_3)$. Similarly, should α_3 be statistically significant, it follows that the intercept for US differs significantly from that of GE.

Finally, considering WH, i is set equal to 4. Hence by assumptions it follows that the linear model obtained for WH, for all $t = 1, 2, \ldots, 20$, is then given by:

$$E(Y_{4t}|D_{24} = 0,\, D_{34} = 0,\, D_{44} = 1,\, X_{24t},\, X_{34t}) = (\alpha_1 + \alpha_4) + \beta_2 X_{24t} + \beta_3 X_{34t}$$

Consequently the intercept coefficient for WH is given by $(\alpha_1 + \alpha_4)$. Similarly, if α_4 is statistically significant, it follows that the intercept for WH differs significantly from that of GE.

A formal test may be conducted to verify which model is valid or rather optimal by comparing two opposing models. [5] observes that in relation to *model 2*, *model 1* is restrictive in that it imposes an assumption that all companies have the same intercept thus restricting the individuality of each cross-sectional unit. Hence [5] suggests that the **restricted *F* test** may be used, where *model 1* is the restricted model and *model 2* is the unrestricted model. The null hypothesis to check the validity of the unrestricted model versus the validity of the restricted model is given by:

$$H_0 : \alpha_2 = \alpha_3 = \alpha_4 = 0$$

which is the restricted model and the alternative hypothesis, which gives the unrestricted model, is given by:

$$H_1 : \text{ atleast one of the parameters is different from 0}$$

with the test statistic given by:

$$F = \frac{(R^2_{UR} - R^2_R)/m}{(1 - R^2_{UR})/(n-k)} \sim F(m,\, n-k)$$

Which is F distributed with m numerator degrees of freedom and $n-k$ denominator degrees of freedom. Where m is the number of linear restrictions in the restricted model, $n-k$ is the *degrees of freedom* for the unrestricted model where $n = 80$ is the number of total observations and $k = 6$ is the number of parameters in the unrestricted model. R^2_{UR} and R^2_R are the *coefficients of determination* for the unrestricted and restricted models respectively.

[5] mentions that, given a α*-level of significance,* the null hypothesis is rejected if the computed F value exceeds the critical value $F_\alpha(m, n-k)$ at the α*-level of significance* otherwise it is not rejected. Note that if the null is not rejected this implies that the intercept for all the companies is the same, i.e the *restricted model* is preferred over the *unrestricted* model.

From model 1 it is clear that the *differential intercept coefficients* i.e. α_2, α_3 and α_4 tell by how much the individual intercepts of each cross-sectional unit differ from the *comparison intercept* α_1. In other words, the differential intercept coefficients are used to detect differences between different cross-

sectional units. Note that the comparison company may be any one of the 4 companies. Furthermore in this case, the intercept is *time invariant*, i.e. the intercept of each company is assumed to remain unchanged through time.

[5] mentions the argument made by Kmenta with regards to the LSDV method. Kmenta notes that it is obvious that in the specification of the regression model is the failure to include relevant explanatory variables that are time invariant (and possibly other variables that are time variant but have the same value for all cross-sectional units) moreover the introduction of dummy variables is to cover this ignorance. [5] further argues that if the dummy variables do in fact portray a lack of knowledge about the true model, then it is more appropriate to express this ignorance through the disturbance term. This is the precise rational underlying the **random effects** approach.

The idea is to start off with the regression equation of model 2

$$Y_{it} = \beta_{1i} + \beta_2 X_{2it} + \beta_3 X_{3it} + \epsilon_{it}$$

for $i = 1, ..., 4$ and $t = 1, ..., 20$. Instead of using dummy variables to express β_{1i}, it is defined as

$$\beta_{1i} = \beta_1 + \mu_i$$

for $i = 1, ..., 4$. What this means is that the four companies are drawn from a larger pool of such companies all having a common mean value for the intercept (given by β_1), then the error term μ_i captures the individual differences in the intercept values of each company. By substitution of β_{1i} into model 2 the regression equation is given by

$$Y_{it} = \beta_1 + \beta_2 X_{2it} + \beta_3 X_{3it} + \epsilon_{it} + \mu_i$$

which reduces to the equation

$$Y_{it} = \beta_1 + \beta_2 X_{2it} + \beta_3 X_{3it} + w_{it}$$

where w_{it} is the composite error term consisting of the individual specific error component μ_i as well as the combined time series and cross-sectional error component ϵ_{it}. Thus the individual differences are captured through the error term.

The assumptions underlying the random effects approach are:

$$\epsilon_i \sim N\left(0, \sigma_\epsilon^2\right)$$

$$\mu_{it} \sim N\left(0, \sigma_{\mu}^2\right)$$

$$E(\epsilon_i \mu_{it}) = 0 \text{ and } E(\epsilon_i \epsilon_j) = 0 \quad (i \neq j)$$

$$E(\mu_{it}\mu_{is}) = E(\mu_{it}\mu_{jt}) = E(\mu_{it}\mu_{js}) = 0 \quad (i \neq j; t \neq s)\,.$$

It then follows that

$$E(w_{it}) = 0$$

and

$$Var(w_{it}) = \sigma_{\epsilon}^2 + \sigma_{\mu}^2 = \sigma_w^2$$

Moreover if $\sigma_{\epsilon}^2 = 0$ then the model is exactly the same as **model 1** or rather the **pooled model**. It then follows that the OLS technique may be applied and estimates may be found as before.

As can be seen, the variance of w_{it} is homoscedastic. However, [5] argues that it can be shown that the error terms for any specific individual taken at two points in time are correlated i.e. w_{it} and w_{is} $(s \neq t)$ are correlated . Thus the correlation coefficient follows as:

$$corr(w_{it}, w_{is}) = \frac{\sigma_{\epsilon}^2}{\sigma_{\epsilon}^2 + \sigma_{\mu}^2}$$

It is clearly seen that the correlation coefficient is constant, irregardless of how far apart the observations for the particular cross-sectional unit are. [5] points out that this is a strong contrast to the AR(1) scheme where it is found that the correlation coefficient between two points decreases as the distance between the points increases. Furthermore the correlation coefficient is identical for all cross-sectional units. Therefore if this structure is to be considered, the OLS estimators would be inefficient. Hence the most appropriate method to use is the method of **generalised least squares**.

Since adding dummy variables decreases degrees of freedom and if dummy variables show ignorance as argued by Kmenta it is compelling that the random effects approach is the more optimal approach as it costs less with regards to the degrees of freedom and it reduces the consequences of ignorance of relevant variables.

2.3 Model 3: Intercept coefficient varies over companies and time

In this section an alternative model is considered for taking the heterogeneity of the companies, throughout the years into account. This is achieved by assuming that the intercepts are also *time variant.* Suppose that profit (Y_{it}) for different companies (cross-sectional units) was measured, then time would

indicate experience which might add onto expertise, and with increased expertise it is expected that a company's profitability is affected perhaps through reduced expenses due to less incompetence or rather increased expertise. Moreover, since technology advances year by year, the extent of new technological advancements adopted by a company may also influence profitability through cutting down expenses due to obsolete/inferior processes. Therefore, the intercept for each company may then be representative of the varying initial expenses that each company may have per time period due to the above mentioned factors, i.e. expertise and technological advancement. It therefore follows that **model 3** takes the time factor into consideration resulting in the following equation:

$$Y_{it} = \beta_{1it} + \beta_2 X_{2it} + \beta_3 X_{3it} + \epsilon_{it} \text{ for } i = 1, 2, \ldots, 4 \text{ and } t = 1, 2, \ldots, 20. \tag{8}$$

Here the subscripts i and t indicate that the intercept to the basic model is allowed to vary across both space and time. Similarly, through the use of the dummy variable technique, it follows that **model 3** may be written as:

$$Y_{it} = \beta_o + \sum_{j=2}^{4} \alpha_j D_{ji} + \sum_{k=2}^{20} \lambda_k D_{kt}^{'} + \beta_2 X_{2it} + \beta_3 X_{3it} + \mu_{it} \text{ for } i = 1, 2, \ldots, 4 \text{ and } t = 1, 2, \ldots, 20. \tag{9}$$

It is clear that model 3 is an extension of the LSDV model, where the extension accounts for the time effect. Similarly, β_0 is the *comparison intercept*, α_j's and λ_k's are the *differential intercept coefficients* taking account of the space and time dimensions respectively. The cross-section dummy variables D_{ji}'s still function in exactly the same way as in the case for model 2, i.e. the assumptions hold as previously stated. [5] states that the time dummy variables $D_{kt}^{'}$ have similar assumptions, i.e. in the year 1935 ($t = 1$)

$$D_{k1}^{'} = 0 \text{ for } k = 2, 3, \ldots, 20.$$

In the year 1936 ($t = 2$)

$$D_{22}^{'} = 1 \text{ and } D_{k2}^{'} = 0 \text{ for } k = 3, 4, \ldots, 20.$$

In the year 1937 ($t = 3$)

$$D_{33}^{'} = 1 \text{ and } D_{k3}^{'} = 0 \text{ for } k = 2, 4, 5, 6, , \ldots, 20.$$

As in *model 2*, perfect collinearity is avoided through the introduction of only 19 differential intercept dummies for the 20 time periods.

Consider GE in the year 1935. Then by the assumptions the linear model for GE at the 1^{st} time

period, for $k = 2, 3, \ldots, 20$, is given by:

$$E(Y_{11}|D_{21} = 0,\, D_{31} = 0,\, D_{41} = 0,\, D_{k1}^{'} = 0,\, X_{211},\, X_{311}) = \beta_0 + \beta_2 X_{211} + \beta_3 X_{311}$$

Consequently the intercept is given by β_0. Now suppose that the intercept of GE in the year 1936 is of interest, then it follows from the assumptions that the linear model for GE in the second time period is then given by:

$$E(Y_{12}|D_{21} = 0,\, D_{31} = 0,\, D_{41} = 0,\, D_{22}^{'}\, D_{k2}^{'} = 0,\, X_{211},\, X_{311}) = \beta_0 + \lambda_2 + \beta_2 X_{212} + \beta_3 X_{312}$$

for all $k = 1, \ldots, 20$. The intercept is then given by $\beta_0 + \lambda_2$. If λ_2 is statistically significant it then follows that the intercept for GE in 1935 is significantly different to that of the year 1936.

The restricted F test may also be used to test which model is valid between *model 2* and *model* 3, i.e the *restricted* and *unrestricted* models. The null hypothesis and alternative hypothesis in this case are given by:

$$H_0 : \alpha_2 = \alpha_3 = \alpha_4 = \lambda_2 = \lambda_3 = \ldots = \lambda_{20} = 0$$

$$H_1 : \text{atleast one of the parameters is not equal to } 0$$

The test statistic remains similar, the only difference is that rather than *model 2*, the unrestricted model is now *model 3* and the restricted model remains *model 1*, the rejection criteria remains the same given the desired level of significance. Furthermore, the *differential intercept coefficients* tell by how much the individual intercepts for GM, US and WH, at each time period, differ from the intercept of GE at each point in time. In other words differential intercept coefficients are used to detect differences between the intercept of different cross-sectional units at specific points in time. It is of interest to note that this model is also restricted to some extent, in that it imposes the restriction of common slopes. Therefore the extent of the heterogeneity present in the four companies might still be restricted.

2.4 Model 4: All coefficients vary over companies

[5] observes that another alternative (perhaps even more realistic) of taking the heterogeneity into account is by allowing each company to have varying intercepts and slope coefficients that remain constant across time. This might be due to the unique managerial style each company employs, quality of employees, rules and code of conduct of each company etc. These "quality" factors may have an influence in the

individual slopes of each company. Then equation (1), i.e. the basic model, may be represented as:

$$Y_{it} = \beta_{1i} + \beta_{2i}X_{2it} + \beta_{3i}X_{3it} + \epsilon_{it} \text{ for } i = 1, 2, 3, 4 \text{ and } t = 1, 2, \ldots, 20, \tag{10}$$

Similarly the subscript i in the coefficients indicate that the coefficients vary across each company. As before, differential dummy variables are used resulting in **model 4** taking the form:

$$Y_{it} = \alpha_0 + \sum_{j=2}^{4} \alpha_j D_{ji} + \gamma_1 W_1 + \gamma_2 W_2 + \gamma_3 W_3 + \gamma_4 W_4 + \gamma_5 W_5 + \gamma_6 W_6 + \beta_2 X_{2it} + \beta_3 X_{3it} + \mu_{it} \tag{11}$$

for all $i = 1, 2, 3, 4$ and $t = 1, 2, \ldots, 20$ where $W_1 = D_{2i}X_{2it}$, $W_2 = D_{2i}X_{3it}$, $W_3 = D_{3i}X_{2it}$, $W_4 = D_{3i}X_{3it}$, $W_5 = D_{4i}X_{2it}$ and $W_6 = D_{4i}X_{3it}$.

Since it is known that multiple collinearity could be a problem, in this modeling approach, the VIF measure will also be considered in all the linear models for each individual company. This is done in order to determine the degree of multicollinearity that is present. As stated by [5], multicollinearity poses little problem if the *VIF* for each company is less than 10.

As in the previous models, α_0 is termed the *comparison intercept*, α_j's are the *differential intercepts*. What characterizes *model 4* is the γ's, these are the *differential slopes* and they have the same interpretation as the *differential intercepts*, i.e they tell by how much the slopes for GM, US and WH differ from the slopes of the comparison company GE, [5]. The assumptions for the dummies, with respect to the differential intercepts, are the same as those discussed in the models above (specifically the LSDV method which is model 2). The assumptions with regards to the slope dummies follow in a similar manner as for the intercept dummies discussed in *model 2*, that is:

if $i = 1$, which implies that GE is considered. Then $D_{21} = 0$, $D_{31} = 0$ and $D_{41} = 0$

if $i = 2$, which implies that GM is considered. Then $D_{22} = 1$, $D_{32} = 0$ and $D_{42} = 0$

if $i = 3$, which implies that US is considered. Then $D_{23} = 0$, $D_{33} = 1$ and $D_{43} = 0$

if $i = 4$, which implies that WH is considered. Then $D_{24} = 0$, $D_{34} = 0$ and $D_{44} = 1$.

Now consider GE, then by assumptions it follows that for all $t = 1, 2, \ldots, 20$, the following linear model is obtained:

$$E(Y_{1t}|D_{21} = 0,\, D_{31} = 0,\, D_{41} = 0,\, X_{21t},\, X_{31t}) = \alpha_0 + \beta_2 X_{21t} + \beta_3 X_{31t}.$$

Consequently the intercept coefficient for GE is given by α_0. The slope coefficient for X_{21t} is given by β_2 and the slope coefficient for X_{31t} is given by β_3. Similarly forGM, it follows from the assumptions that for all $t = 1, 2, \ldots, 20$, the linear model follows as:

$$E(Y_{2t}|D_{22} = 1,\, D_{32} = 0,\, D_{42} = 0,\, X_{22t},\, X_{32t}) = \alpha_0 + \alpha_2 + (\gamma_1 + \beta_2)X_{22t} + (\gamma_2 + \beta_3)X_{32t}$$

It then follows that the intercept coefficient for GM is given by $\alpha_0 + \alpha_2$. If α_2 is statistically significant, it follows that the intercept of GM is significantly different from that of GE. Furthermore the slope coefficient for $X_{2,2,t}$ is given by $(\gamma_1 + \beta_2)$ and the slope coefficient of $X_{3,2,t}$ is given by $(\gamma_2 + \beta_3)$. Similarly if γ_1 and γ_2 are statistically significant it follows that the slopes for X_{22t} and X_{32t} are significantly different from the slopes of X_{21t} and X_{31t}, (exogenous variables for GE). For US, it follows from the assumptions that for all $t = 1, 2, \ldots, 20$, the following model is obtained:

$$E(Y_{3t}|D_{23} = 0,\, D_{33} = 1,\, D_{43} = 0,\, X_{23t},\, X_{33t}) = \alpha_0 + \alpha_3 + (\gamma_3 + \beta_2)X_{23t} + (\gamma_4 + \beta_3)X_{33t}$$

It then follows that the intercept coefficient for US is given by $\alpha_0 + \alpha_3$. If α_3 is statistically significant, it follows that the intercept of US is significantly different from that of GE. The slope coefficient for X_{23t} is given by $(\gamma_3 + \beta_2)$ and the slope coefficient of $X_{3,3,t}$ is given by $(\gamma_4 + \beta_3)$, similarly if γ_3 and γ_4 are statistically significant it follows that the slopes for X_{23t} and X_{33t} are significantly different from the slopes of X_{21t} and X_{31t}. Finally, considering WH, by the assumptions it follows that for all $t = 1, 2, \ldots, 20$, the following model is obtained:

$$E(Y_{4t}|D_{24} = 0,\, D_{34} = 0,\, D_{44} = 1,\, X_{24t},\, X_{34t}) = \alpha_0 + \alpha_4 + (\gamma_5 + \beta_2)X_{24t} + (\gamma_6 + \beta_3)X_{33t}$$

It then follows that the intercept coefficient for WH is given by $\alpha_0 + \alpha_4$. Similarly, if α_4 is statistically significant, it follows that the intercept for WH is significantly different from that of GE. The slope coefficient for X_{24t} is given by $(\gamma_5 + \beta_2)$ and the slope coefficient of X_{34t} is given by $(\gamma_6 + \beta_3)$. Similarly if γ_5 and γ_6 are statistically significant it follows that the slopes for X_{24t} and X_{34t} are significantly different from the slopes of X_{21t} and X_{31t}.

It is important to note that the estimates of the parameters discussed above are only valid if the degree of multicollinearity among the variables is small, i.e. the *VIF* of every estimator is less than 10. Moreover, the restricted F test, as before, may be used to check validity of *model 3* against *model 4* i.e. *restricted model* against the *unrestricted model* respectively. The null hypothesis to be tested in this case

is given by:

$$H_0 : \alpha_2 = \alpha_3 = \alpha_4 = \gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 = \gamma_6 = 0$$

with the alternative hypothesis:

$$H_1 : \text{atleast one of the parameters is not equal to } 0$$

The rejection criteria remains similar to the discussion under model 2 given the desired level of significance. Furthermore, similar to the cases for the *differential intercept coefficients,* the *differential slope coefficients*: γ_1, γ_2,γ_3,γ_4,γ_5 and γ_6, tell by how much the individual slope coefficients of X_2 and X_3 for GM, US and WH differ from those of the *comparison* company GE, [5]. In other words differential slope coefficients are used to detect differences between different companies. In the next section models are estimated and the results are then discussed in detail.

3 Application and data interpretation

In this section, the coefficients as determined in the theory section are analysed with the aim to check for statistical significance and thus producing the estimation models. The differences in the coefficients of the four companies are then identified. Moreover the models are analysed to check if they are optimal, this is done through the interpretation of the Durbin-Watson statistic as well as checking how well the model fits the data through the interpretation of the coefficient of determination, i.e. R^2. Since multicollinearity might be a problem when estimating the parameters, VIF values are also considered. The inference done is based on SAS output, of which the code is available in the appendix.

3.1 Comparison of different companies

3.1.1 Model 1: all coefficients constant

As discussed earlier the regression equation is given by

$$Y_j = \beta_1 + \beta_2 X_{2j} + \beta_3 X_{3j} + \epsilon_j \text{ for all } j = 1, 2, \ldots, 80. \tag{12}$$

Ordinary least squares regression produces the following output.

Variable	DF	Parameter estimate	Standard Error	t-value	Pr > |t|	Variance inflation
Intercept	1	-63.27	29.61	-2.14	0.0358	0
X2	1	0.11	0.01	8.02	<0.0001	1.50
X3	1	0.30	0.05	6.15	<0.0001	1.50

R-square	0.7565
Durbin Watson d	0.310
Degrees of freedom	77

Table 5: Model 1 *proc reg* results

From the output results it follows that analysis may continue seeing that the *vif* values are all less than 10. (applying the usual criteria) it follows that under a 5% level of significance, the OLS estimators of the parameters β_1, β_2, and β_3 are all statistically significant since the p-value associated with each t-statistic for the OLS estimators is less than 5%. The coefficient of determination R^2 has a value equal to 0.7565 (this is reasonably high), i.e. approximately 75% of the variation in the dependent variable is explained by the explanatory variables under model 1. This signifies a measure of how well the model fits the data. Furthermore the Durbin-Watson statistic is given by $d = 0.310$. The low d-value implies that there may be autocorrelation in the data, this poses a problem with regards to the unbiased OLS estimation. However given that only about 75% of the data is represented by the model, the low value might also imply model miss-specification [5]. Bearing the possibility of miss-specification in mind, it follows that the model may *not be optimal*, which is expected since it is reasonable to assume heterogeneity of the companies, i.e. each company may have different coefficients due to initial capital in the business, different managerial styles, expertise in field, experience etc.

None-the-less the null hypothesis that the parameters are statistically insignificant is rejected, given the 5% level of significance. It is also evident from the *VIF* values that the estimations are sound due to insignificant collinearity between the random variables. It then follows that the OLS estimation gives the following estimated model:

$$\begin{array}{rcccc} \hat{Y}_{it} & = & -63.27 & + \ 0.1101X_{2j} & + \ 0.3034X_{3j} \\ & & (29.61) & (0.01) & (0.05) \end{array}$$

3.1.2 Model 2: Intercept coefficient varies over companies

Consider model 2 given by the regression equation

$$Y_{it} = \alpha_1 + \alpha_2 D_{2i} + \alpha_3 D_{3i} + \alpha_4 D_{4i} + \beta_2 X_{2it} + \beta_3 X_{3it} + \epsilon_{it} \tag{13}$$

for $i = 1, 2, \ldots, 4$ and $t = 1, 2, \ldots, 20$. Then running the OLS yields the following output

Variable	DF	Parameter estimate	Standard Error	t-value	Pr > |t|	Variance inflation
Intercept	1	-245.81	35.81	-6.86	<0.0001	0
D2	1	161.55	46.46	3.48	0.0009	5.71
D3	1	339.63	23.99	14.16	<0.0001	1.52
D4	1	186.63	31.51	5.92	<0.0001	2.63
X2	1	0.11	0.02	6.17	<0.0001	8.74
X3	1	0.35	0.03	12.98	<0.0001	1.57

R-square	0.9346
Durbin Watson d	1.108
Degrees of freedom	74

Table 6: Model 2 *proc reg* results

The F-statistic from the restricted F test is given by:

$$F = \frac{(R^2_{UR} - R^2_R)/m}{(1 - R^2_{UR})/(n-k)} = \frac{(0.9346 - 0.7565)/3}{(1 - 0.9346)/(80-6)} = 67.19$$

Suppose, a 5% level of significance is used, clearly the F value is significant. It follows that the null hypothesis is rejected, i.e. *model 2* is optimal. Furthermore it is observed from the output that parameters α_2, α_3 and α_4 are statistically significant given the 5% level of significance. This is observed from the significant t values conversely this is also observed from the small p-values which are all less than the level of significance chosen . Hence this leads to the conclusion that all companies have significantly different intercepts i.e. the null hypothesis for each individual t-test is rejected. The coefficient of determination R^2_{UR}, has increased substantially to a value of 0.9346 signifying that approximately 93% of the variability in the observations is explained by ***model 2***. This is a substantial improvement to ***model 1***. Furthermore the Durbin-Watson statistic is given by $d = 1.108$, this value is much closer to 2 than that of ***model 1***, signifying that this model has less collinearity, hence less bias in estimation of parameters. Furthermore this may imply that the model is more correctly specified than the pooled model. This is expected due to the rejection of the null hypothesis in the restricted F test. The VIF value for each parameter estimate is under 10, hence unbiased estimation is possible due to the low level of variable collinearity. Thus the OLS estimates gives the following estimated model:

$$\begin{array}{rccccccccccc} \hat{Y}_{it} = & -245.81 & + & 161.55D_{2i} & + & 339.63D_{3i} & + & 186.63D_{4i} & + & 0.11X_{2it} & + & 0.35X_{3it} \\ & (35.81) & & (46.46) & & (23.99) & & (31.51) & & (0.02) & & (0.03) \end{array}$$

It then follows that the linear model for GE is given by:

$$E(Y_{1t}|D_{21} = 0,\, D_{31} = 0,\, D_{41} = 0,\, X_{21t},\, X_{31t}) = -245.81 + 0.11X_{21t} + 0.35X_{31t}$$

for GM is given by:

$$E(Y_{2t}|D_{22} = 1,\, D_{32} = 0,\, D_{42} = 0,\, X_{22t},\, X_{32t}) = -84.26 + 0.11X_{22t} + 0.35X_{32t}$$

for US is given by:

$$E(Y_{3t}|D_{23} = 0,\, D_{33} = 1,\, D_{43} = 0,\, X_{23t},\, X_{33t}) = 93.82 + 0.11X_{23t} + 0.35X_{33t}$$

and finally for WH is given by:

$$E(Y_{4t}|D_{24} = 0,\, D_{34} = 0,\, D_{44} = 1,\, X_{2,4,t},\, X_{3,4,t}) = -59.18 + 0.11X + 0.35X_{34t}.$$

The following intercept planes graphically represent the heterogeneity inherent in the intercepts for each company:

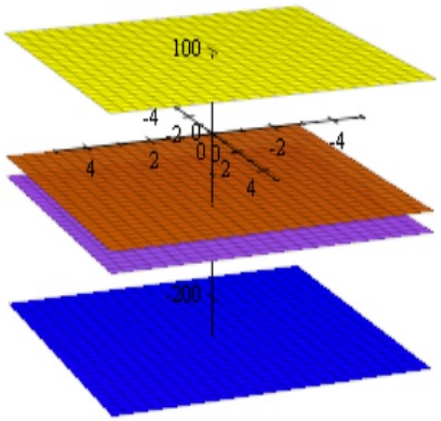


Figure 4: Model 2 intercept planes graphical representation

where the four planes are representative of the intercepts of the four companies with the highest plane belonging to the U.S. Steal, the second highest belongs to Westinghouse the third highest belongs to General Motors and the lowest belonging to General Electric. It is thus evident from the graphical representation that assuming different intercepts for each of the companies is more appropriate, this reflects the fact that restriction on the parameters does not allow for optimal regression model estimation, espe-

cially because each individual company has its own uniqueness that distinguishes it from the rest of the other cross-sectional units or rather companies.

If the random effects approach is considered, then the intercept is a stochastic random variable with expected value β_1. The following regression equation is of interest:

$$\begin{aligned} Y_{it} = & \quad \beta_{1it} + \beta_2 X_{2it} + \beta_3 X_{3it} + e_{it} \\ = & (\beta_1 + u_{it}) + \beta_2 X_{2it} + \beta_3 X_{3it} + e_{it} \\ = & \quad \beta_1 + \beta_2 X_{2it} + \beta_3 X_{3it} + w_{it} \end{aligned}$$

for all $i = 1, ..., 4$ and $t = 1, ..., 20$. with $E(w_{it}) = E(e_{it} + u_{it}) = E(e_{it}) + E(u_{it}) = 0 + 0 = 0$. The following SAS output shows the estimated parameters using the GLS estimation method:

Variable	DF	Parameter estimate	Standard Error	t-value	Pr > |t|
Intercept	1	-73.0353	83.95	-0.8699	0.3870
X2	1	0.1076	0.0168	6.4016	<0.0001
X3	1	0.3457	0.0168	13.0235	<0.0001

Random effect:	GE	-169.93
	GM	-9.51
	US	165.56
	WH	13.87

R^2	0.9323
Degrees of freedom	77

Table 7: Model 2 GLS procedure results

Upon comparison of the results, it is seen that the parameter estimates for the exogenous variables remain essentially the same. It then follows that the intercept for GE is given by $-169.93 - 73.0353 = -242.9653$. Similarly the intercept for GM is -82.5453, for US is 92.5247 and finally the intercept of WH is -59.1653. This is also further evidence that the intercepts for each company are significantly different. It is also worth noting that the intercepts are very close to those obtained using the fixed effects The highest intercept is again given by that of *U.S. Steel*, the second highest is again given by *Westinghouse* and the 3^{rd} largest is also again given by *General Motors* and the lowest is again given by *General Electric*. It is also seen that the coefficient of determination is approximately equal to that of the fixed effects model. But it follows that the degrees of freedom for the random effects model are more. Thus random effects proves to be the more efficient approach since estimation takes place with more degrees of freedom and hence with more efficiency.

3.1.3 Model 3: Intercept coefficient varies over companies and time

$$Y_{it} = \beta_o + \sum_{j=2}^{4} \alpha_j D_{ji} + \sum_{k=2}^{20} \lambda_k D_{kt} + \beta_2 X_{2it} + \beta_3 X_{3it} + \mu_{it} \text{ for } i = 1, 2, \ldots, N \text{ and } t = 1, 2, \ldots, T \quad (14)$$

Variable	DF	Parameter estimate	Standard Error	t-value	Pr > |t|	Variance inflation
Intercept	1	-359.59	82.64	-4.35	<0.0001	0
D2	1	105.30	67.68	1.56	0.1255	11.54
D3	1	341.11	24.81	13.75	<0.0001	1.55
D4	1	220.37	41.18	5.35	<0.0001	4.27
D35	1	134.38	72.01	1.87	0.0674	3.31
D36	1	87.37	66.46	1.31	0.1941	2.82
D37	1	29.64	66.11	0.45	0.6556	2.79
D38	1	48.14	66.83	0.72	0.4743	2.85
D39	1	-7.85	63.92	-0.12	0.9027	2.61
D40	1	51.89	63.79	0.81	0.4195	2.60
D41	1	107.76	63.45	1.70	0.0951	2.57
D42	1	118.38	64.93	1.82	0.0737	2.69
D43	1	72.02	63.48	1.13	0.2614	2.57
D44	1	68.50	63.66	1.08	0.2866	2.59
D45	1	44.64	62.84	0.71	0.4805	2.52
D46	1	104.23	61.83	1.69	0.0975	2.44
D47	1	100.20	62.87	1.59	0.1167	2.52
D48	1	92.30	63.17	1.46	0.1497	2.55
D49	1	31.21	62.11	0.50	0.6174	2.46
D50	1	35.80	61.19	0.59	0.5608	2.39
D51	1	47.85	57.53	0.83	0.4092	2.11
D52	1	57.97	56.39	1.03	0.3085	2.03
D53	1	54.02	55	0.98	0.3303	1.93
X2	1	0.13	0.02742	4.71	<0.0001	20.41
X3	1	0.37	0.04	8.82	<0.0001	3.65

R-square	0.9489
Durbin Watson d	1.110
degrees of freedom	55

Table 8: Model 3 *proc reg* results

The F-statistic from the restricted F test is given by:

$$F = \frac{(R^2_{UR} - R^2_R)/m}{(1 - R^2_{UR})/(n-k)} = \frac{(0.9489 - 0.7565)/22}{(1 - 0.9489)/(80 - 25)} = 9.41$$

Suppose a 10% level of significance, it is clear that $F > F_{0.10}(22, 55) = 1.53$. It follows that the null hypothesis is rejected, i.e. *model 3* is optimal. From the output, under a 10% level of significance it is seen that only 4 of the *time differential intercept coefficients* statistically significant. Hence the intercept also varies over time for only 4 of the years considered.

Furthermore since α_3 and α_4 are statistically significant given the 5% level of significance, it then

Variable	DF	Parameter estimate	Standard Error	t-value	Pr > |t|	Variance inflation
Intercept	1	-9.96	76.35	-0.13	0.8966	0
D2	1	-139.51	109.28	-1.28	0.2061	38.91
D3	1	-40.12	129.23	-0.31	0.7572	54.41
D4	1	9.29	93.12	0.10	0.9208	28.25
D2X2	1	0.09	0.04	2.18	0.0324	116.14
D2X3	1	0.22	0.06	2.24	0.0283	54.49
D3X2	1	0.14	0.06	2.24	0.0283	54.49
D3X3	1	0.26	0.12	2.13	0.0365	5.65
D4X2	1	0.03	0.11	0.24	0.8108	20.67
D4X3	1	-0.06	0.38	-0.16	0.8734	5.72
X2	1	0.03	0.04	0.70	0.4857	50.36
X3	1	0.15	0.06	2.43	0.02	10.65

R-square	0.9512
Durbin Watson d	1.090
Degrees of freedom	68

Table 9: Example

follows that the intercepts for US and WH are significantly different from the intercept of GE. Moreover, since α_2 is statistically insignificant given the 5% level of significance the intercept of GM is not significantly different from that of GE. Again, the significance is evident p-values being smaller or larger than 5% as discussed under *model 2* above. It is also interesting to note that the differential coefficient for GM is now insignificant in this model. The R^2 value has also improved from that of the restricted model, i.e. *model 3* is a better fit for the data accounting for approximately 94.89% of the variation within the data. The Durbin-Watson statistic is given by $d = 1.110$ this is also closer to 2 than that for the restricted model. Implying that the model is more optimal which is expected due to the rejection of the null hypothesis in the restricted F test.

Upon analysis of the *VIF* values it is evident that the model is appropriate i.e. the model produced from the statistically significant parameters has insignificant autocorrelation to affect unbiased estimation. The OLS estimation equation is obtained in exactly the same way as was done in the previous subsections.

3.1.4 Model 4: All coefficients vary over companies

$$Y_{it} = \alpha_0 + \sum_{j=2}^{4} \alpha_j D_{ji} + \gamma_1 W_1 + \gamma_2 W_2 + \gamma_3 W_3 + \gamma_4 W_4 + \gamma_5 W_5 + \gamma_6 W_6 + \beta_2 X_{2it} + \beta_3 X_{3it} + \mu_{it} \quad (15)$$

The following table shows the regression procedure results as per SAS output.

The F-statistic from the restricted F test is given by:

$$F = \frac{(R^2_{UR} - R^2_R)/m}{(1 - R^2_{UR})/(n-k)} = \frac{(0.9512 - 0.7565)/9}{(1 - 0.9512)/(80 - 12)} = 30.14$$

Suppose, a 5% level of significance is used. It is also clear that the F value is significant. It follows that the null hypothesis is rejected, i.e. *model 4* is optimal. From the output, it is evident that α_2, α_3 and α_4 are all statistically insignificant, this is similarly deduced from the p-values of which are all above 20%. Hence it follows that the intercepts of the four companies are not significantly different from one another, in fact they are all equal to 0. Moreover γ_1, γ_2, γ_3 and γ_4 are statistically significant which means that the slope coefficients associated with GM and US are significantly different from the slopes associated with GE. Since γ_5 and γ_6 are statistically insignificant due to their p-values being more than the 5% which is the level of significance of choice. It follows that the slopes associated with WH are not significantly different from the slopes of GE. *Model 4* has the highest coefficient of determination value which suggests that approximately 95% of the variation in the data is explained by the model. The Durbin-Watson statistic is given by $d = 1.090$ this is also closer to 2 than that of the restricted model. This shows that the model is a better fit than the restricted model. Again, this is expected since the null hypothesis of the restricted F test was rejected, indicating that the unrestricted model is optimal. But by the analysis of the VIF values, it is found that most of the values are above 10. This implicates that there is significant a collinearity in the model to bring forth difficult precision of unbiased estimators. Therefore, even though the model is a better fit, it is rejected. The high R^2 value is as a result of multicollinearity.

4 Limitations of the LSDV method and Conclusion

Gujarati [5] mentions that although the preceding models are easily implemented as is demonstrated in section 3, they do however have limitations: The degrees of freedom problem i.e. loss of a degrees of freedom per unit parameter added to the model. Consider *model 3*, for each dummy variable added to the equation a degree of freedom is lost. Gujarati [5] observes that this is particularly a problem when there are too few observations left for meaningful statistical analysis. The solution to the degrees of freedom predicament in estimating panel data regression models is to consider the use of the random effects (RE) approach as an alternative method to the fixed effects approach as discussed in section 3. It is very much important that caution must be taken when using the alternative method of RE, [5] mentions that if the error term and one of the exogenous variables are correlated, then the estimators under the RE are biased.

Furthermore, taking *model 4* into account it is seen that the model has many multiplicative and additive dummies. As such, there is a high possibility of multicollinearity. In fact it is clear that the model does exhibit a lot of multicollinearity among its parameters as observed from the *VIF* values associated with the estimated multiplicative parameters. Not only does multicollinearity contradict the assumptions under the ordinary least squares estimation theory, but it has the consequence that the precise estimation of collinear parameters is difficult. This leads biased estimates.

On the contrary *model 2* is used to detect whether there are inherent differences in the intercepts of General Electric, General Motors, U.S. Steel and Westinghouse. It is observed from the output that parameters α_2, α_3 and α_4 are statistically significant, furthermore the VIF values associated with each parameter is below 10. Hence the estimators are unique minimum variance unbiased estimators i.e. they are efficient. Hence one may conclude with significant confidence that all companies have significantly different intercepts.

Model 3 further seeks to detect the differences in the intercepts for each company in each year, but as is seen, the differential time dummy coefficients are all statistically insignificant if a 5% level of significance is considered. This is evidence that the intercepts for each company remain constant over time. This essentially translates back to *model* 2 but the intercept for GE and GM are not significantly different in this model. This is because dummy variable D_2 is not significant according to the criteria, i.e. the hypothesis of insignificance is not rejected. This seems to be in direct contradiction to the findings in model 2. Closer analysis shows that the VIF value is above 10 with a value of 11.54 i.e. D_2 is deemed significantly collinear. This has the consequence that one is more readily inclined to conclude statistical insignificance when in fact the parameter may be statistically significant under the correct generalised least squares estimation method. [5] makes this remark in his discussion of the consequences of collinearity.

Gujarati [5] further mentions that careful attention must also be on the error term. The assumption made is that the stochastic disturbance ϵ_{it} follows a normal distribution with a mean equal to zero and a constant variance σ^2 i.e. the error term is *homoscedastic*. It therefore follows, that since the error term has both space and time dimensions denoted by i and t respectively a few alterations or rather assumptions may be made e.g. the assumptions of homoscedasticity (as is the case in this discussion) or heteroscedasticity and the assumptions of autocorrelation or non-autocorrelation of the error terms for each individual through time and finally the assumptions of autocorrelation between the error terms

of the different cross-sectional units. [9] further makes the remark that some of the problems that arise from the LSDV can be alleviated by considering the alternative method of FE within group estimator i.e. removing the unobservable effect, as discussed in section 1.

Gujarati [5] discusses the inherent problems that arise when estimation is done in the presence of autocorrelation and collinearity, what is of most importance is that regression assumptions are not violated since unbiased and consistent estimators occur the only fly in the ointment is that the estimators are no longer efficient meaning that they no longer have minimum variance. Because the variance is larger, there is an increased risk of a parameter having a questionable value or more specifically the confidence interval will be wider and thus increasing the chances of failing to reject a parameter estimate that is too far from the actual parameter value. Hence this spells out a degree of less confidence in the estimated parameters.

References

[1] Manuel Arrelano. *Panel data Econometrics*. Oxford University Press, 2003.

[2] Badi H. Baltagi. *Econometric Analysis of Panel Data*. John Wiley & Sons, 2008.

[3] Hsiao R. Cheng. *Analysis of Panel Data*. Cambridge: Cambridge University Press, 1986.

[4] William H. Greene. *Econometric Analysis*. Prentice Hall, N.J., 2000.

[5] Damodar N. Gujarati. *Basic Econometrics*. McGraw-Hill Companies, 2004.

[6] Damodar N. Gujarati. *Econometrics by Example*. Palgrave MacMillan, 2012.

[7] Michael D. Intriligator Kenneth J. Arrow. *Handbook of Econometrics*. North Holland, Amsterdam,New York, Oxford, 1984.

[8] Bo Honore Manuel Arellano. *Panel Data Models: Some Recent Developments*. Princeton University, 2000.

[9] Brent R. Moulton. *Random group effects and the precision of regression estimates*. North Holland, Amsterdam,New York, Oxford, 1986.

[10] Brent R. Moulton. *Diagnostics for group effects in regression analysis*. North Holland, Amsterdam,New York, Oxford, 1987.

[11] Paul A. Rudd. *An introduction to Classical Econometric Theory*. Oxford University Press, 2000.

[12] Jeffrey M. Wooldridge. *Econometric Analysis of Cross Section and Panel Data*. The MIT Press Cambridge, Massachusetts, Landon,England, 2002.

Appendix

Data tables

Time	Year	Comp	Y	X2	X3
1	1935	1	33,1	1170,6	97,8
2	1936	1	45	2015,8	104,4
3	1937	1	44,6	2803,3	118
4	1938	1	48,1	2039,7	156,2
5	1939	1	74,4	2256,2	172,6
6	1940	1	113	2132,2	186,6
7	1941	1	91,9	1834,1	220,9
8	1942	1	61,3	1588	287,8
9	1943	1	56,8	1749,4	319,9
10	1944	1	93,6	1687,2	321,3
11	1945	1	159,9	2007.7	319,6
12	1946	1	147,2	2208,3	346
13	1947	1	146,3	1656,7	456,4
14	1948	1	98,3	1604,4	543,4
15	1949	1	93,5	1431,8	618,3
16	1950	1	135,2	1610,5	647,4
17	1951	1	157,3	1819,4	671,3
18	1952	1	179,5	2079,7	726,1
19	1953	1	189,6	2371,6	800,3
20	1954	1	317,6	2759,9	888,9

Table 10: Data for General Electric

Time	Year	Company	Y	X2	X3
1	1935	2	317,6	3078,5	2,8
2	1936	2	391,8	4661,7	52,6
3	1937	2	410,6	5387,1	156,9
4	1938	2	257,7	2792,2	209,2
5	1939	2	330,8	4313,2	203,4
6	1940	2	461,2	4643,9	207,2
7	1941	2	512	4551,2	255,2
8	1942	2	448	3244,1	303,7
9	1943	2	499,6	4053,7	264,1
10	1944	2	547,5	4379,3	201,6
11	1945	2	561,2	4840,9	265
12	1946	2	688,1	4900	402,2
13	1947	2	568,9	3526,5	761,5
14	1948	2	529,2	3245,7	922,4
15	1949	2	555,1	3700,2	1020,1
16	1950	2	642,9	3755,6	1099
17	1951	2	755,9	4833	1207,7
18	1952	2	891,2	4924,9	1430,5
19	1953	2	1304,4	6241,7	1777,3
20	1954	2	1486,7	5593,6	2226,3

Table 11: Data for General Motors

Time	Year	Company	Y	X2	X3
1	1935	3	209.9	1362.4	53.8
2	1936	3	355.3	1807.1	50.5
3	1937	3	469.9	2673.3	118.1
4	1938	3	262.3	1801.9	260.2
5	1939	3	230.4	1957.3	312.7
6	1940	3	361.6	2202.9	254.2
7	1941	3	472.8	2380.5	261.4
8	1942	3	445.6	2168.6	298.7
9	1943	3	361.6	1985.1	301.8
10	1944	3	288.2	1813.9	279.1
11	1945	3	258.7	1850.2	213.8
12	1946	3	420.3	2067.7	232.6
13	1947	3	420.5	1796.7	264.8
14	1948	3	494.5	1625.8	306.9
15	1949	3	405.1	1667	351.1
16	1950	3	418.8	1677.4	357.8
17	1951	3	588.8	2289.5	341.1
18	1952	3	645.2	2159.4	444.2
19	1953	3	641	2031.3	623.6
20	1954	3	459.3	2115.5	669.7

Table 12: Data for U.S. Steel

Time	Year	Company	Y	X2	X3
1	1935	4	12.93	191.5	1.8
2	1936	4	25.9	516	0.8
3	1937	4	35.05	729	7.4
4	1938	4	22.89	560.4	18.1
5	1939	4	18.84	519.9	23.5
6	1940	4	28.57	628.5	26.5
7	1941	4	48.51	537.1	36.2
8	1942	4	43.34	561.2	60.8
9	1943	4	37.02	617.2	84.4
10	1944	4	37.81	626.7	91.2
11	1945	4	39.27	727.2	92.4
12	1946	4	53.46	760.5	86
13	1947	4	55.56	581.4	111.1
14	1948	4	49.59	662.3	130.6
15	1949	4	32.04	583.8	141.8
16	1950	4	32.24	635.2	136.7
17	1951	4	54.38	732.8	129.7
18	1952	4	71.78	864.2	145.5
19	1953	4	90.08	1193.5	174.8
20	1954	4	68.6	1188.9	213.5

Table 13: Data for Westinghouse

SAS code

```
proc import
datafile = "C:\Users\user\Desktop\honours 2nd semester\WST 795\Final year research project\Book1"
dbms = xlsx
out = work.gujarati;
run;

proc reg
data = gujarati;
id company time;
model Y = X2 X3/dw vif;
run;

proc reg
data = gujarati;
id company time;
model Y = D2 D3 D4 X2 X3/dw vif;
run;

proc reg
data = gujarati;
id company time;
model Y = D2 D3 D4 D35 D36 D37 D38 D39 D40 D41 D42 D43 D44 D45 D46 D47
D48 D49 D50 D51 D52 D53 X2 X3/dw vif;
run;

proc import
datafile = "C:\Users\user\Desktop\honours 2nd semester\WST 795\Final year research project\industry"
dbms = xlsx
out = work.industry;
run;

proc reg
data = industry; /*all constant*/
```

```
id industry time;
model W = Z2 Z3/dw vif;
run;

proc reg
data = industry;/*vary over industry*/
id industry time;
model W = D2 D3 D4 Z2 Z3/dw vif;
run;

proc reg data = industry;/*vary over time*/
id industry time;
model W = D86 D87 D88 D89 D90 D91 D92 D93 D94 D95 D96 D97 D98 D99 D2000 D2001 D2002
D2003 D2004 D2005 D2006 D2007 D2008 D2009 D2010 D2011 D2012 D2013 Z2 Z3/dw vif;
run;

proc reg data = industry;/*all coefficients vary over individuals*/
id industry time;
model W = D2 D3 D4 D86 D87 D88 D89 D90 D91 D92 D93
D94 D95 D96 D97 D98 D99 D2000 D2001 D2002 D2003 D2004
D2005 D2006 D2007 D2008 D2009 D2010 D2011 D2012 D2013 Z2 Z3/dw vif;
run;
```

SAS output

Variable	DF	Parameter estimate	Standard Error	t-value	Pr > |t|	Variance inflation
Intercept	1	-63.27	29.61	-2.14	0.0358	0
X2	1	0.11	0.01	8.02	<0.0001	1.50
X3	1	0.30	0.05	6.15	<0.0001	1.50

R-square	0.7565
Durbin Watson d	0.310
Degrees of freedom	77

Table 14: Model 1 *proc reg* results

Variable	DF	Parameter estimate	Standard Error	t-value	Pr > |t|	Variance inflation
Intercept	1	-245.81	35.81	-6.86	<0.0001	0
D2	1	161.55	46.46	3.48	0.0009	5.71
D3	1	339.63	23.99	14.16	<0.0001	1.52
D4	1	186.63	31.51	5.92	<0.0001	2.63
X2	1	0.11	0.02	6.17	<0.0001	8.74
X3	1	0.35	0.03	12.98	<0.0001	1.57

R-square	0.9346
Durbin Watson d	1.108
Degrees of freedom	74

Table 15: Model 2 *proc reg* results

Variable	DF	Parameter estimate	Standard Error	t-value	Pr > |t|	Variance inflation
Intercept	1	-359.59	82.64	-4.35	<0.0001	0
D2	1	105.30	67.68	1.56	0.1255	11.54
D3	1	341.11	24.81	13.75	<0.0001	1.55
D4	1	220.37	41.18	5.35	<0.0001	4.27
D35	1	134.38	72.01	1.87	0.0674	3.31
D36	1	87.37	66.46	1.31	0.1941	2.82
D37	1	29.64	66.11	0.45	0.6556	2.79
D38	1	48.14	66.83	0.72	0.4743	2.85
D39	1	-7.85	63.92	-0.12	0.9027	2.61
D40	1	51.89	63.79	0.81	0.4195	2.60
D41	1	107.76	63.45	1.70	0.0951	2.57
D42	1	118.38	64.93	1.82	0.0737	2.69
D43	1	72.02	63.48	1.13	0.2614	2.57
D44	1	68.50	63.66	1.08	0.2866	2.59
D45	1	44.64	62.84	0.71	0.4805	2.52
D46	1	104.23	61.83	1.69	0.0975	2.44
D47	1	100.20	62.87	1.59	0.1167	2.52
D48	1	92.30	63.17	1.46	0.1497	2.55
D49	1	31.21	62.11	0.50	0.6174	2.46
D50	1	35.80	61.19	0.59	0.5608	2.39
D51	1	47.85	57.53	0.83	0.4092	2.11
D52	1	57.97	56.39	1.03	0.3085	2.03
D53	1	54.02	55	0.98	0.3303	1.93
X2	1	0.13	0.02742	4.71	<0.0001	20.41
X3	1	0.37	0.04	8.82	<0.0001	3.65

R-square	0.9489
Durbin Watson d	1.110
degrees of freedom	55

Table 16: Model 3 *proc reg* results

Variable	DF	Parameter estimate	Standard Error	t-value	Pr > |t|
Intercept	1	-73.0353	83.95	-0.8699	0.3870
X2	1	0.1076	0.0168	6.4016	<0.0001
X3	1	0.3457	0.0168	13.0235	<0.0001

Random effect:	GE	-169.93
	GM	-9.51
	US	165.56
	WH	13.87

R^2	0.9323
Degrees of freedom	77

Table 17: Model 2 GLS procedure results

Variable	DF	Parameter estimate	Standard Error	t-value	Pr > |t|	Variance inflation
Intercept	1	-359.59	82.64	-4.35	<0.0001	0
D2	1	105.30	67.68	1.56	0.1255	11.54
D3	1	341.11	24.81	13.75	<0.0001	1.55
D4	1	220.37	41.18	5.35	<0.0001	4.27
D35	1	134.38	72.01	1.87	0.0674	3.31
D36	1	87.37	66.46	1.31	0.1941	2.82
D37	1	29.64	66.11	0.45	0.6556	2.79
D38	1	48.14	66.83	0.72	0.4743	2.85
D39	1	-7.85	63.92	-0.12	0.9027	2.61
D40	1	51.89	63.79	0.81	0.4195	2.60
D41	1	107.76	63.45	1.70	0.0951	2.57
D42	1	118.38	64.93	1.82	0.0737	2.69
D43	1	72.02	63.48	1.13	0.2614	2.57
D44	1	68.50	63.66	1.08	0.2866	2.59
D45	1	44.64	62.84	0.71	0.4805	2.52
D46	1	104.23	61.83	1.69	0.0975	2.44
D47	1	100.20	62.87	1.59	0.1167	2.52
D48	1	92.30	63.17	1.46	0.1497	2.55
D49	1	31.21	62.11	0.50	0.6174	2.46
D50	1	35.80	61.19	0.59	0.5608	2.39
D51	1	47.85	57.53	0.83	0.4092	2.11
D52	1	57.97	56.39	1.03	0.3085	2.03
D53	1	54.02	55	0.98	0.3303	1.93
X2	1	0.13	0.02742	4.71	<0.0001	20.41
X3	1	0.37	0.04	8.82	<0.0001	3.65

R-square	0.9489
Durbin Watson d	1.110
degrees of freedom	55

Table 18: Model 3 *proc reg* results

Variable	DF	Parameter estimate	Standard Error	t-value	Pr > |t|	Variance inflation
Intercept	1	-9.96	76.35	-0.13	0.8966	0
D2	1	-139.51	109.28	-1.28	0.2061	38.91
D3	1	-40.12	129.23	-0.31	0.7572	54.41
D4	1	9.29	93.12	0.10	0.9208	28.25
D2X2	1	0.09	0.04	2.18	0.0324	116.14
D2X3	1	0.22	0.06	2.24	0.0283	54.49
D3X2	1	0.14	0.06	2.24	0.0283	54.49
D3X3	1	0.26	0.12	2.13	0.0365	5.65
D4X2	1	0.03	0.11	0.24	0.8108	20.67
D4X3	1	-0.06	0.38	-0.16	0.8734	5.72
X2	1	0.03	0.04	0.70	0.4857	50.36
X3	1	0.15	0.06	2.43	0.02	10.65

R-square	0.9512
Durbin Watson d	1.090
Degrees of freedom	68

Table 19: Model 4 *proc reg* results

The beta-hyperbolic secant distribution

Resego Matshego 13161556

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor(s): Ms BV Omachar, Co-supervisor(s): Dr PJ van Staden

Department of Statistics, University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

30 October 2017

Abstract

The hyperbolic secant distribution (HSD) is a continuous distribution, bell shaped like the Gaussian distribution with a mean of 0 and variance 1. It can be used as the parent distribution in obtaining other generalized distributions which display varying levels of skewness and kurtosis. Various transformation methods for the HSD are explored, focusing mainly on the beta-hyperbolic secant (BHS) distribution and the transformation method.

The BHS distribution is first introduced to be a weighted function of the hyperbolic secant distribution [6]. It has distribution parameters, $\beta_1 > 0$ and $\beta_2 > 0$ which control the shape of the probability density function and converges to a normal distribution for $\beta_1, \beta_2 \to \infty$. It is more flexible over a range of combinations of skewness and of kurtosis values. This property is useful when considering the model under financial application.

Declaration

I, *Resego Matshego,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____

Resego Matshego

_____

BV Omachar

_____

PJ van Staden

_____

Date

Acknowledgements

This work is based on the research supported in part by the National Research Foundation of South Africa, Grant No. 93955.

STATOMET is thanked for their support.

Contents

List of Figures

List of Tables

1 Introduction

The hyperbolic secant distribution (HSD) is a continuous statistical distribution. It is bell shaped with a mean of 0 and variance 1. It has tails which are heavier than the standard normal distribution. The earliest recognition of the HSD is by [16], where the family of Perk's distributions was initially introduced.

The HSD falls under the exponential family of distributions, but more specifically the univariate natural exponential families (NEF) with quadratic variance function (QVF). This means that the variances are at most quadratic functions of their means for each of the distributions [14]. This family includes distributions such as the binomial, gamma, negative binomial, normal and Poisson. The hyperbolic secant distribution is not as familiar as compared to these statistical models, since there is not a sufficient enough association between them and it.

The cumulative distribution function of the standard HSD, defined by [5] as $F(x) = \frac{2}{\pi} \cdot \arctan(e^x)$, can be evaluated as a finite number of expressions. This implies its use in the financial sector to compute credit risk neutral probabilities of option prices resulting in quick and accurate solutions. In addition, this distribution is capable of generating other distributions i.e. the class is preserved under convolution. It has finite moments and is infinitely divisible with an existing moment-generating function.

A relation between the HSD and the half Cauchy distribution is known to exist. Let Z be a random variable that has the standard Cauchy distribution, defined by the following probability density function,

$$f_z(z) = \frac{1}{\pi(z^2+1)}, \text{ for } z \in \mathbb{R}.$$

Secondly, let $V = |Z|$ be the half Cauchy distribution with density function

$$f_V(v) = \frac{2}{\pi(v^2+1)}, \text{ for } v \geq 0.$$

The HSD random variable, $X = \ln(V)$, which is documented by [5], therefore has a probability density function attained as:

$$\begin{aligned} f(x) &\equiv f_X(x) \\ &= f_V(e^x) \cdot e^x \\ &= \frac{1}{2\cosh(\frac{\pi x}{2})}, \text{ for } -\infty < x < \infty. \end{aligned} \tag{1}$$

A generalization of a distribution through the inclusion of one or more shape parameters, aims to increase its flexibility in terms of its distributional shape. These systems of generalized distributions can then be obtained using various transformation methods. These methods have led to the development of generalized families of distributions that include the Perk's distribution family and the NEF-GHS or Meixner distribution family, where the HSD is generalised by considering the ρ^{th} convolution, for $\rho \in \mathbb{N}$. A study of the generalisations of HSD will be reviewed, focusing on the beta-hyperbolic secant (BHS) distribution and the transformation imposed. The book by [5] will be used as the primary source of reference. Other articles and books such as [11] and [3] are adapted into this report .

The density of the BHS distribution is defined by [5] as:

$$f(x;\beta_1,\beta_2) = \frac{B(\beta_1,\beta_2)^{-1}}{\pi\cosh(x)} \frac{\left[\frac{2}{\pi}\arctan(e^x)\right]^{\beta_1-1}}{\left[1-\frac{2}{\pi}\arctan(e^x)\right]^{1-\beta_2}},\ \text{for } -\infty < x < \infty,$$

where $\beta_1 > 0$ and $\beta_2 > 0$ determine the shape of the density curve and $B(a,b)$ denoting the beta function. The BHS density function is always unimodal and all its moments exist. This distribution is more flexible in comparison to the HSD over a range of combinations of skewness and kurtosis values.

The purpose of this report is to show that the BHS distribution is a better distribution to use to fit over financial data rather than the HSD. Different transformation methods for generating skewed hyperbolic secant distributions are explored, with emphasis on the beta hyperbolic secant distribution and the transformation imposed. An application using the BHS distribution into time series analysis will be illustrated [7].

A literature review is provided in Section 2 to give the reader a theoretical framework of the topic discussed. Key terms are explained along with an understanding of how the HSD and the BHS distribution have been previously studied. The different transformations imposed to generalise the HSD will be reviewed. The HSD and its properties are studied in Section 3. Here the reader will receive a better understanding of this distribution and see the similarities and differences it has with its related distributions such as the normal, logistic and Cauchy distribution.

Section 4 will be a discussion on the beta-hyperbolic secant distribution. The transformation technique used to develop the BHS distribution by [6] includes using the HSD as the symmetric or parent distribution and the beta distribution as the weighted function to provide the newly skewed distribution. Interesting properties of the BHS distribution are briefly discussed. The transformation which results in

the BHS is then applied in finance to a dataset which will then be shown in Section 5. Lastly, a conclusion is drawn up in Section 6, revealing the findings of the report.

2 Literature Review

Despite the extensive research done on the hyperbolic secant distribution, many still do not know of it and its immense application in statistical practice, the financial sector, as well as the relation it has with other statistical models. It had its earliest recognition in [15], where Perks discusses the use of generalised hyperbolic distributions in the graduation of mortality statistics, presenting a brief relevance in finance as an instrument of graduation. This generalised formula enabled observed data of mortalities under different experiences to be represented by a single curve instead of multiple curves as it was previously required.

Talacko, [16] expands on Perk's system of distribution functions, by establishing a connection between the hyperbolic secant distribution and Brownian motion. Focus was laid on the HSD and logistic distribution. It was shown that they belonged to Perk's family by setting the parameter $c = e^{-1}$, with the original formula for Perk's distribution given as:

$$f(x) = \frac{\sum_{i=0}^{m} a_i c^{ix}}{\sum_{i=0}^{n} b_i c^{ix}},$$

for real values a_i, b_i and c. This original distribution function was studied in depth with emphasis on the symmetric case which is where the hyperbolic distribution was properly introduced. Its properties are discussed in more detail but individuals were unable to distinguish an association between it and other statistical models even though it was shown to have a wide application in statistical practice.

Generalisation from [8] of the HSD is incorporated into the two parameter family of probability distributions with characteristic function

$$\mathscr{C}(t) = \mathscr{C}(t; \alpha, \rho) = sech^{\rho}(\alpha t), \text{ for } \alpha, \rho > 0.$$

Functions with characteristic functions of this form are known as the generalised HSD. It was shown that if Y_1 and Y_2 are independent random variables coming from the $N(0,1)$ distribution, then

$$X = \ln \left| \frac{Y_1}{Y_2} \right|$$

follows the hyperbolic secant distribution. This generalised distribution, does not however allow for skewness.

An attempt by [3] was made in order to fill the gap relating to the lack of association between the HSD and other statistical models. The HSD is then shown to be the sixth generator in the univariate natural exponential family (NEF) with quadratic variance function (QVF). Examples are provided, including the Jeffreys' prior for contingency tables and Fischer's analysis of similarity between twins. These distributions, naturally generate the HSD.

Jeffrey's prior for contingency tables uses a multinomial distribution which has a link to the Dirichlet distribution. This can be represented by [3] as

$$(p_{11}, p_{10}, p_{01}, p_{00}) \sim \frac{(X_{11}, X_{10}, X_{01}, X_{00})}{X_{11} + X_{10}, + X_{01} + X_{00}},$$

where X_{ij} are identically and independently $Gamma\left(\frac{1}{2}, \frac{1}{2}\right)/2$ distributed. The chi-squared distribution is known to be a special form of the gamma distribution. From this, the following distribution is obtainable: $X_{ij} \sim Z_{ij}^2/2$, where $Z_{ij} \sim N(0,1)$. With this information in mind, the HSD is generated as

$$Y = 2\log\frac{|C_i|}{\pi},$$

where $C_1 = \frac{Z_{11}}{Z_{10}}$ and $C_2 = \frac{Z_{00}}{Z_{01}}$ are identically and independently distributed (iid) standard Cauchy distributions. Fischer's analysis applies a z-transformation to R where R is the intraclass correlation coefficient of a 2×1 matrix normal distribution, where X_1 and X_2 are symmetric with correlation coefficient ρ. It is then shown that $\arctan(R)$ is a location and scale transformation of the HSD.

The hyperbolic secant distribution is explored in further detail by [5]. It is shown to be comparable to the $N\left(0, \frac{1}{4}\right)$ distribution and having a close link to the logistic distribution which has been used to describe growth of populations. [13] briefly discusses this. Derivation of the density function of the HSD comes from the standard Cauchy distribution. The basic properties of the standard hyperbolic secant distribution are also discussed.

Moreover, in order to show leptokurtic nature, it is easier to compare the HSD with other continuous distributions but in particular to compare it to the standard normal distribution. This is because of the direct comparability it has with the normal distribution. They both possess a similar shape and have the same mean and variance skewness moment ratio while other continuous distributions may exhibit some differences.

Various transformations are imposed in order to generate skewed HSD with varying leptokurtosis and tail behaviour. [5] briefly summarises some of these transformations which include manipulating the scale parameters and the Esscher transformation. The form of the density function of the Esscher transformed random variable is denoted as

$$f(x;h) \equiv \frac{e^{hx} f(x)}{\mathscr{M}(h)},$$

where h is a shape parameter with $f(x)$ and $\mathscr{M}_X(t)$ as the density function and the moment generating function respectively.

For a Gaussian distribution, this transformation generates a new symmetric Gaussian distribution with different location and scale parameter. If the random variable does not come from a Gaussian distribution, for example, the HSD, the Esscher transformation will then produce a skew distribution for $h \neq 0$. The parameter h controls the level of skewness of the distribution and thus if $h = 0$, the resulting distribution is once again symmetric. Such transformations result in generalised families of distributions.

The NEF-GHS or Meixner distribution family, is obtained when the HSD is generalised by considering the ρ^{th} convolution, for $\rho \in \mathbb{N}$, i.e. the integral which expresses where the new function is derived from the overlap of two functions. The NEF-GHS distribution, initially introduced by [14], allows for skewness and high excess kurtosis. Its properties make it easier to calculate risk measures.

A general perspective of skewed distributions arising from symmetric parent distributions is provided by [4] along with the effects it may cause on modality and tail behaviour. A set expression is applied to the HSD to generate what is known as the BHS distribution, explored by [12] and [6]. It arises by selecting the beta cumulative distribution as a weighting function and the HSD as the parent distribution. For simplicity, the location parameter, $\mu \in \mathbb{R}$ and scale parameter, $\sigma > 0$, without loss of generality, are set to their standard values, $\mu = 0$ and $\sigma = 1$, so that the effect of the shape parameters $\beta_1, \beta_2 > 0$ on the kurtosis, symmetry and modality of the distribution can be determined.

3 Hyperbolic Secant Distribution

Definition 1. Let X be a real-valued random variable from the HSD, defined by $X \sim HSD(\mu, \sigma)$. The following functions can be used to characterise the distribution of X:

- The cumulative distribution function:

$$F(x) = \frac{2}{\pi} \arctan\left\{\exp\left[\frac{\pi}{2}\left(\frac{x-\mu}{\sigma}\right)\right]\right\}, \text{ where } x \in (-\infty, \infty).$$

- The probability density function:

$$f(x) = \frac{1}{2\sigma} \frac{1}{\cosh\left[\frac{\pi}{2}\left(\frac{x-\mu}{\sigma}\right)\right]}, \text{ with } x \in (-\infty, \infty).$$

  This is derived from the standard Cauchy distribution, $f_z(z) = \frac{1}{\pi(z^2+1)}, z \in \mathbb{R}$.

- The quantile function is derived as:

$$F^{-1}(p) = \mu + \frac{2\sigma}{\pi} \ln(\tan(\frac{\pi p}{2})), \text{ for } p \in (0,1)$$

  and the corresponding quantile density function is given as:

$$\begin{aligned} f^{-1}(p) &= \frac{2}{\pi} \cdot \frac{\sigma}{\tan(\frac{\pi p}{2})} \cdot \sec^2\left(\frac{\pi p}{2}\right) \cdot \frac{\pi}{2} \\ &= \frac{\sigma}{\sin(\frac{\pi p}{2})\cos\left(\frac{\pi p}{2}\right)}. \end{aligned}$$

  This is simply the first derivative of $F^{-1}(p)$ with respect to p for $p \in (0,1)$.

  The HSD is known to have a higher peak and heavier tails than the normal distribution. Using a SAS program, a comparison of the HSD and the normal distribution will be illustrated in the following example. The skewness and excess kurtosis will also be given in the figures below. The quantile function is used to construct the HSD.

Example 2. Let X be a real-valued random variable from the standard HSD. A SAS program is used to generate 5000 values using the HSD quantile function, $F^{-1}(p) = \frac{2}{\pi} \ln(\tan(\frac{\pi p}{2}))$. The values for $p \in (0,1)$ are generated using the uniform distribution. The corresponding code can be obtained in the Appendix. The following figures give a visual representation of the general shape of the standard HSD compared to the standard normal distribution. A QQ-plot is a scatter plot of the quantiles used to determine whether two populations come from the same distribution. It is also considered as it helps illustrate the kurtosis of the standard HSD against the standard normal distribution.

Figure 1 displays a histogram of the standard HSD with an overlay of the density curve of the stan-

dard normal distribution. The shape of the distribution is displayed by the histogram, while the blue line indicates the density curve of standard normal distribution. The HSD has a higher peak and longer tails than that of the normal distribution. Figure 1 also displays the mean, standard deviation, skewness and excess kurtosis of the HSD.

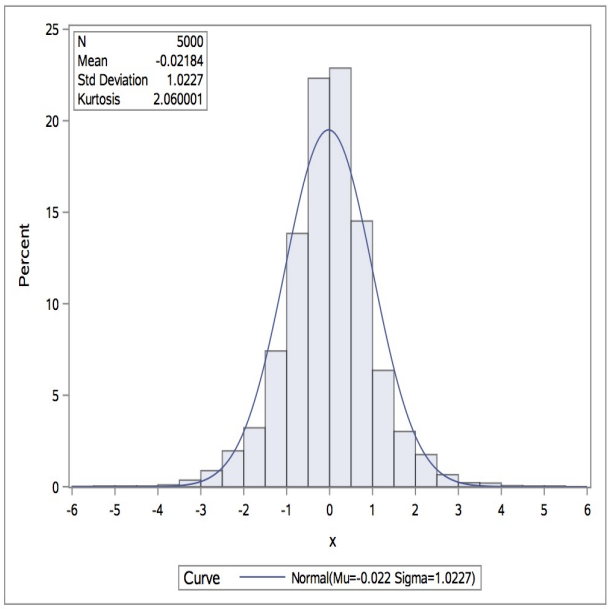


Figure 1: Histogram of the HSD

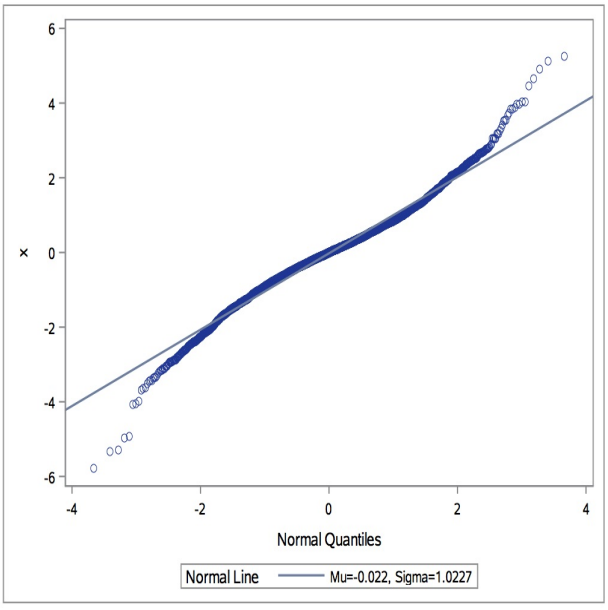


Figure 2: QQ-plot for the HSD

Another comparison of the standard HSD and standard normal is produced in Figure 2 using the QQ-plot of the HSD. A normal distribution would lie on a perfectly straight line. Notice how this graph lies in straight line except for the end points. This confirms that the HSD has heavier tails than the

normal distribution.

3.1 Properties

The standard hyperbolic secant distribution with a given density function, $f(x) = \frac{1}{2} sech\left(\frac{\pi x}{2}\right), x \in \mathbb{R}$ has the following basic properties:

1. The characteristic function, moment-generating function, moments and L-moments:

- The characteristic function of the HSD is defined as

$$\mathscr{C}(t) = sech(t), \text{ for } t \in \mathbb{R}.$$

  It can be used to generate the moment generating function.

- The moment generating function is given as:

$$\begin{aligned} \mathscr{M}(t) &= \mathbb{E}(e^{tX}) \\ &= \mathscr{C}(-\mathbf{i}t) \\ &= \sec(t) \\ &= \frac{1}{\cos(t)}, \text{ for } |t| < \frac{\pi}{2}. \end{aligned}$$

The mean, variance and kurtosis coefficient are derived from the moment generating function in the theorem below:

Theorem 3. *Let X be a real-valued random variable from the standard HSD defined as $X \sim HSD(0,1)$ with a MGF defined as $\mathscr{M}(t) = \sec(t)$. Then the mean, variance, skewness and kurtosis coefficient are given as follows: $\mathbb{E}(X) = 0$, $Var(X) = 1$, $Skew(X) = 0$ and $kurt(X) = 5$.*

Proof. The moment generating function is given as $\mathscr{M}(t) = \sec(t)$. This is used to calculate the first raw moment which is the first derivative of the MGF with respect to $t = 0$:

$$\begin{aligned} m\prime(t) &= \sec\prime(t) \\ &= \tan(t) \cdot \sec(t). \end{aligned}$$

Now:

$$\begin{aligned}\mathbb{E}(X) &= m\prime(0)\\ &= \tan(0)\cdot\sec(0)\\ &= 0.\end{aligned}$$

The variance is:

$$\begin{aligned}var(X) &= \mathbb{E}(X^2) - (\mathbb{E}(X))^2\\ &= m\prime\prime(0)\\ &= \sec^3(0) + \sec(0)\cdot\tan^2(0)\\ &= 1.\end{aligned}$$

The third moment is derived as:

$$\begin{aligned}\mathbb{E}\left(X^3\right) &= m^{(3)}(0)\\ &= 3\sec^3(0)\cdot\tan(0) + 2\sec^3(0)\cdot\tan(0) + \tan^3(0)\cdot\sec(0)\\ &= 0.\end{aligned}$$

The fourth moment is obtained as:

$$\begin{aligned}\mathbb{E}(X^4) &= m^{(4)}(0)\\ &= \sec(0)\cdot\tan^4(0) + 18\sec^3(0)\tan^2(0) + 5\sec^5(0)\\ &= 5.\end{aligned}$$

The third and fourth raw moments will be used to calculate skewness and kurtosis moment ratios of the HSD. Skewness moment ratio of X is:

$$\begin{aligned}skew(X) &= \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right]\\ &= \frac{\mathbb{E}\left(X^3\right) - 3\mu\mathbb{E}\left(X^2\right) + 2\mu^3}{\sigma^3}\\ &= \frac{m^{(3)}(0) - 3\cdot m^{(1)}(0)\cdot m^{(2)}(0) + 2\left[m^{(1)}(0)\right]^3}{1}\\ &= 0,\end{aligned}$$

which verifies symmetry of the HSD.

Lastly, the kurtosis moment ratio is calculated as:

$$\begin{aligned} kurt(X) &= \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] \\ &= \frac{\mathbb{E}\left(X^4\right) - 4\mu\mathbb{E}\left(X^3\right) + 6\mu^2\sigma^2 + 3\mu^4}{\sigma^4} \\ &= 5. \end{aligned}$$

□

Remark 4. Due to the fact that $f(x)$ is symmetric around the mean 0, all the moments and mean coexist. Also, all moments exist and are finite. The raw moments are obtained as $\mathbb{E}(X^k) = \mathscr{M}^{(k)}(0)$, where $\mathscr{M}^{(k)}(t)$ defines the k^{th} raw moment of the random variable. The odd-order raw moments are 0.

Recall that the kurtosis coefficient of the standard normal distribution is 3. The kurtosis coefficient of the standard HSD is 5, so the excess kurtosis of the standard HSD is $kurt(X) - 3 = 2$. This reveals that this distribution has heavier tails and a sharper peak than the normal distribution.

- L-moments:

L-moments were first derived by [9]. Just like the moment generating function, they are used to characterise the probability distribution. They are the expected value of linear combinations of order statistics. [9] derives the r^{th} order L-moments as

$$\lambda_r = r^{-1}\sum_{k=0}^{r-1}(-1)^k\binom{r-1}{k}\mathbb{E}\left(X_{r-k;r}\right) \text{ for } r = 1, 2, \ldots,$$

where $\mathbb{E}\left(X_{r-k;r}\right)$ denotes the expected value of an order statistic which can be written as

$$\mathbb{E}\left(X_{j;r}\right) = \frac{r!}{(j-1)!\,(r-j)!}\int x\left\{F\left(x\right)\right\}^{j-1}\left\{1-F\left(x\right)\right\}^{r-j}dF\left(x\right).$$

Adapting this formula into the HSD, the first four L-moments are:

$$\begin{aligned} \lambda_1 &= \mathbb{E}\left(X_{1:1}\right) \\ &= 0, \end{aligned}$$

$$\lambda_2 = \frac{1}{2}\mathbb{E}\left(X_{2:2} - X_{1:2}\right),$$

$$\lambda_3 = \frac{1}{3}\mathbb{E}\left(X_{3:3} - 2X_{2:3} + X_{1:3}\right)$$

and

$$\lambda_4 = \frac{1}{4}\mathbb{E}\left(X_{4:4} - 3X_{3:4} + 3X_{2:4} - X_{1:4}\right),$$

where λ_1 is the L-location which measure location while λ_2 is the L-scale which measures spread. It is often convenient to standardise the higher moments; λ_r, $r \geq 3$, due to increased variabilities. This is done to generate location or scale invariant measures of shape. [9] then defines L-moment ratios as: $\tau_r = \frac{\lambda_r}{\lambda_2}$ for $r \geq 3$, where τ_3 the L-skewness ratio and τ_4, the L-kurtosis ratio are measures of shape.

2. Tail behaviour and ψ-function:

- The tail behaviour of the hyperbolic secant distribution is equivalent to that of the logistic distribution since they both have heavier tails than that of the normal distribution. The HSD has the following tail function:

  $$1 - F(x) = \frac{e^{-x}}{1 + e^{-x}}$$

  while that of the logistic distribution is:

  $$1 - \frac{e^{x}}{1 + e^{x}} = \frac{1}{1 + e^{x}}.$$

  To verify the similarity of the tail functions, a SAS program is used. This is explained by the following example.

Example 5. Let X be a real-valued random variable from the standard HSD, $f(x) = \frac{1}{2}sech\left(\frac{\pi x}{2}\right)$ and Y a real valued random variable from the standard logistic distribution, $f(x) = \frac{e^x}{(1+e^x)^2}$. The tail functions which describe the tail behaviour of both the probability density functions are used into the SAS, IML procedure. IML is a matrix language which helps calculate statistical procedures which may be too complex to code in SAS.

The same values of x are plugged into the tail functions of the HSD and logistic distribution from a range of -5 to 5. The tail behaviour of the HSD and the logistic distribution are displayed in figure 3. It is clear to see now that the tail behaviour of the HSD is extremely close to that of the logistic distribution. They intersect at the point where $x = 0$ and both have exponentially decaying tails which implies the existence of all moments. The corresponding code can be obtained in the Appendix.

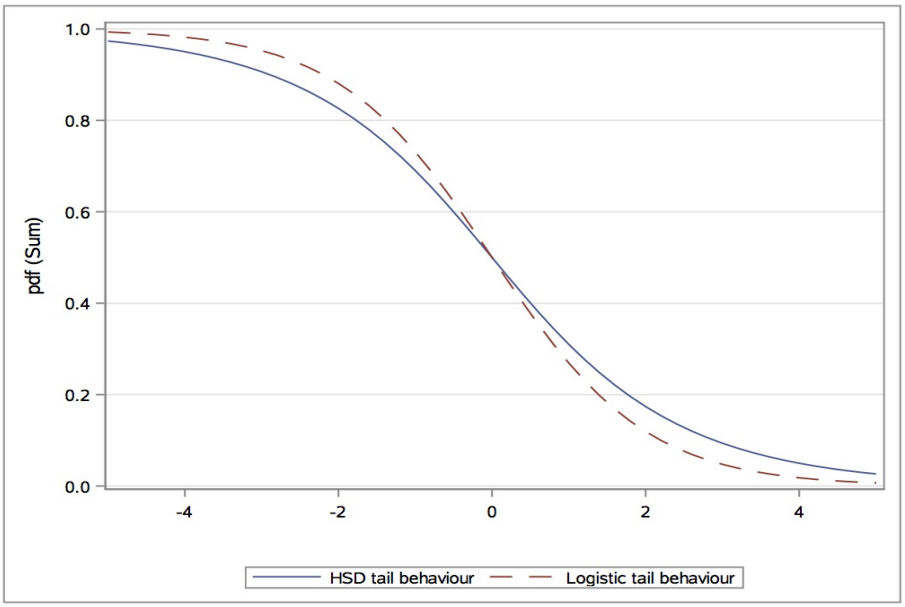


Figure 3: Tail Behaviours of both the HSD and Logistic distribution

- The $\psi-$function is a piecewise continuous function in $\mathbb{R}$ that is used in robust statistics, more particularly robust regression to help remove the influence of outliers and other observations in order to roughly calculate parameters of different models. The hyperbolic secant ψ-function is a partial derivative of the hyperbolic secant distribution which is documented by [5] as:

$$\begin{aligned}\psi(x) &= -\frac{f\prime(x)}{f(x)} \\ &= \frac{e^{2x}-1}{e^{2x}+1}.\end{aligned}$$

3. Self-reciprocality:

The property of self-reciprocality means that the characteristic function, $\mathscr{C}$ and density function, f of a distribution are proportional as defined by [1]. For a constant $\sqrt{2\pi}$, the proportionality of the HSD is shown as follows:

$$\begin{aligned}f\left(x;0,\frac{\pi}{2}\right)\cdot\sqrt{2\pi} &= \frac{\sqrt{2\pi}}{2\cosh\left(\pi\frac{x}{2}\right)} \\ &= \frac{1}{\cosh\left(\sqrt{\pi}\frac{x}{\sqrt{2}}\right)} \\ &= \mathscr{C}\left(x;0,\sqrt{\frac{\pi}{2}}\right).\end{aligned}$$

4. Entropy:

Entropy, (in the context of information theory) is the logarithmic measure of the rate of transfer of information in a particular message or language. Now, the concept of information entropy is completely probabilistic and is considered as a measure of an uncertain random variable related to a random variable X. For a continuous random variable X with density $f_X(x)$, the corresponding differential or Boltzmann-Gibbs-Shannon (BGS) entropy is shown by [5] as:

$$\begin{aligned} H(X) &= H_f(X) \\ &\equiv -\int_{-\infty}^{\infty} f(x)\ln(f_X(x))dx. \end{aligned}$$

X following a Gaussian distribution gives the BGS entropy as, $H(X) = 0.5\ln(2\pi e)$. Considering the case for the standard density function of the HSD, the BGS entropy then reduces to $H(X) = \ln(\pi) + \ln(2)$.

The *maximum entropy (MaxEnt) approach* is also considered. It is demonstrated by [10] depending on finding the best probability distribution having a maximum entropy conditional on whatever information is available. In a more mathematical sense, it is assumed that g denotes some other density with $E_g(\cdot)$ denoting the expected value of the density g. Furthermore, it is further assumed that the $k \times 1$ moment function for finite number k be represented as $\kappa = \kappa(\varepsilon)$. The MaxEnt approach will compute a least biased density function f by maximizing $H(X)$ subject to some given data with moment restriction

$$\mathbb{E}_f(\kappa) = \mathbb{E}_g(\kappa). \tag{2}$$

The general solution to the problem is known as the MaxEnt density, denoted as

$$f(\varepsilon;\lambda) = \frac{1}{C(\lambda)}\exp(-\lambda\prime\kappa(\varepsilon)), \tag{3}$$

with $\varepsilon \in \mathbb{R}$, $\kappa(\varepsilon)$ as the $k\times 1$ parameter, and the normalizing constant $C(\lambda)$ with vector λ, assessed at $\lambda = \lambda_0$. This correlates with the Lagrange multiplier vector which results in the solution presented in (3) to meet the requirements of the moment restriction in (2). [5] provides a solution for the hyperbolic secant distribution which solves the problem when both mean and variance are given: maximize $H(X)$ subject to the restriction in (2) above with $\kappa(\varepsilon) = \ln\{\cosh(\varepsilon)\}$, $C = \pi$ and $\lambda = 1$. Hence this results in a MaxEnt distribution under various moment condition.

5. Relations to other distributions:

The hyperbolic secant distribution is a derivation of the Cauchy distribution or the ratio of two independent random variables coming from $N(0,1)$ distributions. The relation to the half Cauchy is previously shown by taking $X = \ln(V)$, where V is the random variable from the half Cauchy distribution with density

$$f_V(\upsilon) = \frac{2}{\pi(\upsilon^2+1)}, \text{ for } \upsilon \geq 0,$$

resulting in the probability density function of the HSD:

$$f(x) = \frac{1}{2\cosh(\frac{\pi x}{2})}.$$

The second property indicated that the HSD possesses similar tail behaviour to that of the logistic distribution. This association is further explained by [13].

The standard hyperbolic secant density given in Equation 1, which can be rearranged as $f(x) = C_1 \cdot sech(x)$, with $C_1 = \frac{1}{\pi}$. Once the probability density function is squared, the logistic distribution falls into place: $g(x) = C_2 \cdot sech^2(x)$ with

$$C_2 = \left[\int_{-\infty}^{\infty} sech^2(x)dx\right]^{-1} = \frac{1}{2}.$$

In more traditional terms, this can be introduced as the more commonly known normal distribution with a density function given as $h(x) = C_n \cdot sech^n(x)$, where $C_n = \left[2^{n-1}B\left(\frac{n}{2}\cdot\frac{n}{2}\right)\right]^{-1}$.

4 Beta-hyperbolic secant distribution

4.1 Introduction

Numerous techniques are commonly applied to symmetric distributions in order to obtain new distributions with varying levels of skewness and kurtosis. The hyperbolic secant distribution is known to be a base distribution where numerous techniques can be applied to it in order to obtain skewed distributions with modified kurtosis levels. Incorporating [5, 4, 12], describes a skewed distribution G of an original symmetric kernel with a pdf and cdf and cumulative function f and F, respectively to obtain a pdf of the form

$$g(x;\theta) = w \cdot f(x) \cdot (F(x);\theta), \tag{4}$$

provided it exists, where w is a weight function not linked to F on the interval $(0,1)$ with the parameter vector θ. The following lemma is constructed:

Lemma 6. *Let F, w and G have the same form as (4). Then the following holds true:*

i) When w is from a uniform distribution on (0,1) then G will be equal to F.

ii) Let w be fixed and differ from F. It is possible to obtain a symmetric G for any F which is similar to symmetry of w around $\frac{1}{2}$.

Part *i)* directly follows as. G would have to be equal to F provided that w does not transform mass allocation of F into G. In *ii)* however, we have a more interesting case where G can be obtained even if w is not symmetric. In particular, non-uniform densities on $(0, 1)$ can be used to present skewness and altered kurtosis to the original distribution.

Non-uniform densities are required for this and thus from lemma 6 above, the pdf of the (standard) beta distribution is chosen, ie.

$$w(x; \beta_1, \beta_2) = \frac{1}{B(\beta_1, \beta_2)} x^{\beta_1 - 1}(1 - x)^{\beta_2 - 1}, \tag{5}$$

for $\beta_1, \beta_2 > 0$ and $0 < x < 1$. The beta function $B(\cdot, \cdot)$, is denoted as: $B(a, b) = \int_0^1 v^{a-1} - (1 - v)^{b-1} dv$ with $B(a, b) > 0$ for parameters $a, b > 0$. The beta-hyperbolic secant (BHS) distribution is introduced as a weighted hyperbolic distribution with weights originating from (4).

4.2 Definition

Definition 7. Assume X to be a random variable in $\mathbb{R}$ from the standard hyperbolic secant distribution, defined by $X \sim HSD(0, 1)$. Recall that the density function of the HSD is represented as:

$$f(x) = \frac{2}{\pi \left(e^x + e^{-x}\right)}, \text{ for } x \in (-\infty, \infty) \tag{6}$$

with a coinciding cumulative distribution function:

$$F(x) = 1 - \frac{2}{\pi} \arctan\left\{\exp\left(\frac{2}{\pi} x\right)\right\}, \text{ for } x \in (-\infty, \infty) \tag{7}$$

Now, combining Equations (5), (6) and (7) into (4), yields in what is known as the density function of the beta-hyperbolic secant (BHS) distribution. This is defined by [5] as:

$$\begin{aligned} g(x; \beta_1, \beta_2) &= \frac{f(x) \cdot F(x)^{\beta_1 - 1} \cdot (1 - F(x))^{\beta_2 - 1}}{B(\beta_1, \beta_2)} \\ &= \frac{1}{B(\beta_1, \beta_2)} \frac{1}{\pi \cosh(x)} \left[\frac{2}{\pi} \arctan(\exp(x))\right]^{\beta_1 - 1} \cdot \left[1 - \frac{2}{\pi} \arctan(\exp(x))\right]^{\beta_2 - 1}, \end{aligned}$$

where $\beta_1 > 0$ and $\beta_2 > 0$ are the parameters which determine the shape of the density. The cumulative distribution function corresponding with the density function is defined as:

$$G(x; \beta_1, \beta_2) = \frac{B_{F^{-1}(x)}(\beta_1, \beta_2)}{B(\beta_1, \beta_2)}$$

with $B_u(a, b) = \int_0^u v^{a-1} - (1-v)^{b-1} dv$ denoting the beta function.

The BHS distribution is a combination of the HSD density function and its corresponding cumulative function along with a the beta distribution as a weighted function. It then holds a similar shape to that of the HSD. Using a SAS program, the density curve of the BHS is generated. For a symmetric density curve, the shape parameters β_1 and β_2 are then set to be equal in value as there is no influence on the skewness of the distribution. It is also seen that the BHS distribution is unimodal for both $\beta_1 > 0$ and $\beta_2 > 0$.

Figure 4 gives a visual representation of the standard shape of the standard BHS when $\beta_1 = \beta_2 = \beta$. In this case, β is chosen to equal 3. The density function is plugged in directly using the IML procedure, just as it was done in the previous example. The quantile function is not used as in the previous example since it is not as easily identifiable due to the nature of the BHS density function.

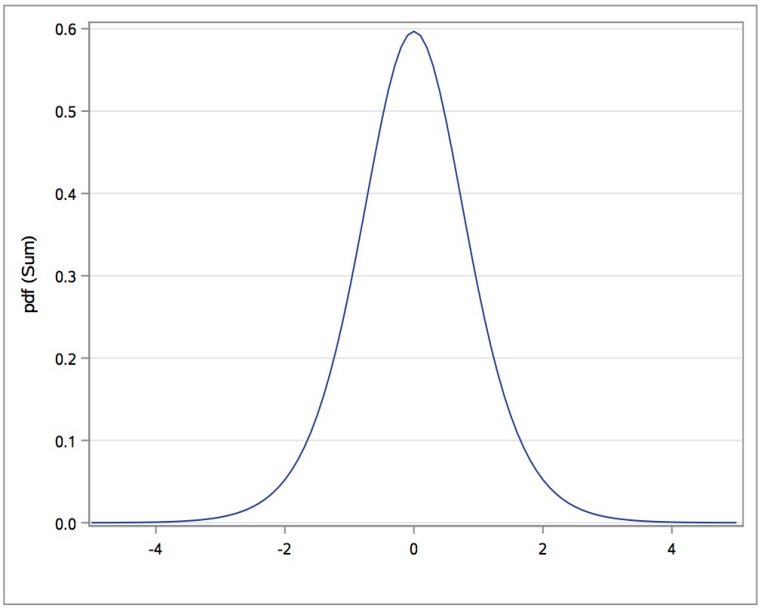


Figure 4: Density curve of the BHS for $\beta_1 = \beta_2 = 3$

4.3 Properties

The BHS distribution has the following properties:

1. Moment generating function and moments:

The k^{th} non-central moment of the BHS density is given as:

$$\mathscr{M}^k(0) = \frac{1}{B(\beta_1, \beta_2)} \int_0^1 \log^k \left(\tan \left(\frac{\pi}{2} u \right) \right) u^{\beta_1 - 1} (1-u)^{\beta_2 - 1} \, du,$$

for $k > 0$, $0 < u < 1$ and $\beta_1, \beta_2 > 0$. From the first four moments, the expected value, variance, skewness and kurtosis coefficients can then be calculated.

Remark 8. Since the BHS distribution has exponential tail behaviour, this implies that all moments exist which can be calculated as $\mathbb{E}(X^k) = \mathscr{M}^{(k)}(0)$, where $\mathscr{M}^{(k)}(t)$ defines the k^{th} moment of the random variable X.

2. Asymmetry and kurtosis:

Consider the general BHS distribution with location parameter $\mu \in \mathbb{R}$, scale parameter $\sigma > 0$ and shape parameters $\beta_1, \beta_2 > 0$. It has the following density function which is given as:

$$g(x; \beta_1, \beta_2) = \frac{B(\beta_1, \beta_2)^{-1}}{\sigma \pi \cosh\left(\frac{x-\mu}{\sigma}\right)} \frac{\left[\frac{2}{\pi} \arctan(\exp\left(\frac{x-\mu}{\sigma}\right))\right]^{\beta_1 - 1}}{\left[1 - \frac{2}{\pi} \arctan(\exp\left(\frac{x-\mu}{\sigma}\right))\right]^{1-\beta_2}}.$$

The shape parameters β_1 and β_2 influence the symmetry and kurtosis of the BHS distribution. The location parameter and scale parameters are then set to equal 0 and 1 respectively in order to obtain the standard shape of the density curves. Figures (5) to (7) demonstrate the effects of the the shape parameters. An inverse relationship is held with the kurtosis level and the shape parameters. It is known that the kurtosis increases as β_1 or β_2 decrease and vice versa. These density functions are constructed for $\beta \in \{0.1, 0.5, 1, 1.5, 2, 2.5, 3\}$ from a range of -6 to 6.

If $\beta_1 = \beta_2 \equiv \beta$, then this distribution will be symmetric around the mean, $\mu = 0$, where a higher kurtosis is obtained as β increases and vice versa. This is demonstrated in Figure 5. The standard HSD density is recovered when $\beta = 1$. This is seen in the curve labelled BHS1. Thus as the value β varies, a generalised family of symmetric distributions of the HSD is achieved. BHS5 represents the density curve where $\beta = 3$ and BHS7 represents the density curve where $\beta = 0.1$.

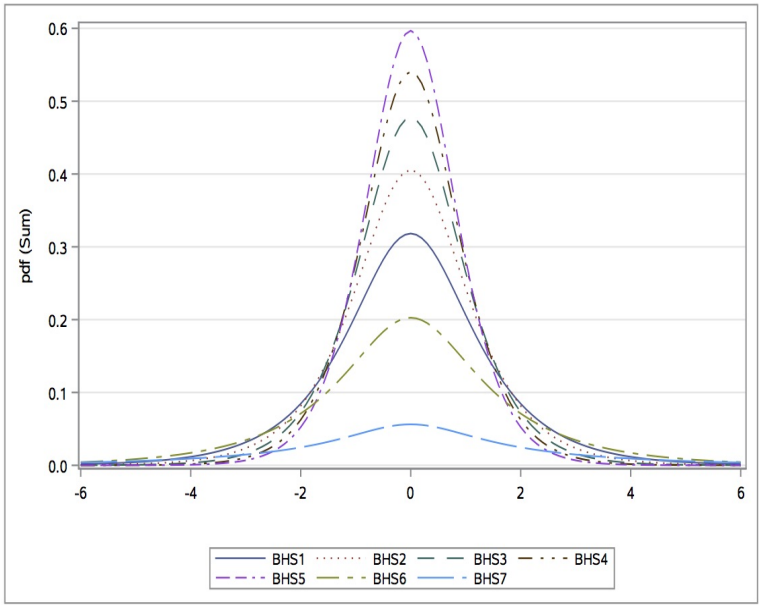


Figure 5: BHS Densities: $\beta_1 = \beta_2 = \beta$: $\beta = \{0.1, 0.5, 1, 1.5, 2, 2.5\}$

Figure 6 and 7 reveal the shape of the distribution when the shape parameters are not equal. Just as earlier it is mentioned that a generalised family of symmetric distributions of the HSD is achieved for varying values of β, a generalised family of asymmetric distributions of the HSD is now obtained for varying combination of values of β_1 or β_2. It is also noted that the skew hyperbolic distribution is achieved when either $\beta_1 = 1$ or $\beta_2 = 1$.

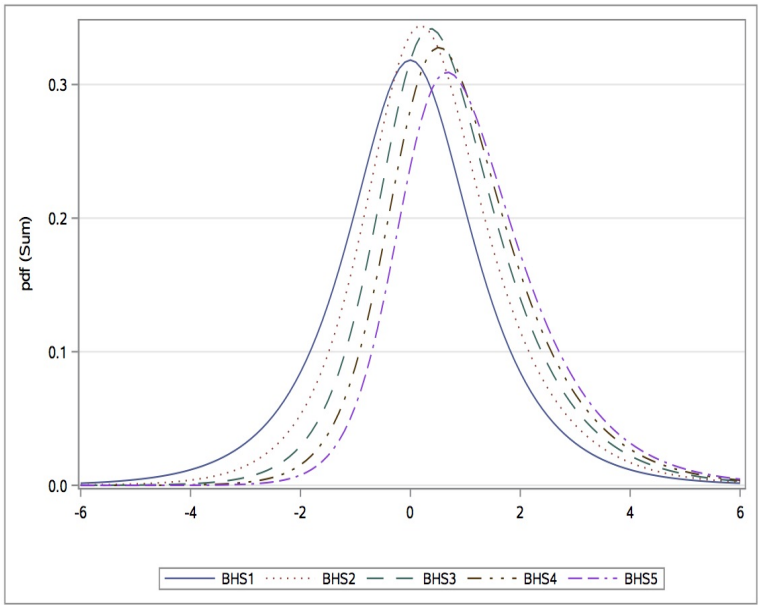


Figure 6: BHS Densities: $\beta_1 > \beta_2$: $\beta_1 = \{1, 1.5, 2, 2.5, 3\}$ and $\beta_2 = 1$

It can be seen that the BHS distribution is positively skewed for $\beta_1 > \beta_2$ which is demonstrated

in Figure 6. Here, the skewness and kurtosis variation are easily observable, where a higher peak is obtained in the density curve labelled BHS2 where $\beta_1 = 1.5$ and $\beta_2 = 1$. BHS5 represents the density curve which lies furthest away from the mean and has the lowest peak with shape parameters valuing at $\beta_1 = 1$ and $\beta_2 = 3$. The density curve labelled BHS1, again represents the HSD case where $\beta_1 = \beta_2 = 1$.

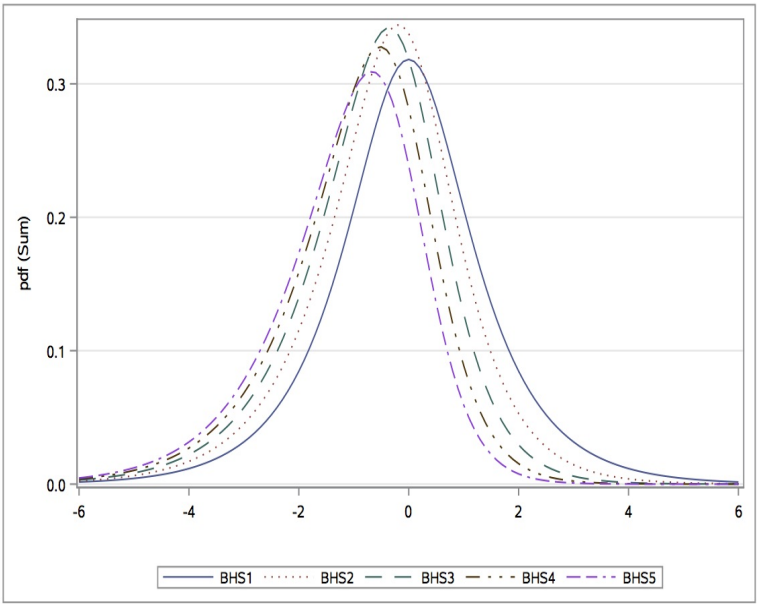


Figure 7: BHS Densities: $\beta_2 > \beta_1$: $\beta_1 = 1$ and $\beta_2 = \{1, 1.5, 2, 2.5, 3\}$

Figure 7 represents density curves which are negatively skewed for $\beta_1 < \beta_2$. The density curve for the symmetric case is again simulated for values $\beta_1 = \beta_2 = 1$. As the value of β_2 increases (while β_1 remains the same at 1), the density curves then shift further to the left, away from the mean value along with the corresponding kurtosis values decreasing.

Remark 9. Figures 6 and 7 substantiates the property earlier stated that kurtosis value is higher when the value of β decreases and vice versa. The same property does not necessarily hold in figure 5, where $\beta_1 = \beta_2 = \beta$. When β increases, the kurtosis level of the BHS density increases too. This then reveals that β_1 and β_2 can not be isolated in order to determine whether any one of these shape parameters affect the kurtosis level. The following result obtained from this is that the kurtosis measures will not be skewness invariant.

3. Tail behaviour and ψ-function:

The beta-hyperbolic secant distribution has tails which decrease exponentially, similar to that of the hyperbolic secant distribution. The BHS ψ-function plays an important role in the theory of

rank test. It is a partial derivative of the BHS distribution and documented by [6] as:

$$\begin{aligned}\psi(x;\beta_1,\beta_2) &= -\frac{g\prime(x;\beta_1,\beta_2)}{g(x;\beta_1,\beta_2)} \\ &= \frac{\tanh(x)\arctan(e^x)(e^{2x}+1)(2\arctan(e^x)-\pi)+(e^x)\beta_1(\pi-2\arctan(e^x))}{(1+e^{2x})\arctan(e^x)(2\arctan(e^x)-\pi)} \\ &- \frac{(e^x)\pi-2(e^x)\arctan(e^x)(2-\beta_2)}{(1+(e^{2x})\arctan(e^x)(2\arctan(e^x)-\pi)}.\end{aligned}$$

This is used to test the sensitivity of the likelihood function to its parameter. If both shape parameters are set to equal 1 then the BHS ψ-function would reduce to $\tanh(x)$.

4. Modality

The BHS distribution is known to have only one mode for both $\beta_1 > 0$ and $\beta_2 > 0$. This is further demonstrated in Figures 4, 5, 6, and 7. For varying combinations of β_1 and β_2, the BHS distribution remains unimodal for varying levels of skewness and kurtosis.

5 Application

Crude oil, otherwise known as unrefined petroleum, is still one of the most commonly traded products in the world. It continues to be the predominant energy source for manufacturing and transportation industries. Hence, the oil price movements possess a significant influence on the economic situation in different countries. The volatility in price changes are high due to the correlation between the supply and demand forces on the international commodity markets. WTI (West Texas Intermediate), Dubai and Brent crude oil are a few of the notable markers of crude oil, which are commonly traded on commodities exchange .

Combinations of the hyperbolic secant distributions have been paid little attention to in their use in financial literature. The purpose of this section is to reveal how well the BHS distribution, as one of the generalisations fits financial data as compared to other symmetric models, such as HSD, normal and SGHS distribution. It has two shape parameters namely, β_1 and β_2 which allows for a more flexible density curve. It is then able to incorporate different levels of skewness and leptokurtosis which then allows for it to fit data better than other symmetric distributions. This application will compare the fit for the HSD and BHS distribution.

Statistics below depicts up-to-date figures on current oil price averages in US Dollars per barrel for 451 months from January 1980 to July 2017. This is adapted from the South African data portal,

http://southafrica.opendataforafrica.org/iaeapfb/monthly-crude-oil-prices. The corresponding code is presented in the Appendix below.

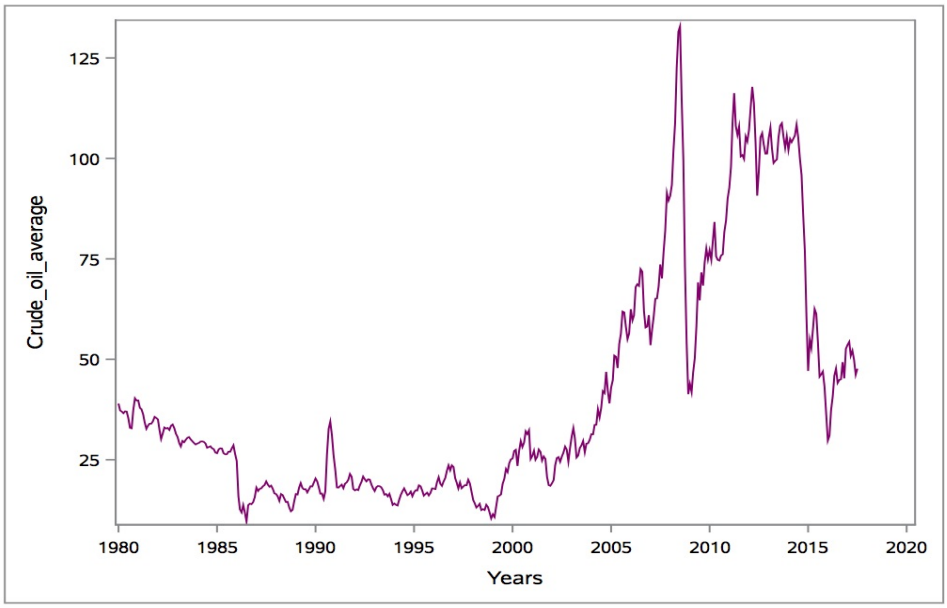


Figure 8: Average monthly crude oil prices

Figure 8 displays the corresponding time series of the raw data along with the corresponding price changes. There is a significant increase in oil price between 2004 and 2008 due to the high demand of oil when the OPEC (Organisation of the Petroleum Exporting Countries) lowered the supply of oil in the years prior to 2004. The drop in oil prices that started in 2008 took place due to The Great Recession, where the demand for oil then dropped drastically.

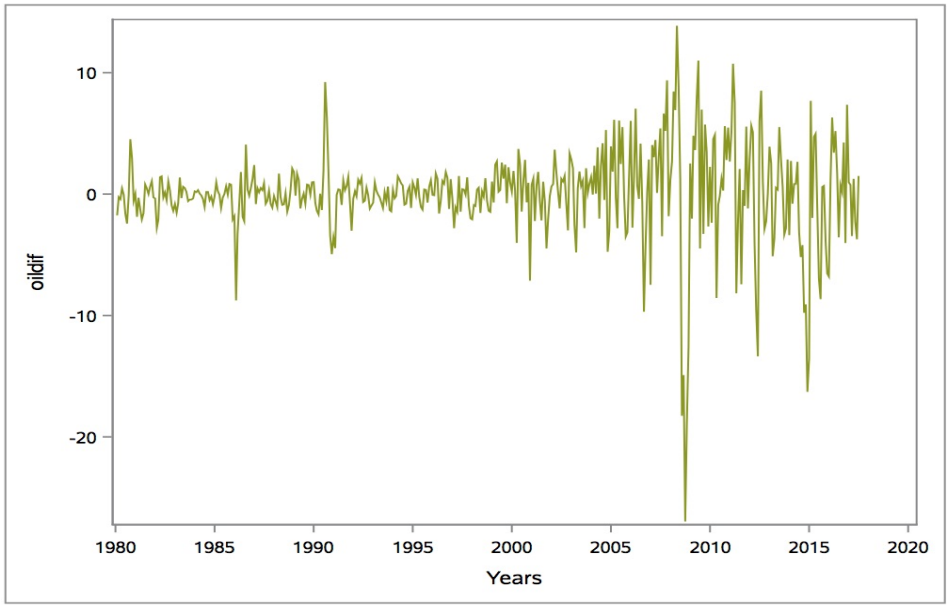


Figure 9: Average monthly crude oil log returns

Figure 9 gives a visual representation of the corresponding log returns of the oil prices which will be incorporated later. These log return values are normalised to reduce variation in the original time series data. This makes it easier to measure values which are comparable and to fit the relevant model. The high peak in 2008 greatly affects the values of the log returns.

Consider the financial dataset related to the average crude oil. The extent to which the observed data matches the theoretical values, otherwise known as, constructing a goodness-of -fit test, will be measured. The criteria placed under consideration are the log-likelihood value, l_N which is obtained from the maximum likelihood estimation, the *Akaike criterion* and the *Bayesian Information Criterion* which are defined as:

$$AIC = -2 \cdot l_N + \frac{2N(k+1)}{N-k-2}$$

and

$$BIC = -2 \cdot l_N + k \cdot \ln(N),$$

respectively. Under the AIC and BIC, l_N represents the log-likelihood value, k represents the number of parameters used in the model and N as the number of observations under consideration. Both are based on the log-likelihood function.

Assume that the underlying log-returns modelled in Figure 9 are independent and identically distributed, defined by [5] as

$$R_t = \mu + U_t \text{ with } U_t \sim D\left(0, \sigma^2, \eta\right) \text{ for } t = 1, \ldots, T.$$

The corresponding theoretical log-likelihood function is defined as $LL\left(\theta\right) = \sum_{i=1}^{N} \ln\left(f_D\left(r_1, \ldots, r_N; \Theta\right)\right)$, where $\Theta = (\mu, \sigma, \eta)$ is a vector defined for the unknown parameters $\mu \in \mathbb{R}$, $\sigma > 0$ and shape parameter η. The maximum likelihood estimator of Θ, denoted as $\hat{l}_{ML}$ is then the solution of the following optimization problem: $\hat{l}_{ML} = \text{argmax}_{\Theta} LL\left(\Theta\right)$. This optimization problem is what will be used to calculate the log-likelihood value l_N with the optimization function *nmlinb* in R software. Rather than maximizing the likelihood function, it is more convenient to work with the negative of the likelihood function which would then need to be minimized. The smallest value of the log-likelihood function will give an indication of which model gives a better fit.

The R-code which is then adapted from [5] is incorporated into both the HSD and BHS distribution. Further calculations are used to calculate the AIC and the BIC for each. The value determined for the AIC can be used to compare various models under the same data set to test which fits best. The model with the smallest AIC value is then preferred. The BIC is similar to the AIC as it also assesses model

fit. It is also sometimes preferred over the AIC as it attempts to lessen the risk of over-fitting by adding the penalty term $k \cdot \ln(N)$, which increases as the number of parameters also increase. This then allows to filter out unnecessarily complicated models, which have too many parameters to be estimated accurately on a given data set of size N [2].

Tables 1 and 2 display the results from the initial optimisation used to calculate the log-likelihood value. Parameters μ, σ, β_1 and β_2 are the values which optimize the objective function l_N. The vales of β_1 and β_2 are fixed to equal 1 in Table 1 since the BHS is stated earlier to be equal to the HSD when $\beta_1 = \beta_2 = 1$. The values of μ and σ are estimated in this case. In Table 2, all 4 parameters are then estimated. It is seen that the parameter values for μ and σ are quite close to one another. The corresponding code is provided for the BHS distribution.

Distribution	μ	σ	β_1	β_2	**Iterations**
HSD	0.4064352	5.0957180	1.0000000	1.0000000	13

Table 1: Optimisation results: The HSD

Distribution	μ	σ	β_1	β_2	**Iterations**
BHS	0.000000	6.000000	3.749994	5.000000	11

Table 2: Optimisation results: The BHS distribution

Table 3 displays the resulting values from the code. From the log-likelihood, AIC and BIC values, it is clear to see that the BHS distribution outperforms the HSD since it has a lower log-likelihood value and corresponding AIC and BIC values which are also lower than those of the HSD. This indicates that the BHS displays a better fit for the crude oil dataset. On thing to also bear in mind, is the similarity in values of the AIC and BIC. One could then have the option of choosing either one or the other formula to test model performance.

Distribution	**k**	l_N	**AIC**	**BIC**
HSD	2	1562.958	3131.97	3138.14
BHS	4	285.5134	581.1616	595.473

Table 3: Goodness-of-fit test for the HSD and BHS distribution

Using a program in R, the density curves of both the hyperbolic secant distribution and that of the beta-hyperbolic secant distribution are simulated and fitted over the histogram of the log returns. Figure 10 gives a visual representation of this. The x-axis represents the log returns and y-axis is represented in terms of percentages. Both density curves are simulated according to scale and use the parameters generated from the optimisation function. The corresponding code for the BHS case can be obtained in the Appendix below.

The graph is slightly negatively skewed. This is confirmed from Table 2 where $\beta_1 = 3.74 < \beta_2 = 5$. From Figure 10, the BHS density curve is represented by the solid green line while that of the HSD is represented by the dotted line. It is then clear to see that the density curve of the HSD has a slight overfit over the given dataset. The BHS density curve is then preferred.

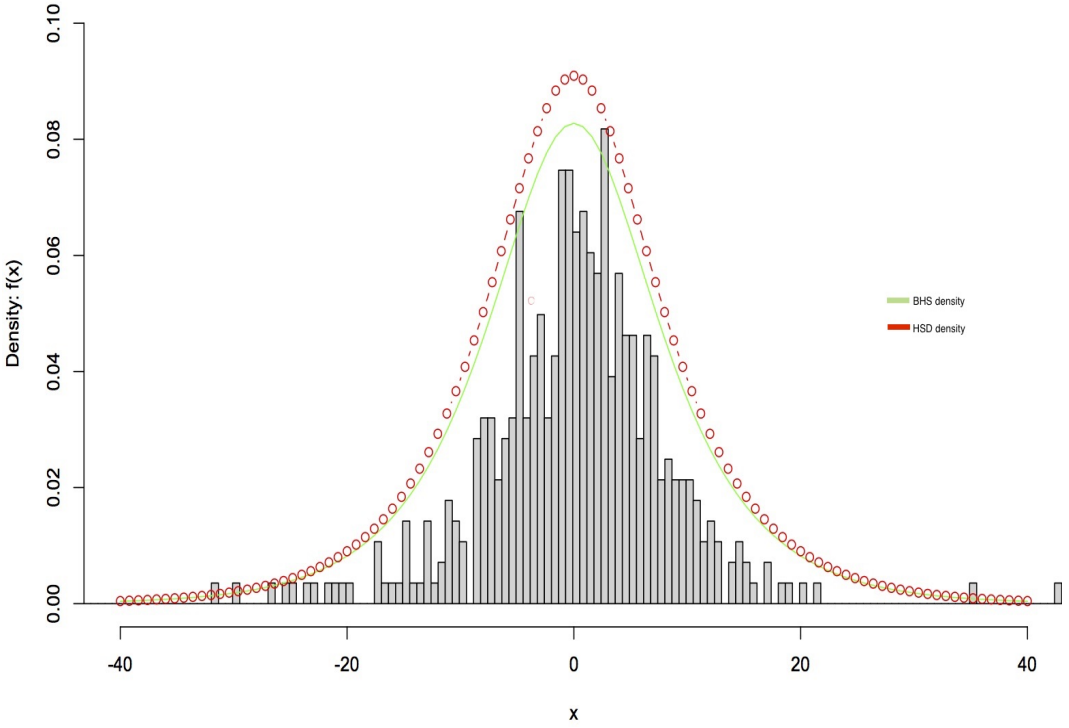


Figure 10: Histogram of the log returns

6 Conclusion

The hyperbolic secant distribution (HSD) is a statistical distribution, possessing a bell-shape, similar to that of the normal distribution. It is not frequently used due to the lack of association between it and other statistical models but then later on shown to fall under the natural and exponential family (NEF) and quadratic variance function (QVF) family through different properties which then come to light. It also revealed that it contains strong use in the financial sector to compute credit risk neutral probabilities of option prices, which results in quick and accurate solutions.

A generalization of a distribution through the inclusion of one or more shape parameters, aims to increase its flexibility in terms of its distributional shape. The beta-hyperbolic secant (BHS) distribution is shown to be a generalisation of the hyperbolic secant distribution, with the HSD as the parent distribution and the beta function as the weighting function of the distribution. It is then observed that an inverse

relationship holds for these shape parameters as the level of kurtosis is seen to decrease as the value of one shape parameter is fixed as the other increases. This is not the case for the symmetric case where $\beta_1 = \beta_2 = \beta$, thus revealing that the kurtosis measures are not skewness invariant.

This transformation allows for more flexibility in terms of shape since the two shape parameters β_1 and β_2 are added to influence the kurtosis and shape of the BHS distribution. Due to the flexibility of the BHS, it can be seen to be a distribution which fits financial data better than other symmetric distributions. The application in Section 5 shows a comparison of the BHS against the HSD and further confirms this result. It is not the most common distribution known in financial literature but results have proven that it holds beneficial use in finance and thus should be taken into a lot more consideration in the future.

References

[1] O. Barndorff-Nielsen, J. Kent, and M. Sørensen. Normal Variance-Mean Mixtures and Z Distributions. *International Statistical Review/Revue Internationale de Statistique*, 50:145–159, 1982.

[2] K. P. Burnham and D. R. Anderson. Multimodel Inference. *Sociological Methods & Research*, 33(2):261–304, 2004.

[3] P. Ding. Three Occurrences of the Hyperbolic-Secant Distribution. *The American Statistician*, 68:32–35, 2014.

[4] J. T. A. S. Ferreira and M. F. J. Steel. A Constructive Representation of Univariate Skewed Distributions. *Journal of the American Statistical Association*, 101:823–829, 2006.

[5] M. J. Fischer. *Generalized Hyperbolic Secant Distributions: With Application to Finance*. Springer, Heidelburg, Germany, 2014.

[6] M. J. Fischer and D. Vaughan. The Beta-Hyperbolic Secant Distribution. *Austrian Journal of Statistics*, 39:245–258, 2016.

[7] J. D. Hamilton. *Time Series Analysis*. Princeton University Press, Princeton, NJ, 1994.

[8] W. L. Harkness and M. L. Harkness. Generalized Hyperbolic Secant Distributions. *Journal of the American Statistical Association*, 63:329–337, 1968.

[9] J. R. M. Hosking. L-moments: Analysis and Estimation of Distributions Using Linear Combinations of Order Statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 52:105–124, 1990.

[10] E. T Jaynes. Information Theory and Statistical Mechanics. *Physical review*, 106:620, 1957.

[11] N. L. Johnson. Systems of Frequency Curves Generated by Methods of Translation. *Biometrika*, 36:149–176, 1949.

[12] M. C. Jones. Families of Distributions Arising From Distributions of Order Statistics. *Test*, 13:1–43, 2004.

[13] E. B. Manoukian and P. Nadeau. A Note on the Hyperbolic-Secant Distribution. *The American Statistician*, 42:77–79, 1988.

[14] C. N. Morris. Natural Exponential Families with Quadratic Variance Functions. *The Annals of Statistics*, 10:65–80, 1982.

[15] W. Perks. On Some Experiments in the Graduation of Mortality Statistics. *Journal of the Institute of Actuaries*, 63:12–57, 1932.

[16] J. Talacko. A Note about a Family of Perks' Distribution. *Sankhyā: The Indian Journal of Statistics*, 20:323–328, 1958.

Appendix

SAS Code for the HSD density curve and QQ-plot:

```
data WST;
call streaminit(13161556); /* set random number seed */
pi=constant('pi');
do i = 1 to 5000;
u = rand("Uniform"); /* u ~ U(0,1) */
x=(2/pi)*log(tan((pi/2)*u)); *quantile function;
output;
end;
run;
proc univariate data=wst;
var u x;
qqplot/normal(mu=est sigma=est) odstitle=none;
histogram x/normal endpoints=-6 to 6 by 0.5 odstitle=none ;
inset n mean stddev kurtosis;
run;
```

SAS Code for the BHS density curve :

```
proc iml ;
be1 = 3 ;
be2 = 3 ;
pp=constant("pi") ;
print be1 be2 pp ;
do x = -6 to 6 by 0.1 ;
fx=2/(pp* (exp(x) + exp(-x))) ;
ffx = 2/pp * atan(exp(2*x/pp)) ;
bf = beta(be1,be2) ;
print x fx ffx bf;
t1=bf**(-1);
t2=fx; t3=ffx**(be1-1);
t4=(1-ffx)**(be2-1);
print t1 t2 t3 t4 ;
```

```
gx = t1*t2*t3*t4 ;
gxd = gxd // (x || gx) ;
print gx ;
end ;
print gxd ;
nm={"x" "pdf"} ;
create gxd from gxd[colname=nm] ;
append from gxd ;
close gxd ;
quit ;
ods graphics / reset width=19cm height=12cm imagemap;
proc sgplot data=gxd ;
title "BHS Density Curve";
vline x/response=pdf lineattrs=(color=red) lineattrs=(thickness=1) legendlabel='BHS' ;
XAXIS TYPE=TIME LABEL = " " FITPOLICY= THIN ;
yaxis grid;
run;
ods graphics / reset;
```

SAS Code for tail behaviour of HSD and logistic distribution:

```
*Tail behaviour of HSD;
proc iml;
pi=constant('pi');
do x = -5 to 5 by 0.2 ;
tffx =1- 2/pi * atan(exp(2*x/pi)) ;
print x tffx;
tffxd=tffxd // (x || tffx);
end ;
print tffxd;
nm2={"x" "pdf"} ;
create tffxd from tffxd[colname=nm2] ;
append from tffxd ;
close tffxd ;
```

```
*Logistic tail behaviour;
proc iml;
pi=constant('pi');
do y = -5 to 5 by 0.2 ;
lffx=1-exp(y)/((1+exp(y)));
print y lffx;
lffxd=lffxd // (y || lffx);
end ;
print lffxd;
nm3={"y" "pdf2"} ;
create lffxd from lffxd[colname=nm3] ;
append from lffxd ;
close lffxd ;
quit;
data plot;
set tffxd lffxd;
run;
proc sgplot data=plot;
vline x/response=pdf lineattrs=(pattern=solid)legendlabel='HSD tail behaviour';
vline x/response=pdf2 lineattrs=(pattern=solid)legendlabel='Logistic tail behaviour';
xaxis type=time label=" " fitpolicy=thin;
yaxis grid;
run;
ods graphics / reset;
```

SAS Code for Average Monthly Crude Oil:

```
options ls=72 nocenter;
*import data;
%let ROOT = /folders/myshortcuts/SAS_Studio/My Folders/WST 795/;
proc import out = oil datafile = "&Root.Average monthly crude oil.xlsx" replace dbms = XLSX;
run;
ods graphics / reset imagemap; proc sgplot data=oil;
;
```

```
/*--Fit plot settings--*/
/*--Scatter plot settings--*/ series x=Title y=Crude_oil_average / linerattrs=(color=CX99006e) trans-
parency=0.0;
/*--X Axis--*/
xaxis label='Years';
/*--Y Axis--*/
yaxis label='Crude oil price $/bbl';;
title "Average Monthly Crude Oil";
run;
ods graphics / reset; data oil;
set oil;
oillag = lag( Crude_oil_average );
oildif = dif( Crude_oil_average );
run;
ods graphics / reset imagemap;
proc sgplot data=oil;
;
/*--Fit plot settings--*/
/*--Scatter plot settings--*/ series x=Title y=oildif / linerattrs=(color=CX879900) transparency=0.0;
/*--X Axis--*/
xaxis label='Years';
/*--Y Axis--*/
yaxis;
title "Lag average Monthly Crude Oil";
run;
ods graphics / reset;
proc print data=oil;
run;
```

SAS Code for symmetry of BHS:

```
*BHS Distribution;
proc iml ;
be1 = 1 ;
be2 = 1 ;
```

```
pp=constant("pi") ;
print be1 be2 pp ;
do x = -6 to 6 by 0.1 ;
fx=2/(pp* (exp(x) + exp(-x))) ;
ffx = 2/pp * atan(exp(2*x/pp)) ;
bf = beta(be1,be2) ;
print x fx ffx bf;
t1=bf**(-1);
t2=fx;
t3=ffx**(be1-1);
t4=(1-ffx)**(be2-1);
print t1 t2 t3 t4 ;
gx = t1*t2*t3*t4 ;
gxd = gxd // (x || gx) ;
print gx ;
end ;
print gxd ;
nm={"x" "pdf"} ;
create gxd from gxd[colname=nm] ;
append from gxd ;
close gxd ;
quit ;
*BHS Distribution2;
proc iml ;
be3 = 1.5 ;
be4 = 1.5 ;
pp=constant("pi") ;
print be3 be4 pp ;
do x = -6 to 6 by 0.1 ;
fx=2/(pp* (exp(x) + exp(-x))) ;
ffx = 2/pp * atan(exp(2*x/pp)) ;
bf = beta(be3,be4) ;
print x fx ffx bf;
```

```
t1=bf**(-1); t2=fx; t3=ffx**(be3-1);
t4=(1-ffx)**(be4-1);
print t1 t2 t3 t4 ;
gx2 = t1*t2*t3*t4 ;
gxd2 = gxd2 // (x || gx2) ;
print gx2 ;
end ;
print gxd2 ;
nm={"x" "pdf2"} ;
create gxd2 from gxd2[colname=nm] ;
append from gxd2 ;
close gxd2 ;
quit ;
....
data Symmetric_graph;
set gxd gxd2 gxd3 gxd4 gxd5 gxd6 gxd7;
run;
proc print data=Symmetric_graph;
run;
proc sgplot data=Symmetric_graph;
vline x/response=pdf lineattrs=(pattern=solid)legendlabel='BHS1';
vline x/response=pdf2 lineattrs=(pattern=dot)legendlabel='BHS2';
vline x/response=pdf3 lineattrs=(pattern=dot)legendlabel='BHS3';
vline x/response=pdf4 lineattrs=(pattern=dot)legendlabel='BHS4';
vline x/response=pdf5 lineattrs=(pattern=dot)legendlabel='BHS5';
vline x/response=pdf6 lineattrs=(pattern=dot)legendlabel='BHS6';
vline x/response=pdf7 lineattrs=(pattern=dot)legendlabel='BHS7';
xaxis type=time label=" " fitpolicy=thin;
yaxis grid;
run;
ods graphics / reset;
```

R Code for BHS optimization:

```
temp<-scan("/Users/resego/Desktop/WST 795/Coal.txt")
temp data<-100*diff(log(temp2)) #Returns suitably lagged and iterated differences.
data
# define density function
BHS.density=function(x,SHAPE){
#b1=SHAPE[1];b2=SHAPE[2]  return(1/beta(b1,b2)/(pi*cosh(x))*((2/pi)*atan(exp(x)))^(b1-1))*(1-
(2/pi)*atan(exp(x)))^(b2-1)
}
BHS.density
# define log-likelihood function
LOGLIKE=function(PARA,DATA){
mu=PARA[1];  sigma=PARA[2];  shape=PARA[3:4]  ll=-sum(log(1/sigma*BHS.density((DATA-
mu)/sigma,shape))) return(ll)
}
LOGLIKE
# start optimisation
result<-nlminb(start=c(3,1,1,1),  obj=LOGLIKE,  lower=c(0,0,1,1),  upper=c(4,6,4,5),  DATA=data,
control=list(trace=1))
result
#AIC value
AIC<-(2)*(285.5134)+(2*451*(4+1)/(451-4-2)) #use objective value
AIC
BIC<-(2)*(285.5134)+4*log(451)
BIC
```

R code to fit BHS density curve over crude oil dataset:

```
temp<-scan("/Users/resego/Desktop/WST 795/Coal.txt")
temp
data<-100*diff(log(temp)) #Returns suitably lagged and iterated differences.
data
```

```
BHS.density=function(x,SHAPE){
b1=SHAPE[1];b2=SHAPE[2]  return(1/beta(b1,b2)/(pi*cosh(x))*((2/pi)*atan(exp(x)))^(b1-1))*(1-(2/pi)*atan(exp(x)))^(b2-1)
}
LOGLIKE=function(PARA,DATA){
mu=PARA[1]; sigma=PARA[2]; shape=PARA[3:4] ll=-sum(log(1/sigma*BHS.density((DATA-mu)/sigma,shape))) return(ll)
}
hist(data, xlim= c(-40,40), ylim= c(0,.1), breaks=seq(min(data),
max(data), length=140), xlab = "x", ylab= "Density: f(x)",
main = "Histogram of log returns of average crude oil",
prob= TRUE, col= "lightgray")
par(oma=c(0,0,0,2))
par(new=T)
plot(BHS.density, xlim= c(-6,6), ylim= c(0,0.35), xlab= "", ylab="", type = "b", col = "red", axes=FALSE)
```

Gaussian processes for regression analysis

Rethabile Matsabu 13129602

STK795 Research Report

Submitted in partial fulfillment of the degree BCom(Hons) Statistics

Supervisor: Dr Alta De Waal

Department of Statistics, University of Pretoria

30 October 2017

Abstract

The objective of this paper is to investigate Gaussian processes (GPs) as an alternative approach for solving non-parametric regression problems. We compare the results of the Gaussian process regression with a local polynomial non-parametric regression. To achieve the comparison of both we use the confidence intervals and SSE as metrics. Both techniques are applied to a very simple non-linear data set.

Declaration

I, *Rethabile Matsabu,* declare that this essay, submitted in partial fulfillment of the degree *BCom(Hons) Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Rethabile Matsabu

_ _

Dr Alta de Waal

_ _

Date

Acknowledgments

I acknowledge the financial support from the Center for AI Research, Meraka Institute, CSIR. I would also like to acknowledge the academic and motivational support that i have received from my supervisor, Dr Alta de Waal. Lastly to acknowledge the opportunity to being part of this program which is grated by the department of statistics, University of Pretoria.

Contents

List of Figures

List of Tables

1 Introduction

If we want to get the most out of data in the information age, we need to consider the analysis of data from a mathematical approach of modern probability theory which gives us an opportunity to take advantage of a framework in which it readily defines the uncertainty, risk of events and outcomes. Bayesian inference offers reasoning in the presences of uncertainty and minimal information provided [9]. We discuss the framework of Bayesian modelling and its concepts in approaching the problem of regression data under uncertainty. The main focus of this research will be the theory and application of Gaussian processes [6].

In many circumstances, little or no prior information exist regarding a suitable model to use when trying to model data. We then find ourselves having to rely on domain knowledge, for example, by observing data we might see some common underlying process which probability distributions are over this kind of function space is evaluated by refining the distributions focusing on its region . Functions like these are not characterised by predefined parameters. This approach is known as non-parametric modelling and serves a key component of the Gaussian process (GP) [9].

Gaussian processes (GPs) can be thought of as an alternative approach in solving regression problems [3], than linear regression. We start with a simple linear regression function $y = f(x) + \epsilon$ where y is the dependent variable, $f(x)$ is the function of independent variables and ϵ is the error term. Assume that the function $f(x)$ has a linear relationship which can be written as $y = \theta_0 + \theta_1 x + \epsilon$. Now we can try to find the parameters θ_0 and θ_1 [6]. A Bayesian linear regression gives probabilistic approach in trying to solve a distribution over the parameters. The parameters are updated as new datapoints are observed. The GP is known as a non-parametric model in the sense that it finds a distribution over a function that is consistent with observed data. Like with all Bayesian methods it starts with a prior distribution over a function then it updates it as it observed new data points, which will lead it to producing the posterior distribution over the functions [10].

2 Literature review

The main focus of this research is the application of GP as an alternative to solving regression problems to the local polynomial nonparametric regression model by comparing the fit to data of the two models. The book by Murphy [6] will give us the majority of our background theory on Gaussian processes and the study of kernels that will be very useful in understanding the Gaussian application. The multivariate Gaussian process theory is made simple by Chuong B Do. [2]. We will use Rasmussen and Christopher KI Williams [8] as a guideline in the approach of further understanding how Gaussian process works in

Machine learning.

In order to understand the GP for regression, Mark Ebden [3] will give us a brief introduction in GP for regression. Christopher KI Williams and Carl Edward Rasmussen [10] will take us through how the process can be applied to regression analysis. We will consult John Fox [4] and John Hughes [5] in understanding nonparametric regression, most importantly the local polynomial regression as our main model of comparison in the standard theory of nonparametric regression models.

Our application is the most important contributor to our conclusion, the work required will be made simple for us with the help provided by the scikit-learn project team [1][7] with the coding work done by their team on Gaussian process for machine learning.

2.1 Bayesian time series analysis

Much work have been done on the application of GP on time series data. In this section, we focus on this specialised application.

The time series form $y(x) = f(x) + \eta$, will be our starting point as a format of a regression problem, where $f(x)$ is an unknown function and η is the white noise. We want to get the probability distribution of $y(x)$ such that $p(y|x)$, to do this we must get the inference by assuming that the is a database of existing observations. Bayesian modeling allows for the inclusion of all considerable data in a set taking all information even past information.

To introduce the ideas or methods of Bayesian modelling we look a simple example at a small set of data samples, placed at x = 0, 1, 2 and the targeted observed values. Least-squares regression on this sample data using a simple model gives form to the curve that appearance as a line in the left panel of Figure 5. We observe that normally this curve fits observed data well. But the question remains what about the region of no-observed values $x \succ 2$.

Working with a distribution over curves in which each offers a description of the data observed is of most importance when modelling with Bayesian. In our example you will find a close relationship between curvature, complexity and Bayesian inference, giving rise to a posterior beliefs over models being an set of how good the data observed is explained. The panel on the rights illustrates curves with similar fit to the data as the least-squares spline. The curves are quite similarly close to the data yet high variability in areas where the data is not observed [9].

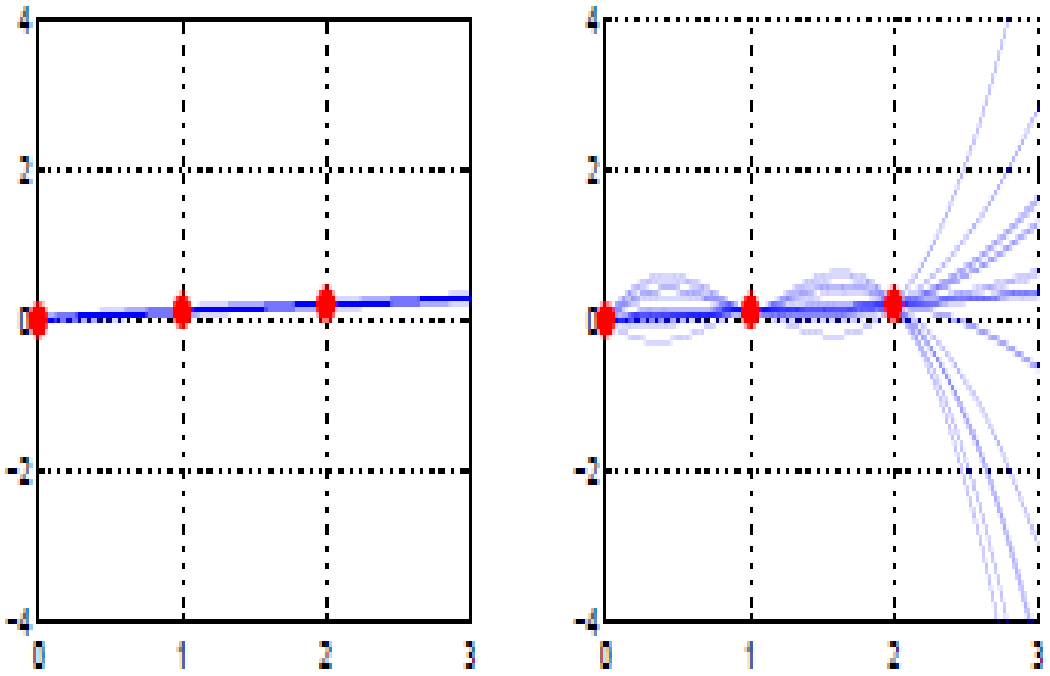


Figure 1: Bayesian modelling simple examples output [9].

3 Background Theory

3.1 Gaussian process

For this section we look at Murphy [6] and Mark Ebden [3].

In a supervised learning environment x_i and y_i are observed as input and output respectively. We assume a unknown function $y_i = f(x_i)$ with a white noise variable. Our goal is to infer a distribution over this function on a specific, $p(f|X, y)$. We do this so as to make predictions given new data, and this is formulated as

$$p(y_*|x_*, X, y) = \int p(y_*|f, x_*)p(f|X, y)df \tag{1}$$

where x_* is new data observed and y_* is the prediction of new data observed, but this is based on a parametric function. For this section we want to apply Bayesian inference over the function. This approach is found in Gaussian processes (GPs) [3].

A GP can be defined as a prior over a function, that forms into a posterior over a function when we observe some data. It has been found to be quite difficult to present a distribution over a function. To

simplify it we need to use a distribution over function's finite values, say a set of points $x_1, ..., x_N$. The GP contents $p(f(x_1), ..., f(x_N))$ is jointly Gaussian, where $\mu(x)$ is the mean and $\sum(x)$ is the covariance defined by $\sum_{ij} = k(x_i, x_j)$, and k is a positive kernel function [6].
Before continuing on GP theory, we first need to understand the multivariate Gaussian distribution.

3.2 Multivariate Gaussian distribution

In this section, we take our Gaussian multivariate theory from Chuong B Do [2].

A multivariate normal (or Gaussian) distribution is said to be a vector-value of a random variable

$$X = [X_1 ... X_n]^T$$

with mean $\mu \in \mathbf{R}^n$ and matrix

$$\Sigma \in \mathbf{S}^n_{++} = \{A \in \mathbf{R}^{n \times n} : A = A^T\}$$

and $x^T Ax \succ 0$ for all $x \in \mathbf{R}^n$ such that $x \neq 0$, as the covariance due to having the probability density function as

$$p(x; \mu, \Sigma) = \frac{1}{(2\mu)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) \tag{2}$$

We can also write this in short as $X \sim N(\mu, \Sigma)$.

This definition is an expansion from the density function of the univariate normal (or Gaussian) distribution write as

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \tag{3}$$

Where $-\frac{1}{2\sigma^2}(x-\mu)^2$ is the quadratic function of x as our variable with downwards parabola points and $\frac{1}{\sqrt{2\pi}\sigma}$ as an independent constant of variable x.

In Figure 1, the diagram of the left illustrates a univariate Gaussian density for the variable X , and the diagram of the right hand side illustrates the multivariate Gaussian density function over the X_1 and X_2 variables [2].

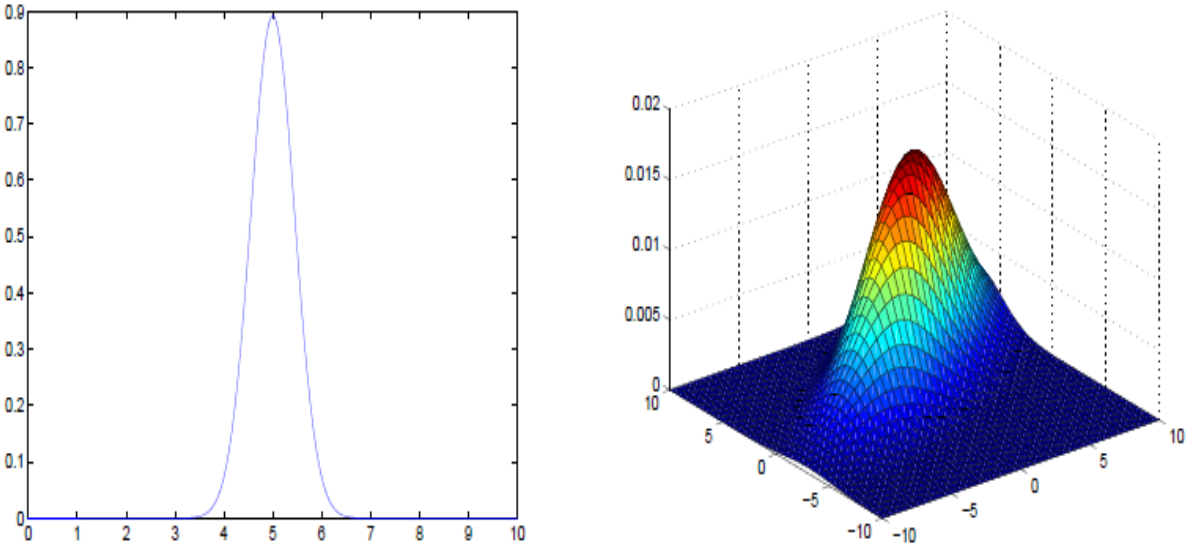


Figure 2: Univariate vs. multivariate Gaussian density functions [2].

3.3 GPs for regression

In this section, we look from Murphy [6] .

Let GP be a prior over the regression function:

$$f(x) \backsim GP(m(x), k(x, x'))$$

where the mean function is denoted by $m(x)$ and covariance or kernel function is denoted by $k(x, x')$. These functions can be computed as

$$m(x) = \mathbb{E}[f(x)]$$

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))^T]$$

and $k(x, x') \geq 0$. For a given finite set of points the process is said to be a joint Gaussian:

$$p(\mathbf{f}|\mathbf{X}) = N(\mathbf{f}|\mu, \mathbf{K}) \tag{4}$$

where $\mathbf{K}_{ij} = k(x_i, x_j)$ and $\mu = (m(x_1), ..., m(x_N))$ [6]. We will present this model again in the application section later.

3.4 Prediction with GP

For this section, we take from Christopher KI Williams and Carl Edward Rasmussen [10].

A stochastic process is a gathering of random variables $\{Y(x)|x \in X\}$ indexed by a set X. A GP is a

stochastic process that can be fully explained by

$$\mu(x) = E[Y(x)],$$

the mean function and

$$C(x, x') = E[(Y(x) - \mu(x))(Y(x') - \mu(x'))]$$

which is the covariance function. Any set finite set of points will give a joint multivariate Gaussian distribution [10].

Consider a GP that has $\mu(x) = 0$ for simplicity sake. The predictive distribution for x cases is obtained from $n+1$ dimensional joint Gaussian distribution for n training cases and one test cases. This process is illustrated in Figure 2 below where we find one training case point and one test case point. The general Gaussian predictive distribution with mean and variance is given as

$$\hat{y}(x) = k^T(x)K^{-1}t$$

$$\sigma^2_{\hat{y}}(x) = C(x, x) - k^T(x)K^{-1}k(x),$$

where

$$k(x) = \left(C(x, x^{(1)}), ..., C(x, x^{(n)})\right)^T,$$

K is defined as the covariance matrix for training cases $K_{ij} = C(x^{(i)}, x^{(j)})$, and $t = (t^{(1)}, ..., t^{(n)})^T$ [10] [8].

3.4.1 Predictions using noise-free observations

Lets assume we observe a training set as $D = \{(\mathbf{x}_i, f_i), i = 1 : N\}$, with $f_i = f(\mathbf{x}_i)$ as the noise-free observation of the function calculated at $\mathbf{x}_i$. We desire to predict the function outcomes $\mathbf{f}_*$ which will be our predicted function, with a given test set $\mathbf{X}_*$of the size $N_* \times D$.

In the case of noise-free observations, the GP must predict $f(\mathbf{x})$ for any $\mathbf{x}$ that it has already observed, with no uncertainty.

Now coming back to the prediction problem. By definition the joint Gaussian distribution has the following form:

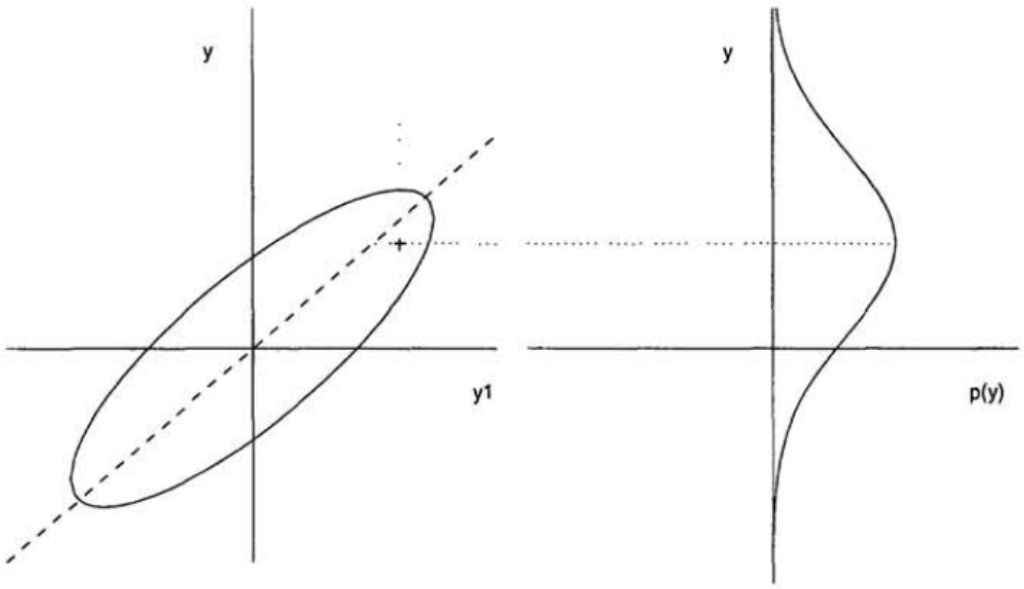


Figure 3: Prediction using GP for one training case point and one test case point. [10].

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{f}_* \end{pmatrix} \sim N\left(\begin{pmatrix} \mu \\ \mu_* \end{pmatrix}, \begin{pmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^{\mathbf{T}} & \mathbf{K}_{**} \end{pmatrix}\right) \tag{5}$$

where

$$\mathbf{K} = k(\mathbf{X}, \mathbf{X})$$

is $N \times N$,

$$\mathbf{K}_* = k(\mathbf{X}, \mathbf{X}_*)$$

is $N \times N_*$, and

$$\mathbf{K}_{**} = k(\mathbf{X}_*, \mathbf{X}_*)$$

is $N_* \times N_*$. With $\mathbf{f}_*$, $\mu_*$and $\mathbf{K}_*$ being our predictive parameters for the new data observed .

By the standard rule for conditioning Gaussian, the posterior has the form

$$p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{f}) = N(\mathbf{f}_*|\mu_*, \mathbf{\Sigma}_*) \tag{6}$$

where

$$\mu_* = \mu(\mathbf{X}_*) + \mathbf{K}_*^{\mathbf{T}}\mathbf{K}^{-\mathbf{1}}(\mathbf{f} - \mu(\mathbf{X}))$$

$$\Sigma_* = \mathbf{K}_{**} - \mathbf{K}_*^{\mathbf{T}}\mathbf{K}^{-\mathbf{1}}\mathbf{K}_*$$

As a example to illustrate this. Let say $x = (x_1, x_2)$ is a joint Gaussian with the parameters defined as:

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$$

,

$$\sum = \begin{pmatrix} \sum_{11} & \sum_{12} \\ \sum_{21} & \sum_{22} \end{pmatrix}$$

,

$$\Lambda = \sum^{-1} = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix}$$

.

And the marginals are presented:

$$p(x_1) = N(x_1|\mu_1, \sum_{11})$$

,

$$p(x_2) = N(x_2|\mu_2, \sum_{22})$$

The posterior conditional will be written as:

$$p(x_1|x_2) = N(x_1|\mu_{1|2}, \sum_{1|2})$$

$$\mu_{1|2} = \mu_1 + \sum_{12} \sum_{22}^{-1} (x_2 - \mu_2)$$

$$= \mu_1 - \Lambda_{11}^{-1} \Lambda_{12} (x_2\text{-}\mu_2)$$

$$= \sum_{1|2} (\Lambda_{11}\mu_1 - \Lambda_{12}(x_2 - \mu_2))$$

$$\sum_{1|2} = \sum_{11} - \sum_{12} \sum_{22}^{-1} \sum_{21} = \Lambda_{11}^{-1} \tag{7}$$

This gives us three different ways to show the posterior mean and two ways of expressing the posterior covariances, where each will be useful depending on specifications.

3.4.2 Predictions using noisy observations

Now we look at a case where noisy observations are found in the function, $y = f(\mathbf{x}) + \epsilon$ with $\epsilon \sim N(0, \sigma_y^2)$. A model like this does not have to interpolate the training data, although we must still get as close as possible to the observed data. We define the covariance of the noisy responses observed as

$$cov[y_p, y_q] = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_y^2 \delta_{pq}$$

where $\delta_{pq} = \mathbb{I}(p = q)$. This can also be simply written as

$$cov[\mathbf{y}|\mathbf{X}] = \mathbf{K} + \sigma_y^2 \mathbf{I}_N \triangleq \mathbf{K}_y$$

We assume that the noise value is independently added to individually observed values, which make the second matrix diagonal [6].

The joint density of data observed and the latent, noise-less function on the test points is given by

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim N\left(0, \begin{pmatrix} \mathbf{K}_y & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{pmatrix}\right) \tag{8}$$

Here we assume the mean of zero, to simplify our notation. Then have the posterior predictive density as

$$p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y}) = N(\mathbf{f}_*|\mu_*, \mathbf{\Sigma}_*) \tag{9}$$

where

$$\mu_* = \mathbf{K}_*^T \mathbf{K}_y^{-1} \mathbf{y}$$

$$\Sigma_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}_y^{-1} \mathbf{K}_*$$

if we have a single test input case, then it can simply be written as follows

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = N(f_*|\mathbf{k}_*^T \mathbf{K}_y^{-1} \mathbf{y}, k_{**} - \mathbf{k}_*^T \mathbf{K}_y^{-1} \mathbf{k}_*) \tag{10}$$

where

$$\mathbf{k}_* = [k(\mathbf{x}_*, \mathbf{x}_1), ..., k(\mathbf{x}_*, \mathbf{x}_N)]$$

and

$$k_{**} = k(\mathbf{x}_*, \mathbf{x}_*).$$

The posterior mean can be written as:

$$\overline{f}_* = \mathbf{k}_*^T \mathbf{K}_y^{-1} \mathbf{y} = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}_*)$$

where $\alpha = \mathbf{K}_y^{-1}\mathbf{y}$.

3.5 Covariance matrix

Assuming that we can measure similarities between objects is an approach that doesn't require pre-processing objects into fixed-size vector format. The measure of similarity of two objects $x, x' \in X$, can be defined as $k(x, x') \geq 0$, where X is an abstract space and k is a kernel function [6].

3.5.1 Kernel function

A covariance matrix (which can as be refer to as kernel function) is defined as a real-valued function of two objects written as $k(x, x') \in R$, for $x, x' \in X$. We earlier interpreted the function as a measure of similarity between objects $x, x' \in X$, this function is said to be symmetric meaning $k(x, x') = k(x', x)$ and non-negative therefore $k(x, x') \geq 0$.

The Squared Exponential Kernel (SE kernel) or also known as Gaussian kernel is written as

$$k(x, x') = exp\left(-\frac{1}{2}(x - x')^T \sum^{-1}(x - x')\right)$$

and we can rewrite the kernel if $\sum$ is diagonal as

$$k(x, x') = exp\left(-\frac{1}{2}\sum_{j=1}^{D}\frac{1}{\sigma_j^2}(x - x')^2\right)$$

whereσ_j is the characteristics length scale of j dimensions [6].

3.5.2 Covariance matrix (Kernel function) for building generative models

There is a special kind of kernel known as a smoothing kernel which if is very useful for creating non-parametric density estimates. We are going to use it to create generative model for regression by making a model of the format $p(y, x)$.

The smoothing kernel satisfies the following properties:

$$\int k(x)dx = 1, \int xk(x)dx = 0, \int x^2 k(x)dx \succ 0$$

A Gaussian kernel serves as a simple example,

$$k(x) \triangleq \frac{1}{(2\pi)^{\frac{1}{2}}} e^{-\frac{x^2}{2}}$$

The width of the kernel is controlled by introducing a parameter h (know as the bandwidth).

$$k_h(x) \triangleq \frac{1}{h} k(\frac{x}{h})$$

We can then generalize this function to a vector by defining an radial basis function or RBF kernel :

$$k_h(X) = k_h(\parallel X \parallel)$$

When take Gaussian kernel into consideration this will become

$$k_h(X) = \frac{1}{h^D (2\pi)^{D/2}} \prod_{j=1}^{D} exp\left(-\frac{1}{2h^2} x_j^2\right)$$

Now we have what we call a parametric density estimator for data in R^D. But we still need to specify the number K and μ_k as location of the clusters. We can also allocate one cluster center per data point to estimate μ_k, so that $\mu_i = x_i$. Therefore the model will become

$$p(x|D) = \frac{1}{N} \sum_{i=1}^{N} N(x|x_i, \sigma^2 I)$$

The we generalize to

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^{N} k_h(x - x_i)$$

This is called the kernel density estimator (KDE), and is considered to be a non-parametric density model. The advantage of this model over a parametric model is that we are not required to fit the model, but the disadvantage is that its takes a lot of memory to store and time when evaluating [6]. Figure 3 shows KDE in 1d for two kinds of kernels.

3.5.3 Effect of the kernel parameters

The predictive performance of GPs is exclusively dependent on a chosen kernel suitability. Let us now say we choose the SE kernel below for noisy observations

$$k_y(x_p, x_q) = \sigma_f^2 exp\left(-\frac{1}{2l^2}(x_p - x_q)^2\right) + \sigma_y^2 \sigma_{pq}$$

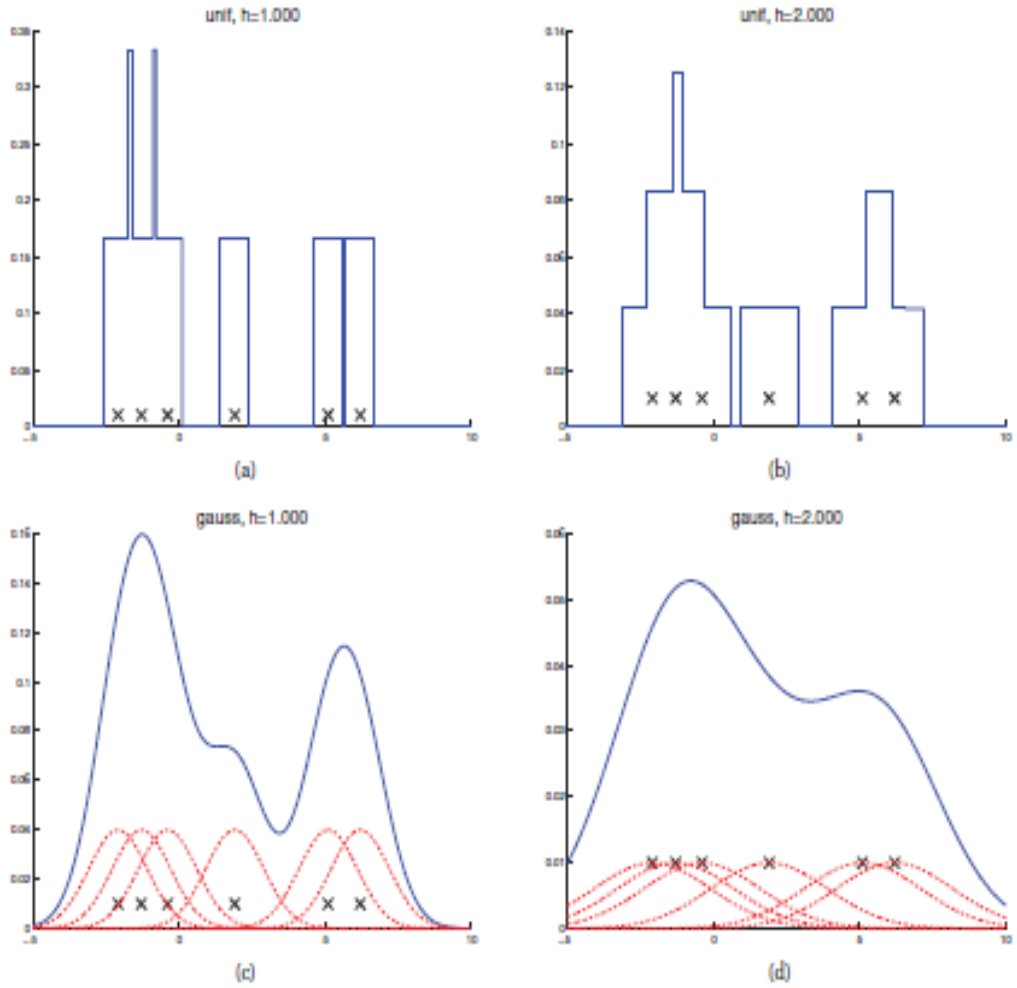


Figure 4: A non-parametric (kernel) density estimated from 6 data points, denoted by x. Top row: uniform kernel. Bottom row: Gaussian kernel [6].

where l is the horizontal scale over which the function changes, σ_f^2 controls the vertical scale of the function, and σ_f^2 is the noise variance.

Figure 4 shows the effects of changes made by these specific parameters. It shows 20 noisy data points sampled from SE kernel with $(l, \sigma_f, \sigma_y) = (1, 1, 0.1)$ for the first one, conditionally on the data it makes prediction various parameters. Again looking at figure 4, we apply the hyper-parameters represented as (l, σ_f, σ_y) are produces as follows: (a)(1,1,0.1) where we see a good fit. In (b)(0.3,1.08,0.0005) we decrease the distance making l=0.3, this makes the function more "wiggly" shaped [8]. This is because the uncertainty increases, as the is also a rapid increase in the effective distance from training data points. Lastly in (c)(3,1.16,0.89) we increase the length to a l=3.0 and the function looks more smoother [6].

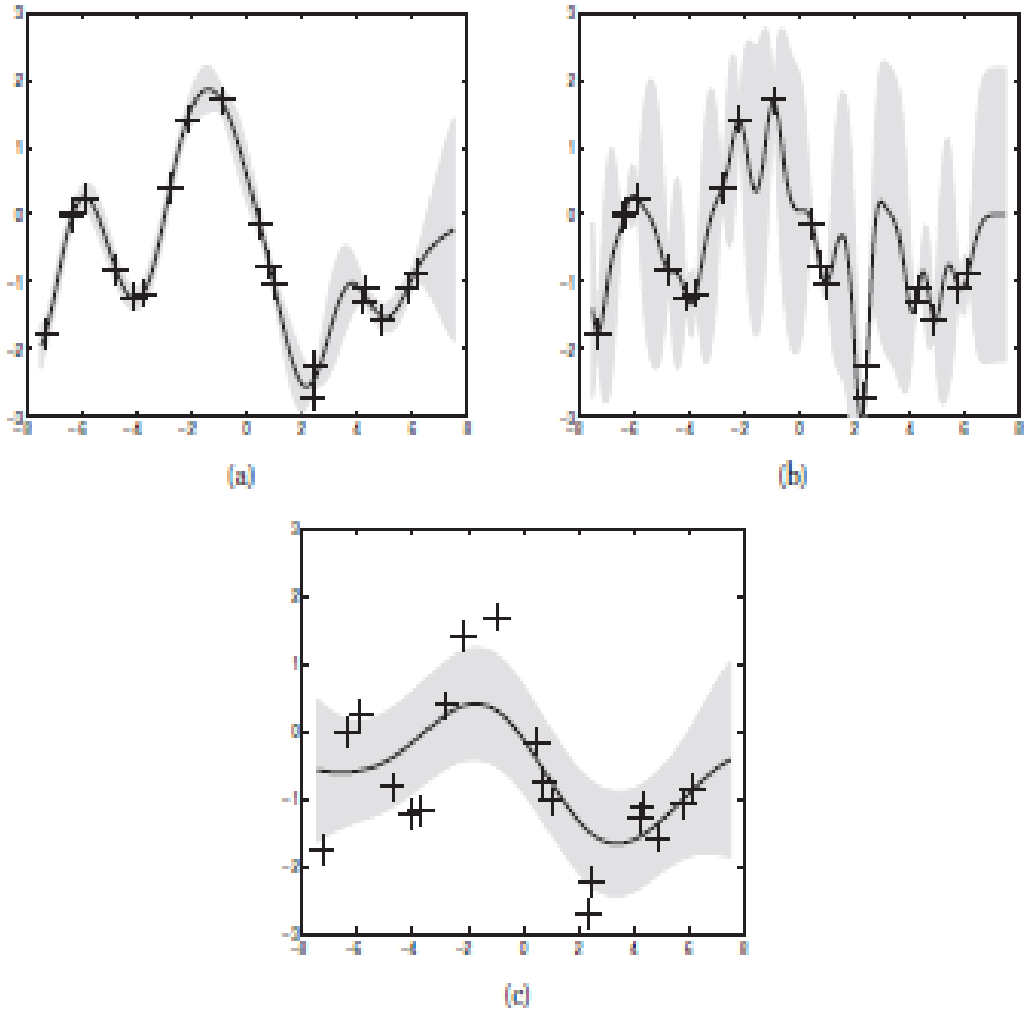


Figure 5: GPs with SE kernels. The hyper-parameters represented as(l, σ_f, σ_y) [6].

3.6 Computational and numerical issues with the GP regression

The function $\overline{f_*} = \mathbf{k}_*^T \mathbf{K}_y^{-1} \mathbf{y}$ gives us the predictive mean. But to directly invert $\mathbf{K}_y$ is not a wise idea, since it brings about numerical instability. We would then rather use a more robust option by computing a Cholesky decomposition , $\mathbf{K}_y = \mathbf{L}\mathbf{L}^T$ [6].

It then makes the predictive mean and log marginal likelihood less complex to compute using the pseudo-code in Algorithm 1 shown below.

Another option to follow is to solve $\mathbf{K}_y \alpha = \mathbf{y}$ as a linear system using conjugate gradients rather than Cholesky decomposition [6].

Algorithm 1 GP regression [6].

1 $\mathbf{L} = cholesky(\mathbf{K}+\sigma_y^2\mathbf{I})$;
2 $\alpha = \mathbf{L}^T \backslash (\mathbf{K} \backslash \mathbf{y})$;
3 $\mathbb{E}[f_*] = \mathbf{k}_*^T \alpha$;
4 $\mathbf{v} = \mathbf{L} \backslash \mathbf{k}_*$;
5 $var[f_*] = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^T\mathbf{v}$;
6 $logp(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^T\alpha - \sum_i logL_{ii} - \frac{N}{2}log(2\pi)$

3.7 Illustration of GP

The objective is to fit parameters a continuous dataset. We consider a domain of say values between -5 and 5. Before we have seen any data we need a prior. The prior must ensure that values are close together in input space will produce values that are close together in output space. We going make use of a covariance matrix.

In this section we compute a simple single-dimensional Gaussian processes regression for the noise-less and noisy cases. We estimate the kernel parameters using the maximum likelihood principles. We will use 95 % confidence interval.

We are simulating data from the kernels to run GP regression model that was introduced in our theory. Further using a simple xy data for applying the local polynomial regression for simplicity in our reasoning or comparisons.

We are going to run a simulated GP regression model for sci-kit learn for three samples checking difference between the prior and posterior.

Before going into the detail about the models being studied, under this illlustration we are going to run a function simulated by python in order to explain properties of the Gaussian process. By showing the noisy case and noise-less case.

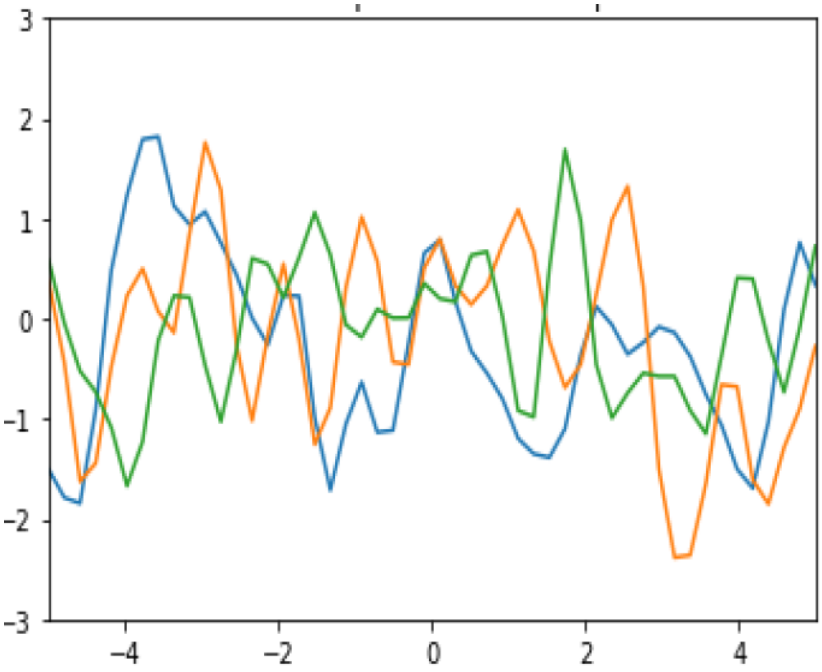


Figure 6: Three samples simulated from the GP prior

In Figure 6 we have a prior, $p(\mathbf{f}|\mathbf{X})$, where we use SE kernel also known as Gaussian kernel or RBF kernel in 1-dimension, and it is written as

$$k(x, x') = \sigma_f^2 exp\left(-\frac{1}{2l^2}(x - x')^2\right)$$

where l controls the horizontal variation length scale over the function, σ_f^2 controls the vertical variations.

In Figure 7 we observe samples from the posterior, $p(\mathbf{f}|\mathbf{X}_*, \mathbf{X}, \mathbf{f})$. The model perfectly interpolates the training data, and the uncertainty of our predictions increases as we gain distance away from our observed data.

The application of the noise-free GP regression can be found in weather forecasting programs because it often offers a computationally cheaper proxy for the behavior of a complex simulator [6].

There are two different ways of computing a simple 1-d regression, one is the noise-free case illustrated in Figure 8 and the other is the noisy case illustrated in Figure 9. For both a maximum likelihood principle is used in estimating the kernel parameters. In the Figures 8 and 9 we see interpolated properties of the Gaussian process model [7][7].

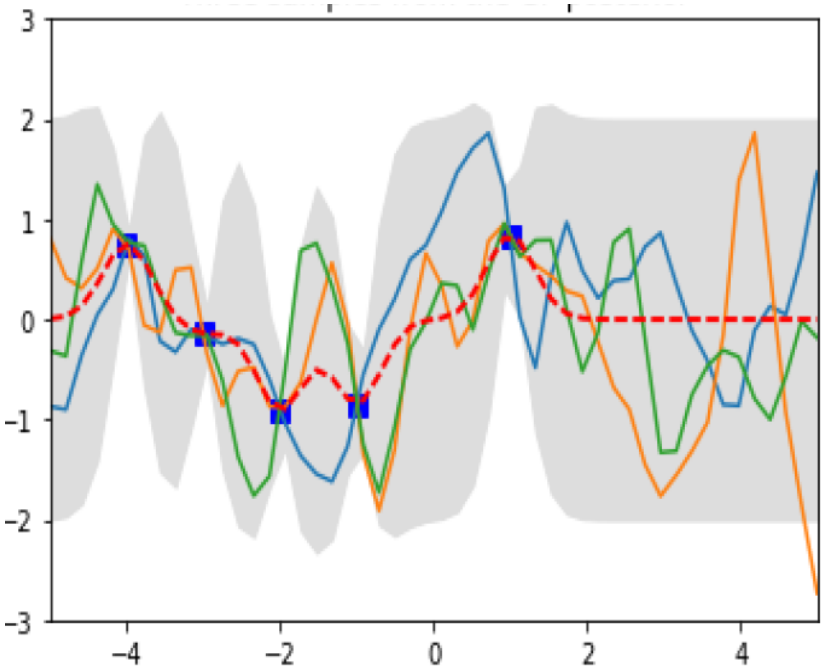


Figure 7: Three samples simulated from the GP posterior

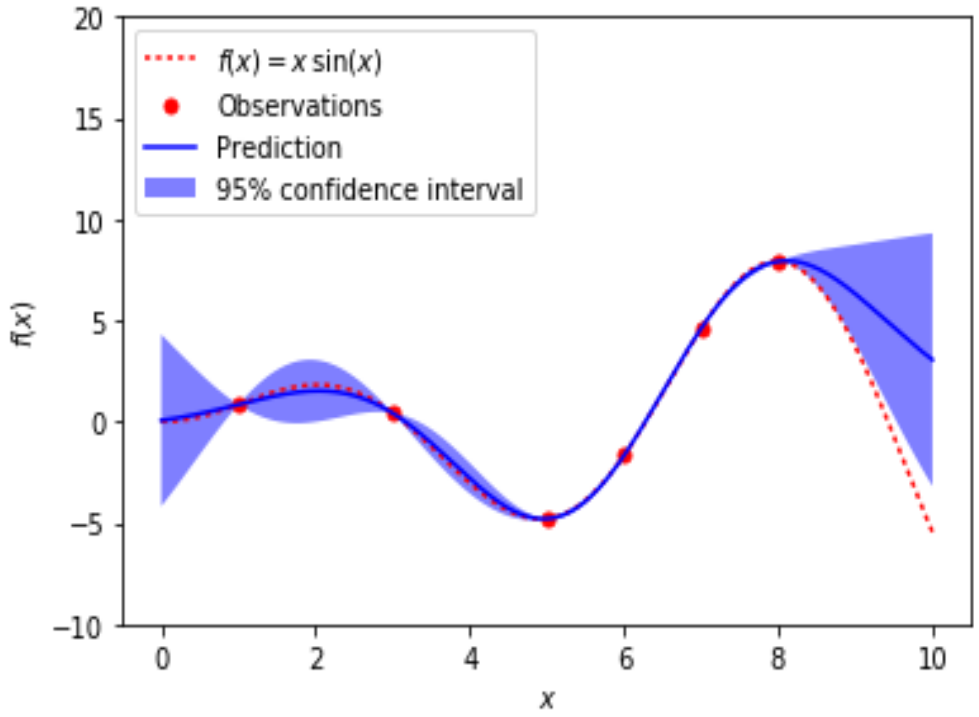


Figure 8: The noiseless case.

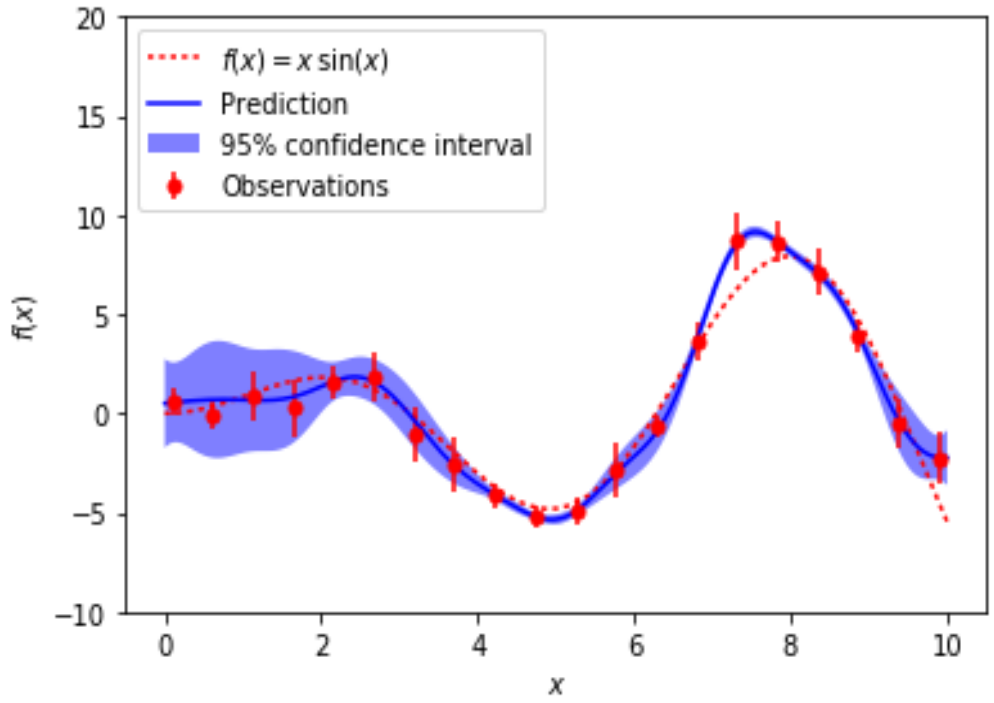


Figure 9: the noisy case

4 Non-parametric regression

In this section we introduce non-parametric regression. The reason for this being that we will compare the results of the GPs with the non-parametric regression.

4.1 Objective

The Nonparametric regression looks at how dependent the response variable (y) is to more than one predictor variable(s) (x) without defining in a function that relates the response to the predictors. The objective of the nonparametric regression is to then estimate the function that fits on the data directly without the need of estimating specified parameters which are estimated through the standard OLS estimation [4].

4.2 The underlying statistical model(s)

The underlying statistical model fits a simple linear regression e.g. $y = f(x) + e$ with an unspecified $f()$ function. Nonparametric regression aims to estimate that $f()$ function is a continuous and smooth function. The error term, e, has an independent normal distribution with zero mean and variance is $\sigma 2$. Using that regression line you estimate what your dependent variable should be at that focal point. One

of the assumptions made is that the dependent variables (y) are indeed independent from each other [4].

4.3 Two methods of nonparametric regression

4.3.1 Kernel estimation

Kernel estimation fits values by using locally weighted averaging, with the aid of some weight function where more weight is placed on data closer to the focal point, x_0. Doing this for several points you obtain a series of $f(x_0)'s$. The further away from the focal point, the less weight placed on that data point. The estimation is achieved by creating a neighborhood around the focal point and weighting using that window [4].

4.3.2 Local polynomial regression

Local polynomial regression is like kernel estimation. Instead of $f(x_0)$ being estimated by locally weighted averaging it will now be estimated by locally weighted regression. The main difference is that it uses locally weighted regression by minimizing the weighted sum of squares. It tends to be less bias that kernel estimation [5].

5 Application

5.1 Data

Table 1, from the appendix is the created xy simple data that is being used for our paper with the purpose of comparing the two models. We are going to firstly fit the local polynomial regression to the data using iml program in SAS, there after fit the Gaussian process regressor on the data using python as our programming language is it works best with Gaussian application.

5.2 Model

The focus model is the Gaussian regressor for the dataset xy which will use python to fit the Gaussian process regression to it and see how well of a fits it is to the data. That will give us an idea of also how well the Gaussian can be used to estimating.

We will then also use a method for the nonparametric regression using called the local polynomial regression to fit the same data and see how well it will fit the data or describe it.
Both these models focus on the nonparametric methods.

5.3 Results

Figure 10 is a plot of the data in Table 1 from the appendix , this simple data is plotted using SAS.

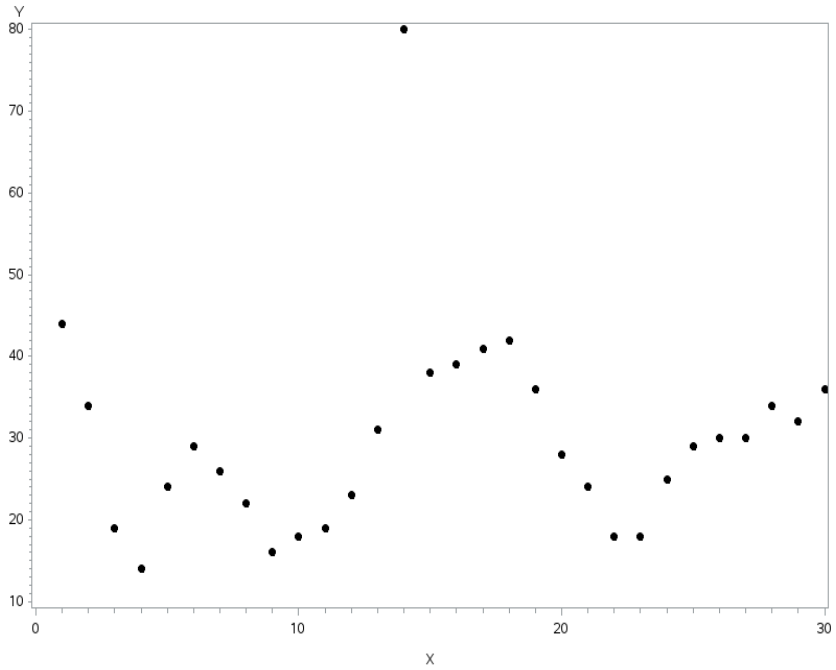


Figure 10: plot of xy in SAS

Figure 11, show the fitted local polynomial regression to the data plotted in Figure 10 using SAS [5]. We observed a very smooth fitting but not well fitted to every observation in the data.

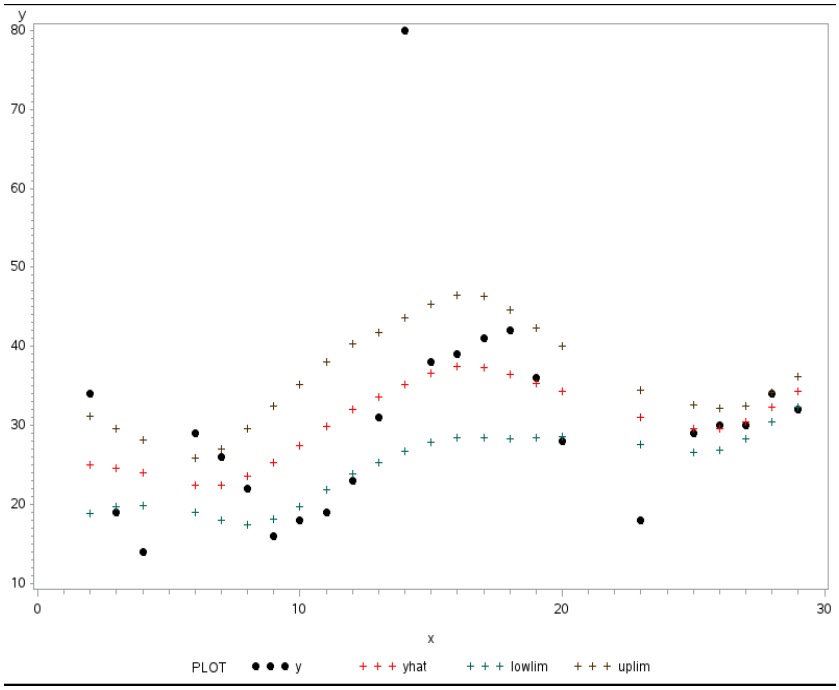


Figure 11: Local polynomial fitting in SAS [4].

Again in Figure 11, model does fit to the most of the observations and sees to able to well estimate the observation but with wider confidence intervals.

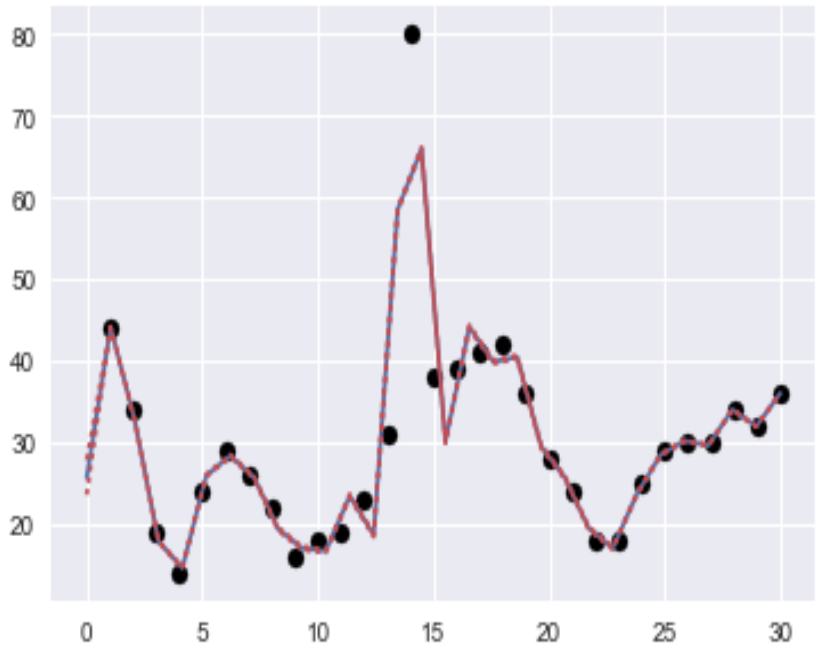


Figure 12: Gaussian regression fitting in Python [1].

In Figure 12, we fitted the Gaussian process regression to the same plot in Figure 10 using Python

[7]. This figure shows a perfect fit to every observation in the data, and has accurate estimates. This shows that the Gaussian fitted the data well and it out performs the local polynomial.

The results show that the sse(error) of the Gaussian regression is 2294.20176077 compare to the local polynomial regression which is 56952.1742 confirms that the Gaussian regression is a better estimator.

6 Conclusion

In studying the models we have found them to be complex to apply and demanding more time in understanding the programming skills required to transfer the theory into pseudo-codes. The Gaussian process for regression shows to be a better fit to than the local polynomial regression to the xy data set. This might insinuate the Gaussian being a better estimator than the local polynomial but we also observe a smaller sse in the Gaussian regression as compare to the local polynomial regression this confirms our believe of the Gaussian being a better estimator.

In as much as the Gaussian process shows a better fit, further studies could be looked into when comparing these two models. There seems to be a need to expand the scope of comparing these models using more than just confidence intervals and estimates.This creates the need to introduce r-squares and sum of squares in to picture even though there will be difficulties in being found in comparing the two using two different programming languages, that also brings the need to study how to apply the Gaussian process regression in SAS. This complexity of comparing could be taken further by using different datasets and seeing how each model responses.

The study has been valuable in understanding different ways of of solving nonparametric regression problems, it has opened our minds in thinking beyond tradition ways of solving regression problems. Hence, further studies on this topic will be of great value in investing time to finding better and convenient ways of solving complex regression problems.

References

[1] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[2] Chuong B Do. The multivariate Gaussian distribution. *Section Notes, Lecture on Machine Learning, CS*, 229, 2008.

[3] Mark Ebden. Gaussian processes: A quick introduction. *arXiv preprint arXiv:1505.02965*, 2015.

[4] John Fox. Introduction to nonparametric regression. *McMaster University, Canada*, 2005.

[5] John Hughes. Local polynomial regression. 2013.

[6] Kevin P Murphy. *Machine Learning: a Probabilistic Perspective*. MIT press, 2012.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian Processes for Machine Learning*, volume 1. MIT press Cambridge, 2006.

[9] Stephen Roberts, M Osborne, M Ebden, Steven Reece, N Gibson, and S Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A*, 371(1984):20110550, 2013.

[10] Christopher KI Williams and Carl Edward Rasmussen. Gaussian processes for regression. *Advances in Neural Information Processing Systems*, pages 514–520, 1996.

Appendix

Data used in the paper .

x	y
1	44
2	34
3	19
4	14
5	24
6	29
7	26
8	22
9	16
10	18
11	19
12	23
13	31
14	31
15	38
16	39
17	41
18	42
19	36
20	28
21	24
22	18
23	80
24	25
25	29
26	30
27	30
28	34
29	32
30	36

Table 1: Manually created xy dataset

Python code for application Figure 6.

```
import numpy as np
import matplotlib.pyplot as pl

# Test data
n = 50
Xtest = np.linspace(-5, 5, n).reshape(-1,1)

# Define the kernel function
def kernel(a, b, param):
    sqdist = np.sum(a**2,1).reshape(-1,1) + np.sum(b**2,1) - 2*np.dot(a, b.T)
    return np.exp(-.5 * (1/param) * sqdist)

param = 0.1
K_ss = kernel(Xtest, Xtest, param)

# Get cholesky decomposition (square root) of the
# covariance matrix
L = np.linalg.cholesky(K_ss + 1e-15*np.eye(n))
# Sample 3 sets of standard normals for our test points,
# multiply them by the square root of the covariance matrix
f_prior = np.dot(L, np.random.normal(size=(n,3)))

# Now let's plot the 3 sampled functions.
pl.plot(Xtest, f_prior)
pl.axis([-5, 5, -3, 3])
pl.title('Three samples from the GP prior')
pl.show()
```

Python code for application Figure 7.

```
# Noiseless training data
Xtrain = np.array([-4, -3, -2, -1, 1]).reshape(5,1)
ytrain = np.sin(Xtrain)

# Apply the kernel function to our training points
K = kernel(Xtrain, Xtrain, param)
L = np.linalg.cholesky(K + 0.00005*np.eye(len(Xtrain)))

# Compute the mean at our test points.
K_s = kernel(Xtrain, Xtest, param)
Lk = np.linalg.solve(L, K_s)
mu = np.dot(Lk.T, np.linalg.solve(L, ytrain)).reshape((n,))

# Compute the standard deviation so ve can plot it
s2 = np.diag(K_ss) - np.sum(Lk**2, axis=0)
stdv = np.sqrt(s2)
# Draw samples from the posterior at our test points.
L = np.linalg.cholesky(K_ss + 1e-6*np.eye(n) - np.dot(Lk.T, Lk))
f_post = mu.reshape(-1,1) + np.dot(L, np.random.normal(size=(n,3)))

pl.plot(Xtrain, ytrain, 'bs', ms=8)
pl.plot(Xtest, f_post)
pl.gca().fill_between(Xtest.flat, mu-2*stdv, mu+2*stdv, color="#dddddd")
pl.plot(Xtest, mu, 'r--', lw=2)
pl.axis([-5, 5, -3, 3])
pl.title('Three samples from the GP posterior')
pl.show()
```

Python code for Figures 8.

```
import numpy as np
from matplotlib import pyplot as plt

from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF, ConstantKernel as C

np.random.seed(1)


def f(x):
    """The function to predict."""
    return x * np.sin(x)

# ----------------------------------------------------------------------
#
#  First the noiseless case
X = np.atleast_2d([1., 3., 5., 6., 7., 8.]).T

# Observations
y = f(X).ravel()

# Mesh the input space for evaluations of the real function, the prediction and
# its MSE
x = np.atleast_2d(np.linspace(0, 10, 1000)).T

# Instanciate a Gaussian Process model
kernel = C(1.0, (1e-3, 1e3)) * RBF(10, (1e-2, 1e2))
gp = GaussianProcessRegressor(kernel=kernel, n_restarts_optimizer=9)

# Fit to data using Maximum Likelihood Estimation of the parameters
gp.fit(X, y)

# Make the prediction on the meshed x-axis (ask for MSE as well)
y_pred, sigma = gp.predict(x, return_std=True)

# Plot the function, the prediction and the 95% confidence interval based on
# the MSE
fig = plt.figure()
plt.plot(x, f(x), 'r:', label=u'$f(x) = x\,\sin(x)$')
plt.plot(X, y, 'r.', markersize=10, label=u'Observations')
plt.plot(x, y_pred, 'b-', label=u'Prediction')
plt.fill(np.concatenate([x, x[::-1]]),
         np.concatenate([y_pred - 1.9600 * sigma,
                         (y_pred + 1.9600 * sigma)[::-1]]),
         alpha=.5, fc='b', ec='None', label='95% confidence interval')
plt.xlabel('$x$')
plt.ylabel('$f(x)$')
plt.ylim(-10, 20)
plt.legend(loc='upper left')
```

Python code for Figure 9.

```
# ----------------------------------------------------------------------
# now the noisy case
X = np.linspace(0.1, 9.9, 20)
X = np.atleast_2d(X).T

# Observations and noise
y = f(X).ravel()
dy = 0.5 + 1.0 * np.random.random(y.shape)
noise = np.random.normal(0, dy)
y += noise

# Instanciate a Gaussian Process model
gp = GaussianProcessRegressor(kernel=kernel, alpha=(dy / y) ** 2,
                              n_restarts_optimizer=10)

# Fit to data using Maximum Likelihood Estimation of the parameters
gp.fit(X, y)

# Make the prediction on the meshed x-axis (ask for MSE as well)
y_pred, sigma = gp.predict(x, return_std=True)
```

```
# Plot the function, the prediction and the 95% confidence interval based on
# the MSE
fig = plt.figure()
plt.plot(x, f(x), 'r:', label=u'$f(x) = x\,\sin(x)$')
plt.errorbar(X.ravel(), y, dy, fmt='r.', markersize=10, label=u'Observations')
plt.plot(x, y_pred, 'b-', label=u'Prediction')
plt.fill(np.concatenate([x, x[::-1]]),
         np.concatenate([y_pred - 1.9600 * sigma,
                         (y_pred + 1.9600 * sigma)[::-1]]),
         alpha=.5, fc='b', ec='None', label='95% confidence interval')
plt.xlabel('$x$')
plt.ylabel('$f(x)$')
plt.ylim(-10, 20)
plt.legend(loc='upper left')

plt.show()
```

SAS code and data import for Figures 10.

```
data a;
input X Y;
cards;
1    44
2    34
3    19
4    14
5    24
6    29
7    26
8    22
9    16
10   18
11   19
12   23
13   31
14   80
15   38
16   39
17   41
18   42
19   36
20   28
21   24
22   18
23   18
24   25
25   29
26   30
27   30
28   34
29   32
30   36
;
run;
proc gplot data=a;
plot y*x;
run;
```

SAS code continuing below for Figure 11.

```
proc iml;
 use a;
 read all into xy;
 n=nrow(xy);
 span=0.7;
 n1=0.8*n;
 m=round(span*n1); /*number of poins to include in the window*/

 randsam=j(n,1,1);
 randgen=rannor(randsam)||xy;
 call sort (randgen, {1});
 xy1=randgen[1:n1,2:3];
 rows_xy1=nrow(xy1);
 *nxy=nrow(xy1);
 *print xy1;

 minxy1=min(xy1[,1]);
 maxxy1=max(xy1[,1]);
 **print minxy1 maxxy1;

 /*kernel*/
 do x_focal=1 to rows_xy1;
 xc=xy1[,1]-xy1[x_focal,1];
 xd=abs(xc);
 mhood=xy1||xc||xd; /*neighborhood*/
 call sort(mhood,{4}); /*sort according to abs differences*/
 mhood1=mhood[1:m,];
 h=0.5*(max(mhood1[,1])-min(mhood1[,1])); /*half the length of the window*/

 z=mhood1[,4]/h; /*equation for abs(z) in the tricube weight*/
 wt=((1-(z)##3)##3)#(z<1); /*condition matrix filled with 1's and 0's*//*'##'=element power*/
 **print wt; /*check point*/
```

```
**print wt; /*checkpoint*/
y_hat=(wt`*mhood1[,2])/(j(1,m,1)*wt); /*calculation of weighted averages of the y values*/
**ker =ker//(x_focal||y0_hat);

**print y0_hat;

/**LPR with df=m-2**/
w=diag(wt);
X=J(nrow(mhood1),1,1)||mhood1[,3];
Y=mhood1[,2];
bhat_lpr=inv(x`*w*x)*x`*w*y;
bhat=bhat//bhat_lpr[1];

yhat=x*bhat_lpr;
**print yhat;
sse=(y-yhat)`*w*(y-yhat);
mse=sse/(m-2);
varb=mse*inv(x`*w*x);

k=tinv(0.975, m-2);
lowlim=lowlim//(bhat_lpr[1]-(k*(sqrt(varb[1,1]))));
uplim=uplim//(bhat_lpr[1]+(k*(sqrt(varb[1,1]))));
end;
lprmtrx=xy1||bhat||lowlim||uplim;

print lowlim uplim;
create assnglpr4 from lprmtrx[colname={'x' 'y' 'yhat' 'lowlim' 'uplim'}];
append from lprmtrx;
```

```
/*create assngkern4 from ker[colname={'x_focal' 'y_hat'}];*/
quit;


goptions reset = all;
symbol1 v=dot c=black h=1;
symbol2 c=red h=1;

/*proc gplot data=assngkern4;
plot y_hat*x_focal/overlay legend;
run;*/

proc gplot data=assnglpr4;
plot (y yhat lowlim uplim)*x/overlay legend /*vaxis=2 haxis=2*/;
run;
proc sgplot data=assnglpr4 ;
band x=x lower=lcl upper=ucl / transparency=.5 legendlabel="Confidence Band" fillattrs=(color=purple);
series x=x y=y / lineattrs=(color=blue);
series x=x y=yhat / lineattrs=(color=red);
series x=x y=lowlim / lineattrs=(color=green);
series x=x y=uplim / lineattrs=(color=orange);

Title "LPR" ;
run ;
```

Python code for Figure 12.

```
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
sns.set(color_codes=True)

%matplotlib inline
```

```
class GP(object):

    @classmethod
    def kernel_bell_shape(cls, x, y, delta=1.0):
        return np.exp(-1/2.0 * np.power(x - y, 2) / delta)

    @classmethod
    def kernel_laplacian(cls, x, y, delta=1):
        return np.exp(-1/2.0 * np.abs(x - y) / delta)

    @classmethod
    def generate_kernel(cls, kernel, delta=1):
        def wrapper(*args, **kwargs):
            kwargs.update({"delta": delta})
            return kernel(*args, **kwargs)
        return wrapper
```

```
 def __init__(self, x, y, cov_f=None, R=0):
     super().__init__()
     self.x = x
     self.y = y
     self.N = len(self.x)
     self.R = R

     self.sigma = []
     self.mean = []
     self.cov_f = cov_f if cov_f else self.kernel_bell_shape
     self.setup_sigma()

 @classmethod
 def calculate_sigma(cls, x, cov_f, R=0):
     N = len(x)
     sigma = np.ones((N, N))
     for i in range(N):
         for j in range(i+1, N):
             cov = cov_f(x[i], x[j])
             sigma[i][j] = cov
             sigma[j][i] = cov

     sigma = sigma + R * np.eye(N)
     return sigma
```

```
def setup_sigma(self):
    self.sigma = self.calculate_sigma(self.x, self.cov_f, self.R)

def predict(self, x):
    cov = 1 + self.R * self.cov_f(x, x)
    sigma_1_2 = np.zeros((self.N, 1))
    for i in range(self.N):
        sigma_1_2[i] = self.cov_f(self.x[i], x)

    # SIGMA_1_2 * SIGMA_1_1.I * (Y.T -M)
    # M IS ZERO
    m_expt = (sigma_1_2.T * np.mat(self.sigma).I) * np.mat(self.y).T
    # sigma_expt = cov - (sigma_1_2.T * np.mat(self.sigma).I) * sigma_1_2
    sigma_expt = cov + self.R - (sigma_1_2.T * np.mat(self.sigma).I) * sigma_1_2
    return m_expt, sigma_expt

@staticmethod
def get_probability(sigma, y, R):
    multiplier = np.power(np.linalg.det(2 * np.pi * sigma), -0.5)
    return multiplier * np.exp(
        (-0.5) * (np.mat(y) * np.dot(np.mat(sigma).I, y).T))
```

```
def optimize(self, R_list, B_list):

    def cov_f_proxy(delta, f):
        def wrapper(*args, **kwargs):
            kwargs.update({"delta": delta})
            return f(*args, **kwargs)
        return wrapper

    best = (0, 0, 0)
    history = []
    for r in R_list:
        best_beta = (0, 0)
        for b in B_list:
            sigma = gaus.calculate_sigma(self.x, cov_f_proxy(b, self.cov_f), r)
            marginal = b* float(self.get_probability(sigma, self.y, r))
            if marginal > best_beta[0]:
                best_beta = (marginal, b)
        history.append((best_beta[0], r, best_beta[1]))
    return sorted(history)[-1], np.mat(history)
```

```
x = np.array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29
y = np.array([ 44, 34, 19, 14, 24, 29, 26, 22, 16, 18, 19, 23, 31, 80, 38, 39, 41, 42, 36, 28, 24, 18, 18, 25, 29, 30, 3
gaus = GP(x, y)

x_guess = np.linspace(0, 35,400)
y_pred = np.vectorize(gaus.predict)(x_guess)

plt.scatter(x, y, c="black")
plt.plot(x_guess, y_pred[0], c="b")
plt.plot(x_guess, y_pred[0] - np.sqrt(y_pred[1]) * 3, "r:")
plt.plot(x_guess, y_pred[0] + np.sqrt(y_pred[1]) * 3, "r:")
```

Sensitivity and specificity in evaluating the accuracy of screening tests

Tinashe Victoria Mhazo 11253909

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr. E.M. Louw

Department of Statistics, University of Pretoria

30 October 2017 (final)

Abstract

This research report will provide a theoretical overview of procedures used to compare a screening test against a 'gold standard' test. The ability of a screening test to discriminate between individuals with or without disease is described in terms of sensitivity, specificity, positive predictive values and negative predictive values.

Sensitivity and specificity are the core measures of the accuracy of a screening test but unfortunately are not useful when calculating the probability of a disease in an individual. It is the predictive values (positive and negative) that are used to estimate the probability of a disease in an individual. Likelihood ratios, which combine sensitivity and specificity, are used to give a synopsis of how many more times likely or less prone individuals with a disease are to have a certain test result than individuals that do not have the disease.

The optimal decision for the presence or absence of disease is determined by the chosen cut-off point for sensitivity and (1 - specificity). Different cut-off points will yield a receiver operating curve (ROC) that is used to select the optimal cut-off values to assess the screening accuracy of a test.

It will be shown how to calculate and interpret the above measures by using a contingency table with two rows and two columns (2×2 contingency table). A practical example using a SAS software procedure will be used as an illustration.

Declaration

I, *Tinashe Victoria Mhazo,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

Tinashe Victoria Mhazo

Dr. Elizabeth Magrietha Louw

Date

Acknowledgments

I would like to thank my supervisor Dr. Elizabeth Magrietha Louw for her never give up spirit, for her attention to detail and great suggestions that made this research report feasible. I am grateful for her time and guidance that I will cherish for years to come.

I would also like to thank my mother, I really appreciate the hours that she spent thinking about the questions I asked and even reading some of my drafts.

And finally to all my family and friends who were my support system.

Contents

List of Figures

List of Tables

1 Introduction

In the medical field, diagnostic or screening tests are constantly being evaluated against a well established gold standard test. The most used statistical measures to describe the validity of these tests are sensitivity and specificity [13]. The gold standard is a test that is currently preferred for diagnosing a particular disease. All the other methods of diagnosing a disease, including any new screening tests, are compared to the gold standard test [15]. In this research report, the evaluation of screening test measures will be investigated and illustrated by means of a practical example where SAS software procedures will be used to perform calculations of these measures.

A gold standard test is usually expensive to perform, thus, clinicians normally use a screening test as a substitute of the gold standard. This screening test would have been evaluated and tested for validity. Hence the purpose of this research report is to effectively show the steps required to validate a screening test. Screening tests provide various kinds of information, for example, medical tests (e.g X-rays), medical signs (e.g high blood pressure) and or medical symptoms (e.g weight loss). The accuracy of screening tests is essential in medical care since a doctor's decision of medical treatment relies on the results of the screening test [16]. The core statistical measures that are used to evaluate a screening test are known as sensitivity and specificity. The numerical measures: positive and negative predictive values, likelihood ratios and accuracy are all dependent on the sensitivity and specificity of a screening test [12].

Studies of screening test accuracy often report sensitivity and specificity simultaneously [9]. One way of simultaneously analyzing sensitivity and specificity is by using receiver operator characteristics curves (ROC). The relationship between sensitivity and specificity can be graphically represented by ROC curves which are derived from plotting false positives against true positives for all cut-off values [4]. The ROC curve assists in choosing the ideal model through deciding the best limit for the screening test [16]. This method uses the diagnostic odds ratio as the main outcome measure. Although the ROC method removes the effect of a possible threshold, it loses relevant clinical information about the test performance [14].

The research report will concentrate on the concepts of sensitivity, specificity, predictive values and accuracy of a screening test in the context of disease diagnosis. Firstly, definitions of basic concepts will be given, followed by a theoretical overview of the research topic. An outlay of the contingency table will be provided, followed by equations on how to calculate screening test measures, likelihood ratios, ROC analysis and associated 95% confidence intervals. Lastly, a practical example of disease screening and related SAS software procedures will be discussed with the calculations and interpretations of the above mentioned statistical measures.

2 Background Theory

2.1 Theoretical Overview

2.1.1 Gold standard and screening test

Gold standard This is a well-established diagnostic test that is assumed to be able to predict the true disease state of an individual, regardless of other diagnostic tests used [16].

Screening test A screening test is a laboratory or medical test that aims to detect a disease in its earliest and most treatable phases [5]. Through out the research report, the terms diagnostic test and screening test will be used interchangeably. The validity of a screening test is based on its accuracy in determining whether an individual is diseased or not. This, however, can only be determined if the accuracy of the screening test is compared to a gold standard test [16].

2.1.2 Terms used to define screening test measures

False negative False negative refers to when the individual has the disease but the screening test is negative [9]. False negative will be abbreviated as FN.

False positive False positive refers to when an individual does not have the disease but the screening test is positive [9]. False positive will be abbreviated as FP.

True negative True negative refers to when the individual is not infected by the disease and the absence of the disease is also reflected by the screening test [9]. The abbreviation TN will be used for true negative.

True positive True positive refers to when the individual has the disease and the screening test is positive [9]. The abbreviation TP will be used for to true positive value.

2.1.3 Outlay of a general 2 x 2 contingency table

When evaluating a screening test, a 2 x 2 contingency table lists the current disease status, as determined by the gold standard test, in the columns and the observed screening test results in the rows [16]. It is advised to always put the 'GOLD STANDARD' test as the column variable at the top of the contingency table, and the SCREENING test as the row variable on the left-hand side of the table. The category 'positive test' should be in the first row with the category 'negative test' in the second row. The category 'person with disease' should be in the first column, with the category 'person without disease' in the

second column [16].

		GOLD STANDARD		
		Person with disease	**Person without disease**	**Total**
SCREENING TEST	**Positive**	TP	FP	TP + FP
	Negative	FN	TN	TN + FN
	Total	TP + FN	TN + FP	TP + TN +FP + FN

Table 1: Contingency table with abbreviations used to define screening test measures

The total of the entries in the first column (TP + FN) are all the people with the disease while the total entries in the second column (TN+FP) are all the disease-free individuals[16]. According to the screening test, TP + FP (total entries in the first row) are the individuals who tested positive and TN+FN (total of the entries in the second row) are the individuals who tested negative on the disease [16].

According to [16], both true positive and true negative show that the outcome given by the screening test is consistent with the gold standard test. False positive and false negative suggest that the test results are contradicting the gold standard test.

Henceforth, the ideas discussed are summarized in Table 1. The concepts of sensitivity and specificity will be discussed first since these are important measures that form the basis of the accuracy of a screening test.

2.1.4 Sensitivity and specificity

Sensitivity

Sensitivity refers to the ability of a test to correctly identify an individual as diseased [15]. Sensitivity is the probability that a test will yield a positive result amongst people that are already diseased. When calculating sensitivity it is only the number of all positive assessments that are of interest [10]. Thus, sensitivity does not give any information about whether or not some individuals without the disease would have a positive result [1]. A formula for sensitivity is expressed as

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \tag{1}$$

A screening test is less likely to return false negative results if the numerical value of sensitivity is high, thus a test with a high sensitivity is usually desirable for screening test purposes. A screening test with high sensitivity is useful when the test result is negative. A sensitivity of 100% for a screening test means that the test will capture all possible positive assessments without missing any [16], but a person

who tests positive might or might not have the disease. However, it is highly likely that a person who tests negative indeed does not have the disease [1].

Specificity

Specificity refers to the ability of a test to correctly identify an individual as disease-free [15]. This is the percentage of the healthy individuals who test negative [10].This measure suggests how good a test is at correctly identifying a negative disease status [16]. Since specificity is only be calculated from the individuals that have been identified to have a negative assessment by the gold standard test, it is only the number of negative assessments that are of interest . It is then clear that specificity does not give any information about whether or not the individuals with the disease would also test negative, and if they do, what their proportion would be [1]. Following is a formula for specificity:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \tag{2}$$

A screening test is less likely to return false positive results if the numerical values of specificity are high [16]. Thus, a screening test with high specificity is desirable. A screening test with high specificity is useful when the test result is positive. A test with a specificity of 100% means that many individuals without the disease test negative, but it does not necessarily mean that an individual who tests negative does not have the disease. However, it is highly likely that an individual, who returns a positive result, has the disease [1].

Usefulness of sensitivity and specificity

A test can be very sensitive without being specific and vice versa, but a test that has both high sensitivity and high specificity is preferred [16]. As seen above, a high sensitivity is useful since one can almost surely conclude that an individual, who tests negative, indeed is disease free. Also a high specificity is useful since it can be concluded that an individual, who tests positive, indeed has the disease.

Limitations of sensitivity and specificity

As can be deduced from the above paragraphs, sensitivity and specificity are of no use when it comes to estimating the probability of a disease in a single individual. If an individual tests positive, clinicians would be most interested in the probability of the disease given the test is positive [14]. In this case, neither sensitivity nor specificity is useful since these two measures are defined on the basis of people with or without the disease [1]. The presence of the disease in individuals with a negative or a positive result is measured by predictive values [14]. The following section will discuss both positive and negative predictive values.

2.1.5 Predictive values

Positive predictive value

This is the percentage that a positive test correctly identifies a diseased individual [10]. Positive predictive value will be abbreviated as PPV and can be expressed as

$$\text{PPV}=\frac{\text{TP}}{\text{TP+FP}} \tag{3}$$

From Equation 3, it is evident that the PPV of a screening test is the proportion of all individuals who tested positive by both the screening test and gold standard to individuals who have a positive screening test [1]. PPV can also be referred to as the 'post-test probability of the disease given a test is positive'. PPV varies with change in the prevalence of the disease.

Negative predictive value

This is the percentage that disease-free individuals test negative [15]. Negative predictive value will be abbreviated as NPV. This percentage aims to answer the question of how likely it is that an individual does not have the disease given that the test result is negative [13]. The equation for NPV is expressed as

$$\text{NPV}=\frac{\text{TN}}{\text{TN+FN}} \tag{4}$$

From Equation 4, it is clear that NPV is the proportion of the individuals that have a negative result who do not have the disease. NPV also varies with the change in prevalence of the disease.

2.1.6 Prevalence

Prevalence is the probability of the actual presence of the disease in the population at a given time [12]. The equation for prevalence is expressed as

$$\text{Prevalence} = \frac{\text{TP+FN}}{\text{TP+FN+TN+FP}} \tag{5}$$

The predictive value of a screening test is determined from sensitivity and specificity where the prevalence of the condition is known. Equation 3 can be rewritten in terms of sensitivity, specificity and prevalence as

$$\text{PPV}=\frac{\text{sensitivity}\times\text{prevalence}}{\text{sensitivity}\times\text{prevalence}+(\text{1-specificity})\times(\text{1-prevalence})} \tag{6}$$

Since PPV vary with the change in prevalence of disease, it will be wrong to directly apply a PPV,

calculated from Equation 3, to a new population, where the prevalence of the disease is different from the prevalence of the previous population [1].

The higher the prevalence, the higher the PPV and thus the higher the chance that a positive result is able to predict the presence of the disease. The lower the prevalence, the lower the PPV, even when using a test with high sensitivity and a high specificity [1]. When prevalence is low, an individual, who tests positive, might not necessarily have the disease.

Like PPV, NPV can also be computed from sensitivity, specificity and prevalence of disease as shown below [1]:

$$\text{NPV}=\frac{\text{specificity}\times(\text{1-prevalence})}{(\text{1-sensitivity})\times\text{prevalence}+\text{specificity}\times(\text{1-prevalence})} \tag{7}$$

The higher the prevalence, the higher the NPV and thus the higher the chance that a negative result is able to predict an absence of the disease. When the prevalence is low an individual who tests negative might not necessarily be disease free [1].

2.1.7 Accuracy

Any assessment of a screening test requires a comparison with the gold standard test. The simplest measure of this comparison is a percentage of the case where the screening test is correct and this percentage is called accuracy [12]. The numerical value of accuracy is given by the proportion of the number of correct assessments in the selected population as shown below [16]:

$$\text{Accuracy} =\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{8}$$

Equation 8 is a very simple index and must be interpreted cautiously [16]. The effect of prevalence of the disease must be incorporated, and also right and wrong screening test decisions must be separated [12]. Thus, accuracy will be computed as:

$$\text{Accuracy} = \text{sensitivity}\times\text{prevalence}+\text{specificity}\times(\text{1-prevalence}) \tag{9}$$

Equation 9 suggests that, even though the numerical values of both sensitivity and specificity are high, it does not imply that the accuracy of the test is similarly high since accuracy is controlled by prevalence [16]. Therefore, it is possible to have both a high specificity and a high sensitivity, and a low value of accuracy.

2.2 Probability indices

The relationship among sensitivity, specificity, PPV, NPV, accuracy and prevalence can be derived using probabilities.

2.2.1 Decision fractions

Some additional terminology for decision fractions will be introduced [12, 16]:

- True positive rate (TPR) which is the same as sensitivity
- True negative rate (TNR) which is the same as specificity
- False positive rate (FPR) which is (1-specificity) or (1-TNR)
- False negative rate (FNR) which is (1-sensitivity) or (1-TPR)

Recall that the mathematical formulaes for TPR and TNR were given in Equation 1 and Equation 2 . Now the mathematical formulaes for FPR and FNR will be derived:

$$\begin{aligned} \text{FPR} &= 1 - \text{TNR} \\ &= 1 - \frac{\text{TN}}{\text{TN+FP}} \\ &= \frac{\text{FP}}{\text{TN+FP}} \end{aligned}$$

$$\therefore \text{FPR} = \frac{\text{FP}}{\text{TN+FP}} \tag{10}$$

$$\begin{aligned} \text{FNR} &= \text{1-TPR} \\ &= 1 - \frac{\text{TP}}{\text{TP+FN}} \\ &= \frac{\text{FN}}{\text{TP+FN}} \end{aligned}$$

$$\therefore \text{FNR} = \frac{\text{FN}}{\text{TP+FN}} \tag{11}$$

It can easily shown from Equations 1 and 11 that

$$\text{TPR+FNR} = 1$$

and from Equations 2 and 10 that

$$\text{TNR+FPR} = 1$$

Now let D represent the disease and T represent the result of the screening test, such that the symbols

- T+ represents a positive test,
- T- represents a negative test,
- D+ represents the presence of the disease and
- D- represents the absence of the disease.

TPR, TNR, FPR and FNR will now be redefined in terms of conditional probabilities:

$$\text{TPR} = P(T+|D+) \tag{12}$$

Equation 12 will be read as the probability of a positive test given the presence of the disease [12].

$$\text{TNR} = P(T-|D-) \tag{13}$$

Equation 13 will be read as the probability of a negative test given the absence of the disease [12].

$$\text{FPR} = P(T+|D-) \tag{14}$$

Equation 14 will be read as the probability of a positive test given the absence of the disease [12].

$$\text{FNR} = P(T-|D+) \tag{15}$$

Equation 15 will be read as the probability of a negative test given the presence of the disease [12].

The above use of conditional probabilities emphasizes that the decision fractions are conditional on the actual disease status. The prevalence of the disease is denoted as $P(D+)$. The probability of the absence of the disease in the population at a given time is given as $1 - P(D+) = P(D-)$ [12].

2.2.2 Predictive values

Equations 6 and 7, which represent the PPV and NPV respectively will be written in terms of probability functions. Equation 6 can be written as

$$\begin{aligned} \text{PPV} &= \text{PPV}=\frac{\text{TPR}\times\text{prevalence}}{\text{TPR}\times\text{prevalence}+\text{FPR}\times(\text{1-prevalence})} \\ &= \frac{P(T+|D+)\times P(D+)}{[P(T+|D+)\times P(D+)]+[P(T+|D-)\times P(D-)]} \end{aligned}$$

Thus PPV, in terms of conditional probabilities is expressed as

$$\text{PPV} = \frac{P(T+|D+)\times P(D+)}{[P(T+|D+)\times P(D+)]+[P(T+|D-)\times P(D-)]} \tag{16}$$

Now, applying Bayes theorem to Equation 16, results in

$$\begin{aligned} \text{PPV} &= \frac{P(T+|D+)\times P(D+)}{[P(T+|D+)\times P(D+)]+[P(T+|D-)\times P(D-)]} \\ &= \frac{P(T+|D+)\times P(D+)}{P(T+)} \\ &= P(D+|T+) \end{aligned}$$

$$\therefore \text{PPV} = P(D+|T+) \tag{17}$$

In a similar way, Equation 7 can be expressed

$$\begin{aligned} \text{NPV} &= \frac{\text{TNR}\times(\text{1-prevalence})}{\text{FNR}\times\text{prevalence}+\text{TNR}\times(\text{1-prevalence})} \\ &= \frac{P(T-|D-)\times P(D-)}{[P(T-|D+)\times P(D+)]\times P(T-|D-)\times P(D-)} \end{aligned}$$

Thus NPV, in terms of conditional probabilities is expressed as

$$\text{NPV} = \frac{P(T-|D-)\times P(D-)}{[P(T-|D+)\times P(D+)]\times P(T-|D-)\times P(D-)} \tag{18}$$

Applying Bayes theorem to Equation 18, results in

$$\begin{aligned} \text{NPV} &= \frac{P(T-|D-) \times P(D-)}{[P(T-|D+) \times P(D+)] \times P(T-|D-) \times P(D-)} \\ &= \frac{P(T-|D-) \times P(D-)}{P(T-)} \\ &= P(D-|T-) \end{aligned}$$

$$\therefore \text{NPV} = P(D-|T-) \tag{19}$$

2.2.3 Likelihood ratio

For the purpose of estimating an individual's probability of disease status, the likelihood ratio is used [2]. The likelihood ratio is a combination of sensitivity and specificity and is a more useful measure. The likelihood ratio provides a synopsis of how many times more likely individuals with a disease are to have a specific result than individuals without the disease [8]. The likelihood ratio for a positive test is denoted as LR+ and the likelihood ratio for a negative test is denoted as LR-.

Likelihood ratio for a positive test (LR+)

The definition of LR+ is given as [2]:

$$\text{LR+} = \frac{\text{probability of a positive test in an individual with disease}}{\text{probability of a positive test in an individual who is disease free}}$$

It can be seen from the above definition that the numerator is the definition of the sensitivity of the test and the denominator essentially is the opposite of specificity (1-specificity)[2]. Thus LR+ can therefore be written as:

$$\text{LR+} = \frac{\text{TPR}}{\text{FPR}} \tag{20}$$

Using Equation 20, as well as the probability definitions of TPR and FPR, LR+ can be expressed as

$$\text{LR+} = \frac{P(T+|D+)}{P(T+|D-)} \tag{21}$$

LR+ value	Interpretation
LR+ > 10	increases the probability to rule-in a disease for a person who tests positive
LR+ > 1	a positive test is most likely to occur in people who are diseased than people who are not diseased
LR+ < 0.1	rule-out the probability that a person who has the disease tests positive
LR+ < 1	a positive test is less likely to occur in people who are diseased than people who are not diseased

Table 2: Interpretation of LR+

Table 2 provides a summary of the different interpretations of LR+ values [2, 6].

Likelihood ratio for a negative test (LR-)

The definition of LR- is given as

$$\text{LR-} = \frac{\text{probability of a negative test in an individual with the disease}}{\text{probability of a negative test in an individual who is disease free}}$$

It can be seen from the above definition that the numerator is the opposite of sensitivity (1-sensitivity) and the denominator is the definition of specificity [2]. Hence, LR- can be written as

$$\text{LR-} = \frac{\text{TNR}}{\text{FNR}} \tag{22}$$

Using the probability definitions of TNR and FNR, Equation 22 will be represented as follows

$$\text{LR}- = \frac{P(T-|D+)}{P(T-|D-)} \tag{23}$$

LR- value	Interpretation
LR- > 10	increases the probability to rule-in a disease for a person who tests negative
LR- > 1	a negative test is most likely to occur in people who are diseased than people who are not diseased
LR- < 0.1	rule-out the probability that a person who has the disease tests negative
LR- <1	a negative test is less likely to occur in people who are diseased than people who are not diseased

Table 3: Interpretation of LR-

Table 3 provides a summary of the different interpretations of LR- values [2, 6].

2.2.4 Pre-test probability

Prior to knowing the test result, clinicians usually estimate an individual's disease status based on their personal knowledge and experience and also on the prevalence of the disease. This estimated probability is known as the pre-test probability.

2.2.5 Post-test probability

The probability of an individual being diseased, after the test result is known, is called the post-test probability. Post-test probability is important since it provides further information useful for diagnosis [7]. The post-test probability can either be estimated by using Bayes theorem or by using the Fagan's nomogram [2].

Bayes theorem

$$\text{post-test odds}=\text{pre-test odds} \times \text{likelihood ratio} \tag{24}$$

In Equation 24, odds were used instead of probabilities. Pre-test probabilities need to be converted into pre-test odds first before the calculation can take place [2]. First it will be shown how to convert to pre-test odds:

$$\text{pre-test odds}=\frac{\text{pre-test probability}}{1-\text{pre-test probability}}$$

Let P(D+) denote the pre-test probability, thus substituting with P(D+) in the above definition and using Equation 21, Equation 24 can be mathematically represented as

$$\begin{aligned}\text{post-test odds} &= \frac{P(D+)}{1-P(D+)} \times \frac{P(T+|D+)}{P(T+|D-)} \\ &= \frac{P(D+)P(T+|D+)}{P(D-)P(T+|D-)}\end{aligned}$$

Hence;

$$\text{post-test odds}=\frac{P(D+)P(T+|D+)}{P(D-)P(T+|D-)} \tag{25}$$

Clinicians are interested in post-test probabilities and not post-test odds, therefore Equation 25 has to be converted back to post-test probabilities. The conversion is shown below:
Since, post-test probability $= \frac{\text{post-test odds}}{1+\text{post-test odds}}$ it follows that

$$\begin{aligned}
\text{post-test probability} &= \frac{\frac{P(D+)P(T+|D+)}{P(D-)P(T+|D-)}}{1+\frac{P(D+)P(T+|D+)}{P(D-)P(T+|D-)}} \\
&= \frac{\frac{P(D+)P(T+|D+)}{P(D-)P(T+|D-)}}{\frac{P(D-)P(DT+|D+)+P(D+)P(T+|D+)}{P(D-)P(T+|D-)}} \\
&= \frac{P(T+|D+)\times P(D+)}{[P(T+|D+)\times P(D+)]+[P(T+|D-)\times P(D-)]}
\end{aligned}$$

By Bayes theorem post-test probability can be expressed as

$$\begin{aligned}
&= \frac{P(D+)P(T+|D+)}{P(T+)} \\
&= P(D+|T+)
\end{aligned}$$

$$\therefore \text{ post-test probability } = P(D+|T+) \tag{26}$$

Equation 26 is the same as the equation for PPV. This indeed supports the statement that was mentioned in section 2.1.5 that PPV is also referred to as the post-test probability of disease, given that the test is positive.

The Fagan's nomogram Another way of calculating post-test probabilities is by using the Fagan's nomogram graphical tool. With the aid of a diagram in Figure 1, it will be explained how the Fagan's nomogram works [11].

A straight line is drawn from an individual's pre-test probability of illness through the likelihood ratio of the test and this line will meet the correct post-test probability at the post-test probability of the test[2]. A practical illustration will be shown in section 2.3.

2.2.6 Accuracy

At this point, the concept of accuracy will also be represented in terms of conditional probability functions. Equation 9 will be rewritten as:

$$\begin{aligned}
\text{Accuracy} &= \text{TPR}\times\text{prevalence}+\text{TNR}\times(\text{1-prevalence}) \\
&= [P(T+|D+)\times P(D+)]+[P(T-|D-)\times P(D-)]
\end{aligned}$$

$$\therefore \text{Accuracy} = [P(T+|D+)\times P(D+)]+[P(T-|D-)\times P(D-)] \tag{27}$$

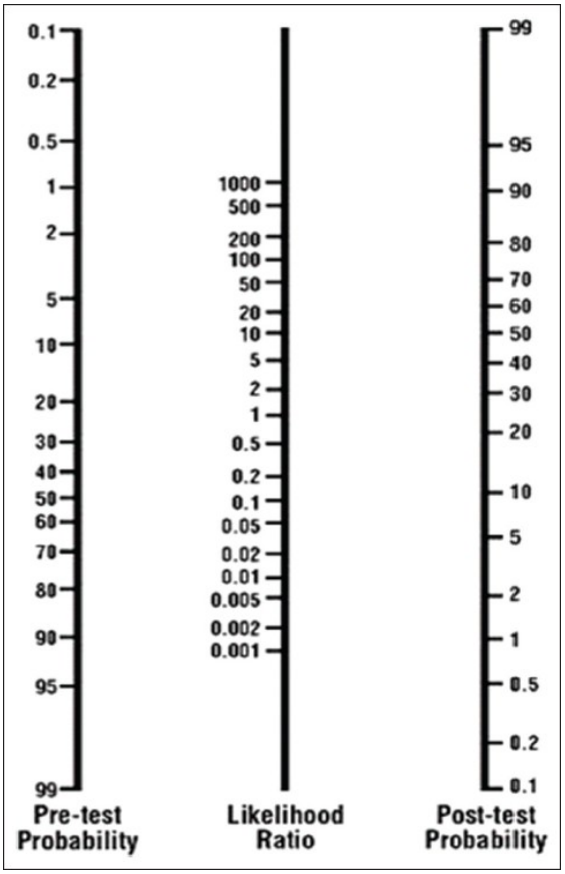


Figure 1: The Fagan's nomogram

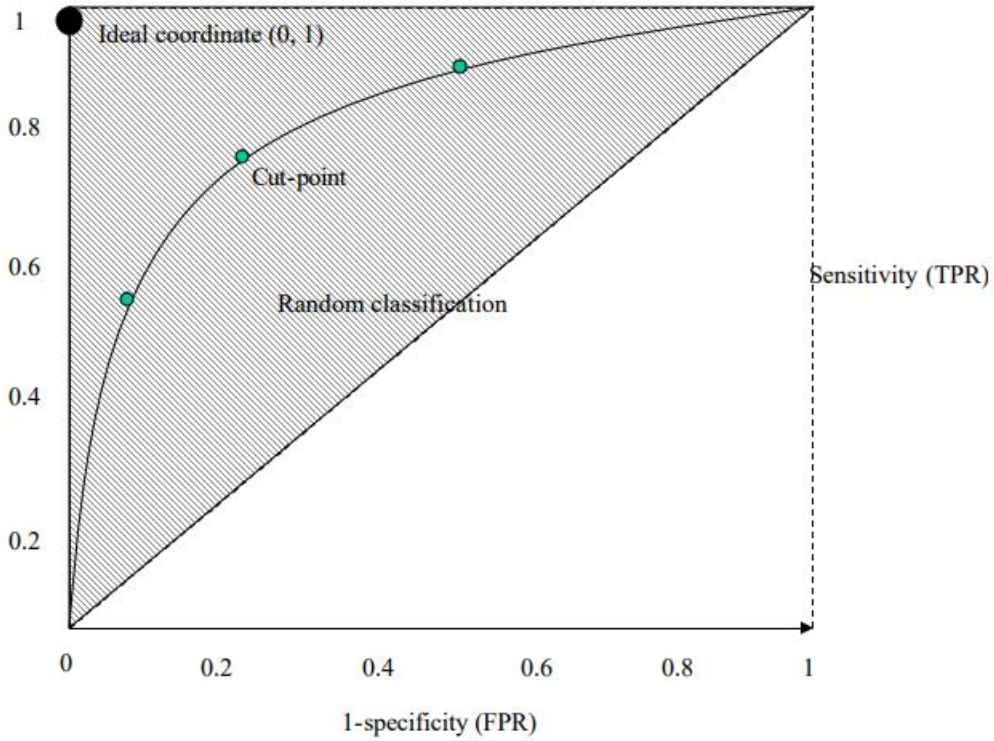


Figure 2: A ROC space

2.2.7 Receiver operating characteristics (ROC) analysis

Most clinical test results are provided on a continuous scale since their results are quantitative, thus a cut-off point for positive and negative test results needs to be chosen. This cut-off point is used to decide the presence and the absence of the disease. A positive test result is considered as abnormal and a negative test result is considered as normal. As a result, the sensitivity and the specificity vary, depending on the chosen cut-off point for normal or abnormal results [3]. The receiver operator curve (ROC) is a graph that represents the relationship between sensitivity and specificity [16]. The ROC curve is useful for determining and comparing the accuracy of a screening test [3]. The ROC curve also assists in the optimal model decision (i.e. a cut-off value such that the optimal values of sensitivity and specificity are achieved) through deciding the best limit for the screening test [16].

Figure 2 shows a ROC space which is a figure that is formed by all the possible combinations of TPR and FPR. The ROC curve is a graphical technique that is plotted inside the ROC space. The ROC curve is obtained by plotting TPR (sensitivity) on the y-axis against FPR (1-specificity) on the x-axis [16]. The diagonal line in Figure 2 goes from the coordinates (0,0) through (1,1) and serves as a reference line. These coordinates represent a screening test with a sensitivity of 50% and a specificity of 50%. The reference line actually represents the characteristics of a test which is useless at discriminating between those with disease and those without the disease, since the screening test detects an equal number of true and false positives [3, 16]. Thus the closer the points on the ROC curve are to the diagonal, the

less accurate the screening test results are. A diagnostic test that perfectly discriminates between those with and those without disease will yield a curve. An ideal medical test will yield a sensitivity of 100% and a specificity of 100%, corresponding to the coordinates (0,1) on the ROC graph. Such a test will perfectly identify individuals that are diseased and those that are disease free [16]. A test that is better at discriminating between those are diseased and those that are disease free has a ROC curve that is closer to the ideal coordinate (0,1). The faster a curve approaches (0,1) the more accurate the test outcomes are [3].

Purposes of a ROC curve

- to assess the usefulness of the screening test,
- to determine the cut-off point where optimal sensitivity and specificity can be achieved and
- to compare the usefulness of two or more screening tests.

Diagnostic accuracy The area under the ROC curve (AUC) is important because it measures the accuracy of a screening test [16]. The following equation is used to measure the AUC of a ROC curve:

$$\text{AUC} = \int_0^1 ROC(t)dt \tag{28}$$

where t=(1-specificity)

Table 4 gives a summary of the different interpretations of AUC ranges to classify accuracy [16].

AUC range	**Classification**
$0.9 < \text{AUC} < 1.0$	Excellent
$0.8 < \text{AUC} < 0.9$	Good
$0.7 < \text{AUC} < 0.8$	Not good
$0.6 < \text{AUC} < 0.7$	Useless

Table 4: Grouping of accuracy by AUC of a ROC curve

Lastly, LR+ can be determined by the tangent line to a cut-off point and if [3]:

LR+ value	**Interpretation**
LR+ < 1	the selected cut-off point decreases the disease likelihood
LR+ =1	the chosen cut-off point will not give any extra useful information to identify a true positive outcome
LR+ > 1	the selected cut-off point will be useful in identifying the true positive value

Table 5: Determining the value of LR+ from the tangent line to a cut-off point

2.2.8 Asymptotic and exact 95% confidence intervals

Sensitivity, specificity, predictive values, accuracy and AUC are statistical estimates of the population screening test measures and therefore should be reported as confidence intervals with a confidence coefficient that is usually 95%. The importance of the 95% confidence interval is that it informs the reader about the interval in which 95% of a certain screening test measure will fall if the study was done repeatedly. The corresponding confidence intervals can be calculated by using standard techniques for proportions [16, 3]. The binomial distribution is used to construct the exact confidence interval and the asymptotic confidence interval is constructed by using a normal approximation to the binomial distribution. The exact confidence interval is preferred since it can reach the exact estimate. The asymptotic confidence interval is dependent on whether the sample proportion is a good approximation of the binomial distribution [16].

2.3 Hypothetical example to illustrate concepts

2.3.1 Formulation of the problem

For illustration purposes, a hypothetical example will be discussed. Table 6, containing data for a disease will be used for evaluating a screening test against a gold standard test. This evaluation will be achieved by calculating the screening test measures. The population exists of 1172 people. According to the screening test, 532 of these individuals are diseased and 640 disease-free. However, according to the gold standard test, 469 out of the 1172 individuals are actually diseased and 703 are disease-free.

		GOLD	**STANDARD**	
		Person with disease	**Person without disease**	**Total**
SCREENING TEST	**Positive**	TP=401	FP=131	TP + FP=532
	Negative	FN=68	TN=572	TN + FN=640
	Total	TP + FN=469	TN + FP=703	1172

Table 6: Table containing decision functions from a hypothetical population

2.3.2 Sensitivity and specificity

Sensitivity The numerical value for sensitivity will be calculated using Equation 1:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} = \frac{401}{469} = 0.855$$

The sensitivity of this test is 86% which means that the test has correctly identified about 86% of the individuals to be diseased. The above numerical value for sensitivity is high enough to suggest that an individual with a negative result is indeed disease free.

Specificity Using Equation 2, the numerical value of specificity is calculated as:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} = \frac{572}{703} = 0.814$$

The specificity of this test is 81% which means that the test has correctly excluded about 81% of the individuals to be diseased. i.e they are disease free. The above numerical value for specificity is high enough to suggest that an individual, tested positive, indeed has the disease.

2.3.3 Predictive values

Positive predictive value

Method 1

Using Equation 3, the numerical value for PPV is calculated as follows:

$$\text{PPV} = \frac{\text{TP}}{\text{TP+FP}} = \frac{401}{532} = 0.754$$

Method 2

Recall that the prevalence of the disease is given by the proportion of all positive assessments divided the total number of assessments;

$$\text{prevalence} = \frac{469}{1172} = 0.40$$

Now, using Equation 6, the numerical value for PPV can also be calculated as follows:

$$\begin{aligned} \text{PPV} &= \frac{\text{sensitivity} \times \text{prevalence}}{\text{sensitivity} \times \text{prevalence} + (\text{1-specificity}) \times (\text{1-prevalence})} \\ &= (0.855 \times 0.40) + (1 - 0.814) \times (1 - 0.40) \\ &= 0.754 \end{aligned}$$

The PPV means that the screening test has correctly identified about 75% of the individuals with the disease.

Negative predictive value

Method 1

Using Equation 4, NPV is calculated as:

$$\text{NPV} = \frac{\text{TN}}{\text{TN+FN}} = \frac{572}{640} = 0.891$$

Method 2

Using Equation 7 and the above calculated value for prevalence:

$$\begin{aligned} \text{NPV} &= \frac{\text{specificity}\times(\text{1-prevalence})}{(\text{1-sensitivity})\times\text{prevalence}+\text{specificity}\times(\text{1-prevalence})} \\ &= \frac{0.814 \times 0.40}{(1-0.855)\times(0.40)+(0.814)\times(1-0.40)} \\ &= 0.891 \end{aligned}$$

The above NPV means that the screening test has correctly identified 89% of the individuals to be disease -free.

REMARK: Predictive value and disease prevalence Consider a second population of 1000 people that has the disease, but with a different prevalence value as the one mentioned above. A prevalence of the disease in the second population of 24% means that 240 people have the disease and 760 are disease free. The sensitivity and specificity values of the screening test are already known to be 86% and 81% respectively. A sensitivity of 86% means that 86% of the 240 individuals will test positive (i.e. $206.4 \approx$ 206 people have a true positive test). A specificity of 81% means that 81% of the people without the disease will test negative (i.e TN) or that 19% of the 760 individuals that are disease free will test positive (i.e FP). The value for FP is 144.4. Using Equation 3, PPV is calculated as follows:

$$\text{PPV} = \frac{\text{TP}}{\text{TP+FP}} = \frac{206.4}{206.4+144.4} = 0.59$$

However, it is much simpler to calculate the PPV value of the new population where the sensitivity and specificity of the screening test and prevalence are known, by using Equation 6:

$$\begin{aligned} \text{PPV} &= \frac{\text{sensitivity}\times\text{prevalence}}{\text{sensitivity}\times\text{prevalence}+(\text{1-specificity})\times(\text{1-prevalence})} \\ &= \frac{0.855 \times 0.24}{(0.855\times0.24)+(1-0.814)\times(1-0.24)} \\ &= 0.59 \end{aligned}$$

The above procedure can also be used to calculate the NPV of the second population.

2.3.4 Likelihood ratios

The likelihood ratios for this screening test are calculated as follows:

Likelihood ratio for a positive test (LR+)

From Equation 21, using the above numerical values of sensitivity and specificity, the value for LR+

is calculated as

$$\text{LR}+ = \frac{P(T+|D+)}{P(T+|D-)} = \frac{0.855}{0.186} = 4.597$$

A numerical value of 4.597 for LR+ means that a person with this disease is about 5 times more likely to test positive than a person who does not have this disease.

Likelihood ratio for a negative test (LR-)

Using Equation 23, the numerical value for LR- is calculated as

$$\text{LR}- = \frac{P(T-|D+)}{P(T-|D-)} = \frac{0.145}{0.814} = 0.178$$

Thus, there is an 18% chance of to test negative for an individual who is diseased. Hence, a person without the disease is about 5 ($= \frac{1}{0.2}$) times more likely to test negative than a person with the disease.

2.3.5 Post-test probability

- Bayes theorem: using Bayes theorem, the post-test probabilities will be calculated the same way as PPV and NPV were calculated.
- Fagan's nomogram: in Figure 3, a straight line was drawn through the pretest probability ($P(D+)$) of 40% and the LR+ of 5. This yields a post-test probability of about 76%.

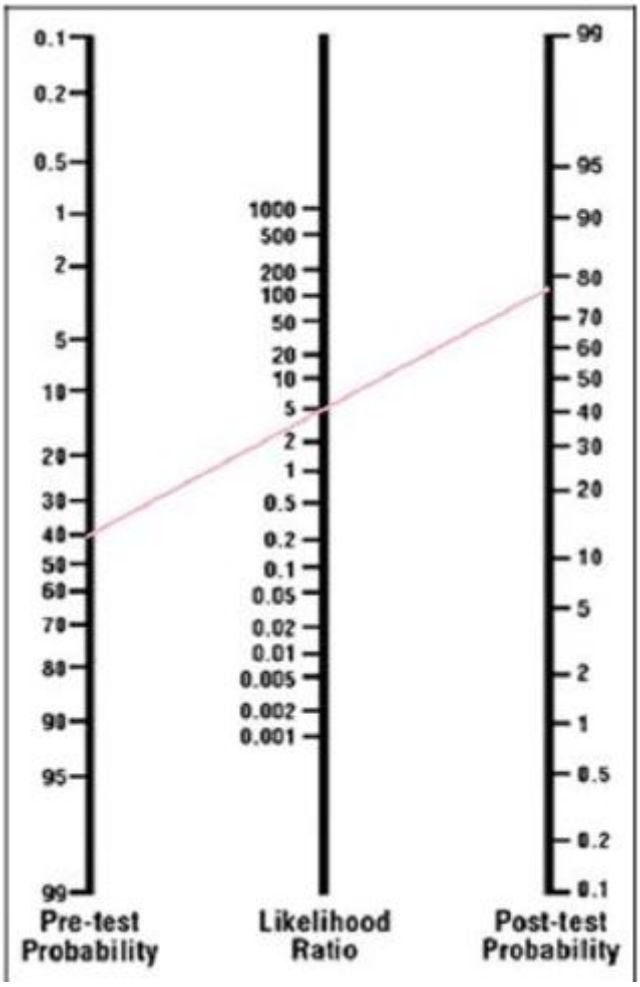


Figure 3: The utilization of the Fagan's nomogram

3 Application

The data analysis for this research report will be generated using SAS software. Copyright © 2017 SAS Institute Inc. SAS/ACCESS® 9.4 Interface to ADABAS: Reference. Cary, NC: SAS Institute Inc.

3.1 Formulation of the screening and gold standard test

A clinical trial is conducted to evaluate a diagnostic test designed to detect the presence of atherosclerosis. Atherosclerosis is usually tested for using coronary angiography, the gold standard test. The diagnostic test is performed on a random sample of 653 individuals that were divided according to their body mass index (BMI) as shown in Table 7.

The data provided in Table 7 will be divided into two categories according to the BMI. The categories are as follows:

- overweight with a BMI between 25 and 29.9 and

BMI	DISEASED	DISEASE FREE	TOTAL
25	1	50	51
26	2	46	48
27	4	45	49
28	6	44	50
29	7	35	42
30	30	27	57
31	38	27	65
32	55	11	66
33	67	11	78
34	73	7	80
35	20	5	25
36	10	0	10
37	8	0	8
38	12	0	12
39	12	0	12
TOTAL	345	308	653

Table 7: Sample of individuals to be diagnosed for atherosclerosis

- obese with a BMI greater than 29.9

Table 8 summarizes the data for individuals in the two weight groups.

WEIGHT	DISEASED	DISEASE FREE	TOTAL
Overweight	20	220	240
Obese	325	88	413
TOTAL	345	308	653

Table 8: Categorized data for the two weights

It is clear from Table 8 that weight is an important factor in diagnosing a patient for atherosclerosis since most of the diseased individuals are obese.

Below are 2×2 cross-tabulation tables that summarize the data for individuals who are overweight and obese.

Overweight		**CORONARY ANGIOGRAPHY**		
		Person with disease	**Person without disease**	**Total**
SCREENING TEST	**Positive**	19	41	60
	Negative	1	179	180
	Total	20	220	240

Table 9: 2×2 contingency table to summarize data for individuals who are overweight

Obese		**CORONARY ANGIOGRAPHY**		
		Person with disease	**Person without disease**	**Total**
SCREENING TEST	**Positive**	305	26	331
	Negative	20	62	82
	Total	325	88	413

Table 10: 2×2 contingency tables to summarize data for individuals who are obese

Below is the SAS software output of the decision functions in Tables 9 and 10. The SAS software code is given in the appendix.

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Exposure by Response		
Exposure(SCREENING TEST)	Response(CORONARY ANGIOGRAPHY)		
	Positive	Negative	Total
Positive	19 7.92 31.67 95.00	41 17.08 68.33 18.64	60 25.00
Negative	1 0.42 0.56 5.00	179 74.58 99.44 81.36	180 75.00
Total	20 8.33	220 91.67	240 100.00

Table 11: 2×2 Contingency table to summarize data for individuals who are overweight

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Exposure by Response		
Exposure(SCREENING TEST)	Response(CORONARY ANGIOGRAPHY)		
	Positive	Negative	Total
Positive	305 73.85 92.15 93.85	26 6.30 7.85 29.55	331 80.15
Negative	20 4.84 24.39 6.15	62 15.01 75.61 70.45	82 19.85
Total	325 78.69	88 21.31	413 100.00

Table 12: 2×2 Contingency table to summarize data for individuals who are obese

In the following sections the SAS software codes and outputs will be provided for the numerical measures used to evaluate the screening test measures for the two weight groups.

3.2 Sensitivity and specificity

SAS software Code

```
*sensitivity and specificity*;
sensitivity = TP/(TP+FN);
specificity = TN/(TN+FP);
print sensitivity specificity;
```

SAS software Output

Overweight		Obese	
sensitivity	specificity	sensitivity	specificity
0.95	0.8136364	0.9384615	0.7045455

Table 13: Measures for sensitivity and specificity

- The sensitivity of the screening test for individuals who are overweight is 95%, which means that the test has correctly identified 95% of the individuals to have atherosclerosis . The specificity is 81% which means that the test has correctly excluded 81% of the overweight individuals to have atherosclerosis.
- The sensitivity of the screening test for individuals who are obese is 94%, which means that the test has correctly identified 95% of the individuals to have atherosclerosis . The specificity is 71% which means that the test has correctly excluded 81% of the obese individuals to have atherosclerosis.

3.3 Prevalence

SAS software Code

```
*prevalence*;
prevalence = (TP+FN)/(TP+TN+FN+FP);
print prevalence;
```

SAS software Output

Overweight	Obese
prevalence	prevalence
0.0833333	0.7869249

Table 14: Measures for prevalence

Using the definition of prevalence, it is correct to say that in the overweight category only 8% of the individuals have atherosclerosis and 79% of the individuals that are obese have atherosclerosis. There is a higher prevalence of the disease in individuals that are obese because individuals who are obese are more likely to have atherosclerosis.

3.4 Predictive values

Predictive values will be calculated using both methods that were mentioned in sections 2.1.5 and 2.1.6.

SAS software Code

```
*PPV*
**method1**;
PPV1 = TP/(TP+FP);
**method2**;
A = sensitivity*prevalence;
B = (1-specificity)*(1-prevalence);
PPV2 = A/(A+B);
print PPV1 PPV2;
*NPV*
**method1**;
NPV1 = tn/(tn+fn);
**method2**;
C = specificity*(1-prevalence);
D = (1-sensitivity)*prevalence;
NPV2 = C/(C+D);
print NPV1 NPV2;
quit;
```

SAS software Output

Overweight		Obese	
PPV1	PPV2	PPV1	PPV2
0.3166667	0.3166667	0.9214502	0.9214502
NPV1	NPV2	NPV1	NPV2
0.9944444	0.9944444	0.7560976	0.7560976

Table 15: Measures for predictive values

- The PPV for people who are overweight means that the positive test has correctly identified 31% of the individuals that have atherosclerosis. The NPV on the other hand, means that the screening test has correctly identified 99% of the overweight individuals that do not have atherosclerosis.
- The PPV for people who are obese means that the positive test has correctly identified 92% of the individuals that have atherosclerosis. The NPV on the other hand, means that the screening test has correctly identified 75% of the obese individuals that do not have atherosclerosis.

3.5 Accuracy

SAS software Code

```
*accuracy*;
proc sql;
create table acc as select (TP+TN)/(TN+TP+FN+FP) as Accuracy from cont1;
proc print;
quit;
```

SAS software Output

Overweight	Obese
accuracy	accuracy
0.825	0.8886199

Table 16: Measures for accuracy

- The screening test is 83% accurate in correctly identifying and excluding atherosclerosis in individuals who are overweight.
- The screening test is 89% accurate in correctly identifying and excluding atherosclerosis in individuals who are obese.

3.6 Asymptotic and exact 95% confidence intervals for sensitivity, specificity and accuracy

A data set for calculating both the asymptotic and exact confidence intervals of sensitivity, specificity and accuracy will be created from the values of the decision functions. The SAS software code for this procedure will be given in the appendix.

SAS software Code to calculate the confidence intervals;

```
 proc sort data=confint;
by group;
 proc freq data=confint;
weight count;
by group;
tables response/alpha=0.05 binomial (p=0.5);
exact binomial;
run;
```

SAS software output

Binomial Proportion	
response = 0	
Proportion (P)	0.8250
ASE	0.0245
95% Lower Conf Limit	0.7769
95% Upper Conf Limit	0.8731
Exact Conf Limits	
95% Lower Conf Limit	0.7709
95% Upper Conf Limit	0.8709

Binomial Proportion	
response = 0	
Proportion (P)	0.9500
ASE	0.0487
95% Lower Conf Limit	0.8545
95% Upper Conf Limit	1.0000
Exact Conf Limits	
95% Lower Conf Limit	0.7513
95% Upper Conf Limit	0.9987

Binomial Proportion	
response = 0	
Proportion (P)	0.8136
ASE	0.0263
95% Lower Conf Limit	0.7622
95% Upper Conf Limit	0.8651
Exact Conf Limits	
95% Lower Conf Limit	0.7558
95% Upper Conf Limit	0.8628

Table 17: The 95% confidence intervals from the SAS software procedure output for individuals who are overweight

Binomial Proportion	
response = 0	
Proportion (P)	0.8886
ASE	0.0155
95% Lower Conf Limit	0.8583
95% Upper Conf Limit	0.9190
Exact Conf Limits	
95% Lower Conf Limit	0.8542
95% Upper Conf Limit	0.9173

Binomial Proportion	
response = 0	
Proportion (P)	0.9385
ASE	0.0133
95% Lower Conf Limit	0.9123
95% Upper Conf Limit	0.9646
Exact Conf Limits	
95% Lower Conf Limit	0.9066
95% Upper Conf Limit	0.9620

Binomial Proportion	
response = 0	
Proportion (P)	0.7045
ASE	0.0486
95% Lower Conf Limit	0.6092
95% Upper Conf Limit	0.7999
Exact Conf Limits	
95% Lower Conf Limit	0.5978
95% Upper Conf Limit	0.7971

Table 18: The 95% confidence intervals from the SAS software procedure output for individuals who are obese

The results from Figures 17 and 18 are summarized in Table 19.

	Proportion	Asymptotic 95% CI	Exact 95% CI
ACCURACY			
Overweight	0.8250	(0.7769,0.8731)	(0.7709,0.8709)
Obese	0.8886	(0.8583,0.9190)	(0.8542,0.9173)
SENSITIVITY			
Overweight	0.9500	(0.8545,1.0000)	(0.7513,0.9987)
Obese	0.9385	(0.9123,0.9646)	(0.9066,0.9620)
SPECIFICITY			
Overweight	0.8136	(0.7622,0.8651)	(0.7758,0.8628)
Obese	0.7045	(0.6092,0.7999)	(0.5978,0.7971)

Table 19: The 95% confidence intervals results for the overweight and obese groups

If the study was to be done repeatedly, 95% of the values for accuracy, sensitivity and specificity would respectively fall in the confidence intervals given in Table 19.

3.7 Receiver operating characteristics (ROC) curve and AUC analysis

To compute the ROC curve, the data for the individual BMIs in Table 7 was used. The SAS software code that generated the data that was used to compute the ROC curve will be given in the appendix. Below is the code that was used to construct the ROC curve.

```
ods graphics on;
proc logistic data=dataroc plots(only)=roc(id=obs);
model diseased/n=bmi / scale=none
     clparm=wald
clodds=pl
     rsquare;
units bmi=1;
effectplot;
run;
ods graphics off;
```

SAS software output

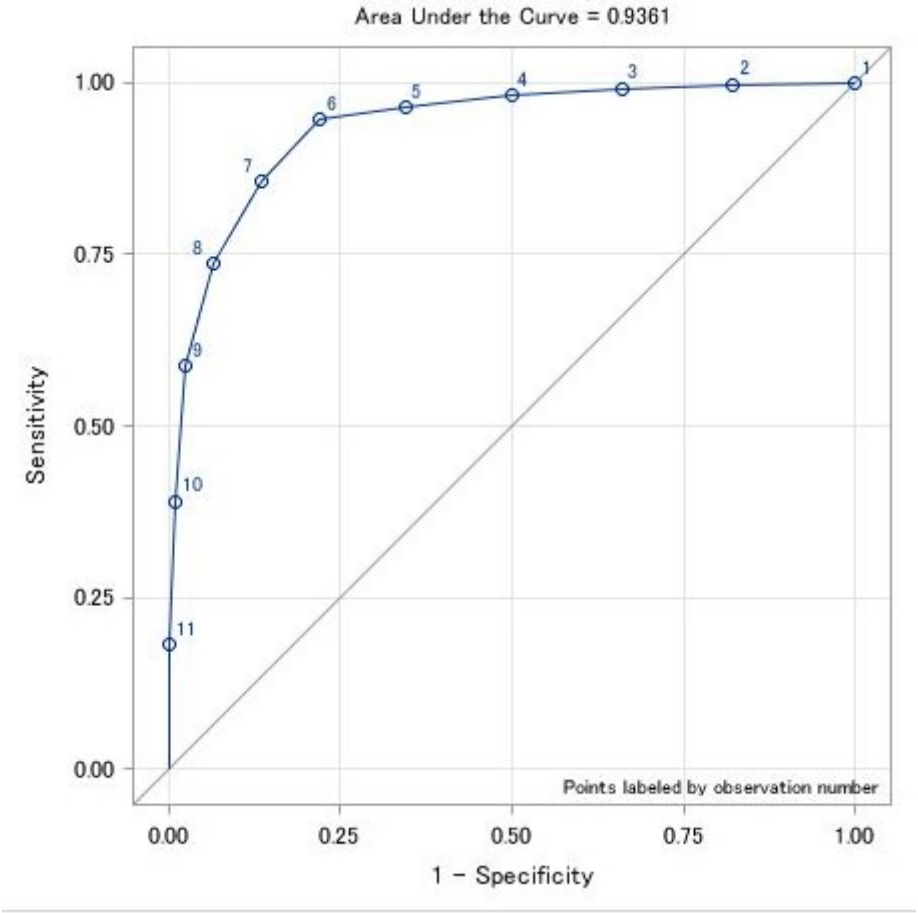


Figure 4: ROC curve for combined groups

The ROC curve helps in deciding where to draw the line between 'diseased' and 'disease-free'. The accuracy of the screening test is represented by the area under the ROC curve, where the curve is determined by multiple cut-off points of the trial test. The AUC obtained from the ROC curve in Figure 4 is 0.9361. According to Table 4, the screening test has an excellent accuracy.

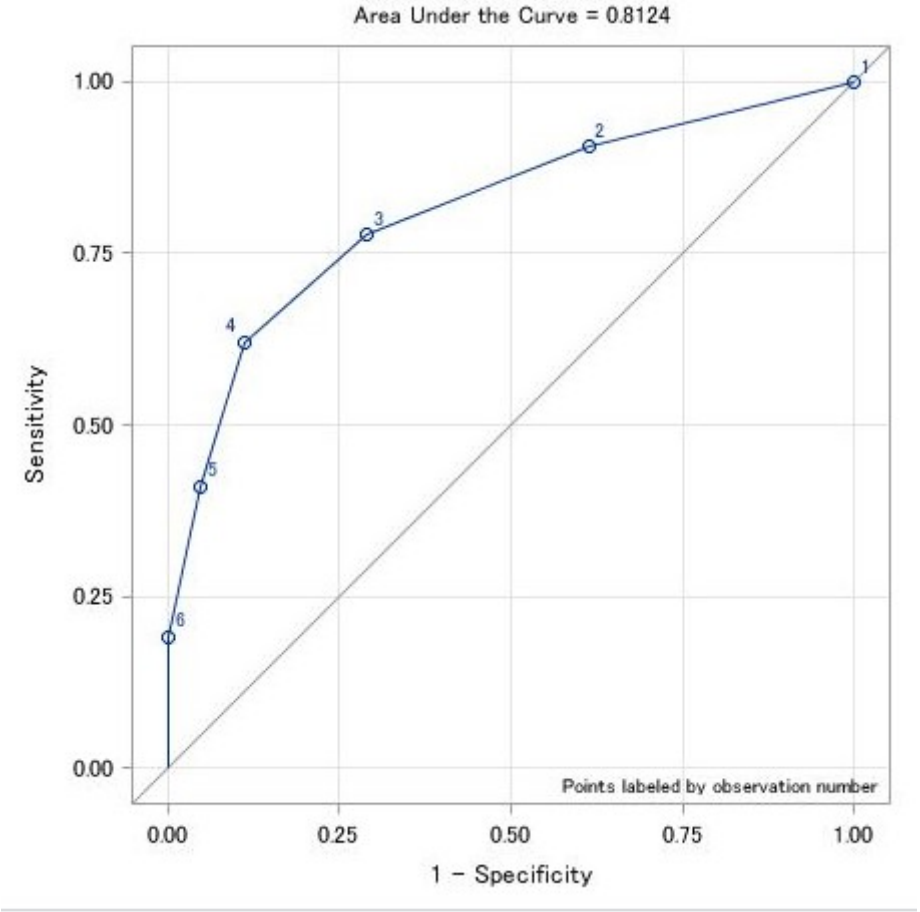


Figure 5: ROC curve for the obese group

Figure 5 shows the ROC curve that was constructed only from the obese group. The AUC is 0.8124 and according to Table 4 the trial test has a good accuracy.

4 Conclusion

The usefulness of sensitivity and specificity is limited since these two measures cannot be used to estimate the probability of the disease in an individual. In order to estimate this probability predictive values may be used. Both the positive and negative predictive values vary according to the prevalence of the disease.

More information about a screening test is given by the likelihood ratios. When the likelihood ratio of a screening test is known, the post-test probability of the disease can be computed. The most convenient tool, that is used to estimate this probability, is the Fagan's nomogram. Bayes theorem can also be used, but the calculations can be very tedious.

In this research report, the focus was on evaluating the accuracy of a screening test to discriminate between the individuals with or without disease. The accuracy of the screening test in the practical example was 0.9361. It can be concluded that the screening test can be used as a proper substitute to

the coronary angiography test.

References

[1] Anthony K Akobeng. Understanding diagnostic tests 1: sensitivity, specificity and predictive values. *Acta Paediatrica*, 96(3):338–341, 2007.

[2] Anthony K Akobeng. Understanding diagnostic tests 2: likelihood ratios, pre-and post-test probabilities and their use in clinical practice. *Acta Paediatrica*, 96(4):487–491, 2007.

[3] Anthony K Akobeng. Understanding diagnostic tests 3: receiver operating characteristic curves. *Acta Paediatrica*, 96(5):644–647, 2007.

[4] Douglas G Altman and J Martin Bland. Diagnostic tests. 1: Sensitivity and specificity. *British Medical Journal*, 308(6943):1552, 1994.

[5] William C Black and H Gilbert Welch. Screening for disease. *American Journal of Roentgenology*, 168(1):3–11, 1997.

[6] Jonathan J Deeks and Douglas G Altman. Diagnostic tests 4: likelihood ratios. *British Medical Journal*, 329(7458):168–169, 2004.

[7] Noel Espallardo. Decisions on diagnosis in family practice: Use of sensitivity, specificity, predictive values and likelihood ratios. *Asia Pacific Family Medicine*, 2(4):229–232, 2003.

[8] A Halkin, J Reichman, M Schwaber, O Paltiel, and M Brezis. Likelihood ratios: getting diagnostic testing into perspective. *Monthly Journal of the Association of Physicians*, 91(4):247–258, 1998.

[9] Abdul Ghaaliq Lalkhen and Anthony McCluskey. Clinical tests: sensitivity and specificity. *Continuing Education in Anaesthesia Critical Care & Pain*, 8(6):221–223, 2008.

[10] Tze-Wey Loong. Understanding sensitivity and specificity with the right side of the brain. *British Medical Journal*, 327(7417):716–719, 2003.

[11] Ramanitharan Manikandan and Lalgudi N Dorairajan. How to appraise a diagnostic test. *Indian Journal of Urology*, 27(4):513, 2011.

[12] Charles Metz. Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, 8(4):283–298, 1978.

[13] Rajul Parikh, Annie Mathai, Shefali Parikh, G Chandra Sekhar, and Ravi Thomas. Understanding and using sensitivity, specificity and predictive values. *Indian Journal of Ophthalmology*, 56(1):45, 2008.

[14] Johannes B Reitsma, Afina S Glas, Anne WS Rutjes, Rob JPM Scholten, Patrick M Bossuyt, and Aeilko H Zwinderman. Bivariate analysis of sensitivity and specificity produces informative summary measures in diagnostic reviews. *Journal of Clinical Epidemiology*, 58(10):982–990, 2005.

[15] Richard Simon. Sensitivity, specificity, ppv, and npv for predictive biomarkers. *Journal of the National Cancer Institute*, 107(8), 2015.

[16] Wen Zhu, Nancy Zeng, and Ning Wang. Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas implementations. *NESUG proceedings: health care and life sciences, Baltimore, Maryland*, 19, 2010.

Appendix

The SAS software code that generated the data that was used to compute the 2×2 contingency table

```
proc format;
value ExpFmt 1='Positive'
0='Negative';
value RspFmt 1='Positive'
0='Negative';
run;
data coronary;
input Exposure Response Count;
label Response='CORONARY ANGIOGRAPHY';
label Exposure='SCREENING TEST';
datalines;
0 0  179
0 1  1   //here different values for different categories were used//
1 0  41
1 1 19
;
proc sort data=coronary;
by descending Exposure descending Response;
run;
```

Below is the SAS software code used when inputting the decision functions in Tables , 9 and 10.

```
*create the contingency table*;
proc freq data=coronary order=data;
format Exposure ExpFmt. Response RspFmt.;
tables Exposure*Response;
weight Count;
title 'Contingency table to summarize data for BMI category';
run;
```

Create the data set used for calculating 95% confidence interval. The SAS software output was shown in Figure 17.

```
data1;
```

```
tp = 19;
fn = 1;    //here different values for different categories were used//
tn = 179;
fp = 41;
proc sql;
create table cont1 as select tp as TP, fp as FP, fn as FN, tn as TN, tn+tp as TNTP, fn+fp as FNF
from data1;
proc print;
quit;
proc transpose data=cont1 out=t_data;
var TP FN TN FP TNTP FNFP;
proc print;
run;
data confint (drop = _name_ col1);
length group $20;
set t_data;
count=col1;
if _name_ = "TNTP" then do;
group="Accuracy";
response = 0;
output;
end;
else if _name_ = "FNFP" then do;
group="Accuracy";
response = 1;
output;
end;
else if _name_ = "TP" then do;
group="Sensitivity";
response = 0;
output;
end;
else if _name_ = "FN" then do;
group="Sensitivity";
```

```
response = 1;
output;
end;
else if _name_ = "TN" then do;
group="Specificity";
response = 0;
output;
end;
else if _name_ = "FP" then do;
group="Specificity";
response = 1;
output;
end;
proc print;
run;
```

The SAS software code that generated the data that was used to compute the ROC curve

```
data dataroc;
input bmi diseased n;
cards;
25 1 51
26 2 48
27 4 49
28 6 50
29 7 42
30 33 57
31 45 65
32 55 66
33 74 78
34 77 80
35 25 25
36 10 10
37 8 8
38 12 12
39 12 12
```

Initialization strategies in parameter estimation of Gaussian mixtures

Quintine Mkhondo 13244630

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Mr S.M Millard

Department of Statistics, University of Pretoria

31 October 2017

Abstract

In this study the main focus is initialization strategies in parameter estimation of the Gaussian mixture. TheIn this study the focus is on initialization strategies in parameter estimation of the Gaussian mixture model. The Expectation Maximization (EM) algorithm is used to estimate parameters of the mixture model. Two initialization strategies for the EM algorithm will be considered and reports on the absolute bias of the means, mixing proportions and number of iterations until convergence will be made.

Declaration

I, *Quintine Mkhondo,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Quintine Mkhondo

_ _

Sollie M. Millard

_ _

Date

Acknowledgements

I acknowledge the finacial support from the Centre for Artificial Intelligence Research, Meraka Institute, CSIR and Investec for the financial support in the form of a postgraduate bursary. I would also like to extend special thank you to his parents Stanley and Angel Mkhondo for the continued love and support for my academics and Sollie Millard for his time, patience and assistance as a supervisor for this report, he has ignited my interest in data science and machine learning.

Contents

List of Figures

1 Introduction

A mixture of distribution is a probabilistic model used in latent class modelling. A model of this nature is useful in instances where only a data sample is available and no other information about the data set is given; the model can then be used to give a general structure about the given data set. The model uses the data sample and groups the observations into different clusters with different parametric form. Mixtures of distributions are semiparametric since the of groups and observations are unknown, they compromise a finite or infinite number of components of different distributional types that can describe different features of data [8]. The model works where the observations in the sample are assumed to be independent and identically distributed and belong to a probability density function thereby providing a flexible, parametric framework for statistical modeling and analysis [8]. It is called mixtures of distributions because the probability density function consists of different density functions which represent the different clusters. The clusters are weighted, therefore they can be assigned using probabilities. Mixtures of distributions can be homogeneous or inhomogenous, however for in this study the will focus on homogeneous models [12].

Mixture distributions arises when unobserved heterogeneity is present in a population for which a particular random characteristic is observed [5]. In this research report the homogeneous mixture distribution that will be focused on is the Gaussian mixture model. A multivariate Gaussian mixture model with K components has the following probability density function:

$$p(\boldsymbol{x}|\Theta) = \sum_{k=1}^{K} p(\boldsymbol{x}|\boldsymbol{\theta_k})\pi_k = \sum_{k=1}^{K} p(\boldsymbol{x}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})\pi_k \tag{1}$$

where π_k represents the weight probability of the k^{th}component (with the constraints $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$, $p(\boldsymbol{x}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$ represents the k^{th}component-conditional Gaussian density function, $\boldsymbol{\mu_k}$ and $\boldsymbol{\Sigma_k}$ represent the mean and covariance of the k^{th} Gaussian density function such that $\boldsymbol{\theta}_k = \{\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}\}$ and $\boldsymbol{\Theta} = (\theta_1, \theta_2, ..., \theta_K, \pi_1, \pi_2, ..., \pi_K)$ represents the complete set of parameters of the model.

In this study the EM algorithm will be used to approximate the likelihood estimates of parameters of Gaussian mixture model [10]. Point of convergence of the EM algorithm is known to be dependent on where the process start, therefore two initialization techniques will be explored; the convergence rate of the two strategies will also be considered. The first strategy is by [2] which selects a starting point randomly. The other strategy separates the given sample in to three quatiles and uses the average of Q_1 and Q_3 as the initial parameters.

2 Background Theory/Literature Review

In papers focused on mixture models there is always a discussion of how the log of the maximum likelihood function can be maximized to obtain components of the mixture model; the same method extends to mixtures of Gaussians.

Frank Picard's paper gives fundamentals of the mixture models. This source outlines the importance of mixture models in cluster analysis. Explicit formulas that can be used to estimate the parameters of the Gaussian mixture model are provided; these formulas can be used in direct application to successfully fit the Gaussian mixture via the EM algorithm. The basic idea behind the EM algorithm is discussed like how the method requires iterative procedures that lead us to straight forward estimates (through convergence), its general properties and limitations. The source also provides detailed examples of how the EM algorithm can be applied to obtain the Gaussian mixture model .

N.E Day's paper provides comprehensive writing on the components that make up the Gaussian mixture model. The source discusses the impact of adding more components to the maximum likelihood and how the performance of the mixture model will be affected. The source also discusses how the Bayesian approach appears to be greatly inferior to maximum likelihood which is used by the EM algorithm in the multivariate case expect for univariate case[4] .

Bishop's book provides a comprehensive study on how to perform the EM algorithm for two components [2]. Practical examples are provided to demonstrate how the procedure will be performed to obtain the means and variances of the mixture of Gaussian.

3 Gaussian mixtures

Machine learning is a field in Statistics that solves problems by either a supervised learning technique or an unsupervised one. Supervised learning uses classification and regression models to replicate the results of the input independently. Therefore the system is then able to work on its own and perform tasks that the operation has 'taught' the machine; unsupervised learning uses clustering and association models in order to find a general structure of the data set. Supervised learning in machine learning approximates the output of the data of the input data set, while unsupervised learning provides output that gives a general description the input data set. Tools to solve problems in unsupervised learning are the Gaussian mixture model, Hidden Markov Model, Hierarchical models and Neural Networks. However the focus of this study is the Gaussian mixture.

3.1 Gaussian mixture models

The Gaussian mixture model is semiparametric model meaning i.e it is a statistical model made of both parametric and non parametric components. The focus of this study is a homogeneous mixture, the Gaussian mixture. In this report homogeneous mixture model will be investigated. Homogeneous mixture models are models where all its components have the same density function which is Gaussian. The model uses a probabilistic approach to cluster input data set from the normal distribution.
The Gaussian model falls under unsupervised learning as it provides a general structure of data set input. The Gaussian mixture model is useful in providing a general structure of the data set using different groups that represent different clusters of the data set. This clustering approach gives each group its own parametric form, observations found are found in each of the structures and are assumed to have sufficient characteristics to be put in the same group. Each group will be characterized by the different parametric forms of the Gaussian mixture model [12].

In Gaussian mixture models it is assumed that the sample data set to be used represent a sample with observations that are independent and identically distributed. The sample data originates from a population that is normally distributed, and can be described using a probability density function [6]. The probability function is made up the different clusters, all clusters can be identified with a probability density function. The joint probability function that make up the Gaussian mixture model consists of probability density functions of each of the clusters. The density functions that form the joint probability functions are weighted, the different density functions represents the clusters, the different weights can then be used as descriptive statistics for the input data set [12]. In order to obtain descriptive statistics from the Gaussian mixture model a strategy is used to derive the component densities of the data set. The Gaussian mixture model is then an effective tool that can be used to find descriptive analysis when the sample originates from the Gaussian distribution.

The Gaussian mixture model clusters the data set into K groups and the number of groups is assumed to be fixed. In this report we will assume that there are two groups ($k = 2$). The k groups are assumed to have large variance between then and the variance of the the observations within the groups is small. Suppose that our random variable of interest X in the data set is defined in the sample space $\Re^p$. The random variable can be defined to be any quantitative data the model can be applied in both the univariate or multivariate dimension [12]. Consider a vector $\boldsymbol{x} = (x_1, x_2, ..., x_n)$, this vector represents a random sample of size n. The Gaussian mixture model the clusters the n observations. Consider the realization of x_t and its probability density function $p(x_t)$, the probability density function is defined on

its appropriate measure on the sample space $\Re^p$ and the measure is chosen according to the context of the nature of the sample, which is Lesbesgue measure, counting measure or both measures [12].

Consider X our variable of interest from a sample size of n observations. The Gaussian mixture model is assumed to be derived from K different groups, the components of the k groups each have the same probability density function which is Gaussian. The K groups arise from the same parametric family $p(x_t, \psi)$, in other words ψ is the same for each group since the sample is assumed to originate from an i.i.d population and are all Gaussian distributed. The Gaussian mixture density of of x_t can be expressed as:

$$p(x_t;\theta) = \sum_{i=1}^{k} \pi_i p(x_i;\theta_i) \tag{2}$$

where π_i represents the weight of the k^{th} component of the probability density function and $\boldsymbol{\pi} = (\pi_1, \pi_2 ..., \pi_k)$ is a vector that represents the weights of the K clusters of the data set and $\sum_{i=1}^{k} \pi_i = 1$. The K mixture models will be characterized by θ_i. Then each $p(x_i;\theta_i)$ belongs can be characterized into the θ_i with probability π_i. Let $\boldsymbol{\psi} = (\pi_1, \pi_2, ..., \pi_k, \theta_1, \theta_2, ..., \theta_k)$ represent the complete set of parameters that make up the Gaussian model.

The Gaussian mixture model is used as a clustering technique in unsupervised learning. It's objective is to find homogeneous groups in a data set. The Gaussian mixture model finds the K different clusters which are initially hidden, this is done by fitting a K component Gaussian mixture density. Observations that belong to the same cluster as expected to have small variance, and the K different groups are expected to have high variance between them, in this way the groups are balanced. This optimizes the process and improves the chances of the model obtaining estimates that correspond to the true mean of the hidden group.

Multivariate Gaussian mixture

A multivariate Gaussian mixture model with K different components has the following form:

$$p(\boldsymbol{x}|\boldsymbol{\psi}) = \sum_{i=1}^{K} \pi_i p(\boldsymbol{x};\boldsymbol{\theta}_i) = \sum_{i=1}^{K} \pi_i p(\boldsymbol{x}_i|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{3}$$

π_i - represents the weight of each component in the entire Gaussian mixture model $\sum_{i=1}^{K} \pi_i = 1$, $p(\boldsymbol{x}_i|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ represents the Gaussian probability that calculates the probability the k^{th} component belongs to a particular cluster

$\boldsymbol{\mu}_k$ - the mean of the k^{th}Gaussian density component

$\boldsymbol{\Sigma}_k$ - the covariance matrix of the k^{th} Gaussian density component. Both $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ belong to the same

characteristic group (i.e $\boldsymbol{\theta}_k = \{\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}\}$)

$\boldsymbol{\psi}-$a vector that represents the complete set of parameters of the Gaussian mixture model, $\psi = (\pi_1, \pi_2..., \pi_K, \theta_1, \theta_2, ..., \theta_K)$

Consider the observation $\boldsymbol{x}$ from an input set of p-dimensions, then the conditional Gaussian density of the k^{th} component can be obtained using the following expression:

$$p(\boldsymbol{x}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-\frac{1}{2}} (\boldsymbol{x} - \boldsymbol{\mu}_k)\} \tag{4}$$

The number of components that make up the Gaussian mixture model can be estimated by using sophisticated selection models like the number of components of mixture models can be estimated using model selection methods like the Alkaike Information Criterion, Bayesian Information Criterion, cross validation and Bayesian methods. However for the purpose of this study the number of components in a Gaussian mixture model is assumed to be known.

Consider $\boldsymbol{x}_i$ to be the i^{th} observation from our given sample of size N, $i = \{1, 2, ..., N\}$. Then the probability that of the $\boldsymbol{x}_i^{th}$ observation belonging to component θ_k can be obtained using the Bayes' theorem:

$$\gamma(\theta_k) = p(\theta_k = 1|\boldsymbol{x}) = \frac{p(\theta_k = 1)p(\boldsymbol{x}|\theta_k = 1)}{\sum_{j=1}^{K} p(\theta_j = 1)p(\boldsymbol{x}|\theta_k = 1)}$$

$$= \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{5}$$

The quantity $\gamma(\theta_k)$ is the corresponding posterior probability once we have observed $\boldsymbol{x}$.

An unbiased estimate of $\gamma(\theta_k)$ is given by:

$$\hat{\theta}_{ik} = P(\theta_i = k|\boldsymbol{x}_i, \hat{\gamma}) = \frac{\hat{\pi}_k p(\boldsymbol{x_i}|\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)}{\sum_{j=1}^{K} \hat{\pi}_j p(\boldsymbol{x}_i|\hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j)} \tag{6}$$

the equation above is called the responsibility and it will be used to estimate the probability that the i^{th} observation belongs to component k. $\theta = \{\theta_1, ..., \theta_k\}$ represent of the different characteristics the K components are allocated. The Gaussian components that make up the Gaussian mixture model will be allocated into a group according to the corresponding responsibility. Components with highest responsibility matching a component will be allocated in the same cluster. This gives an indication of where the input data set is most likely been generated. The observations that are clustered using probabilities is an example of "soft-clustering".

3.2 Maximum likelihood

Finding the maximum likelihood estimates for the Gaussian mixture is an essential step to help obtain the components of the Gaussian mixture model. The EM algorithm is an iterative procedure used in this study to obtain parameter estimates of the Gaussian mixture model.

Consider the following data set of observations $\{x_1, x_2, ..., x_N\}$ from a sample of size N. It is required to fit a Gaussian mixture model for this sample. The assumption that the observations were drawn independently from the Gaussian distribution holds, then the Gaussian mixture model can be obtained.

The mixture of the likelihood function can be obtained in the following way:

$$p(\boldsymbol{X}|\boldsymbol{\psi}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\psi})$$

$$= \prod_{i=1}^{N} (\sum_{k=1}^{K} p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\pi_k)$$

The log of the likelihood function can be obtained in the following way:

$$logp(\boldsymbol{x}|\boldsymbol{\psi}) = log \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\psi})$$

$$= \sum_{i=1}^{N} log(\sum_{k=1}^{K} p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\pi_k)$$

Singularities are a problem when maximizing this function. Singularities occur when a Gaussian component 'run into' a data point, this brings rise to multiplicative factors when computing the likelihood function, the multiplicative factors belong to the other data points. This then causes the likelihood function of the overall Gaussian mixture model to go to zero. This problem is solved by seeking local maxima of the log-likelihood function that make sense. There are other ways in which singularities can be solved, for instance whenever a Gaussian component 'runs into' a data point it's mean can be re-defined to a randomly available value in the sample and the covariance matrix to a large variance.

Another worthwhile issue to consider when dealing with maximum likelihood estimators is the issue of identifiability. This occurs when component means are very close to each other. It is an important issue to consider when interpreting parameters that have been discovered by a model. This issue arises because the maximum likelihood solution provides solutions $K!$ and $K-1!$ different ways in which the components can be arranged. This means that in a space of a parameter value there are $K!-1$ data points that may give rise to exactly the same distribution. However this issue is not a problem since one solution is as good as the other.

3.3 EM algorithm

Expectation-maximization or EM is an iterative method that is elegant and powerful for finding solutions for models with latent variables. The algorithm performs iterations with the aim of obtaining parameter estimates. The estimates are obtained by approximating the maximum-likelihood estimates for the data with latent variables. In this report the EM algorithm will be used to find estimates of the Gaussian model.

Let $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N) \in \Re$ represent an independent and identically distributed sample with N unlabeled observations that originates from a Gaussian mixture model with K components. Therefore a Gaussian mixture density for an observed point $\boldsymbol{x}_i$ can be expressed as follows:

$$p(\boldsymbol{x}_i|\boldsymbol{\psi}) = \sum_{k=1}^{K} p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\pi_k$$

$\boldsymbol{\psi} = (\pi_1, \pi_2, ..., \pi_K, \theta_1, \theta_2 ..., \theta_K)$ represents the complete set of parameters for the Gaussian mixture model.

$\theta_1, \theta_2, ..., \theta_K$ represents the K different characterized clusters.

$\pi_1, \pi_2, ..., \pi_K$ represents the weights of the k clusters, $\pi_k \geq 0$ for $k = 1, 2, ..., K$ and the weights all up to 1.

The EM algorithm uses the maximum likelihood to estimate $\boldsymbol{\psi}$ which is the complete set of parameters. The following expression of the maximum likelihood estimator will be used to obtain the estimates:

$$p(\boldsymbol{x}|\boldsymbol{\Theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\Theta})$$

$$= \prod_{k=1}^{K} (\sum_{k=1}^{K} p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\pi_k)$$

However the expression above is very difficult to compute and the log likelihood function will be used to solve the issue. The log-likelihood estimator is an easier expression to compute; it can be expressed in the following way:

$$logp(\boldsymbol{x}|\boldsymbol{\Theta}) = log \prod_{k=1}^{K} p(\boldsymbol{x}_i|\boldsymbol{\Theta})$$

$$= \sum_{k=1}^{K} logp(\boldsymbol{x}_i|\boldsymbol{\Theta})$$

$$= \sum_{k=1}^{K} log(\sum_{k=1}^{K} p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\eta_k)$$

In order to obtain estimates from the log-likelihood function requires to be maximized with respect to $\boldsymbol{\psi}$. The equation used to find the maximum likelihood estimates is as follows:

$$\frac{\partial}{\partial \boldsymbol{\psi}} logp(\boldsymbol{x}|\boldsymbol{\psi}) = 0$$

Obtaining an explicit expression of the parameters is difficult since the logarithm contains the sum of the terms [12].

The Gaussian mixture model can be viewed can be viewed as incomplete data problem in unsupervised learning. The data set $\boldsymbol{x} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$ is considered incomplete data, the each observation in the data set can be $Z = \{Z_1, Z_2, ..., Z_N\}$. The objective is to find the K different components that are unknown, the K components make up the Gaussian mixture model.

The EM algorithm can be used as a special tool to obtain estimates from incomplete data by using the log-likelihood function. The labels of the data set (i.e $Z's$) are used to find out origin of the k^{th} component of the Gaussian mixture model. $\boldsymbol{Z}_i = (Z_{i1}, Z_{i2}, ..., Z_{im})$ is a vector of binary variables used to indicate whether the i^{th} observation belongs to the k^{th}component. Z_i takes the value 1 if the observation $\boldsymbol{x}_i$ belongs to the k^{th} observation and zero if otherwise. $\boldsymbol{Z}_i$ follows a multinomial distribution with parameters 1 and $\boldsymbol{\pi}$ where $\boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_K)$. The observed data vector $\boldsymbol{x}$ and the label $\boldsymbol{Z}$ form the complete data vector $(\boldsymbol{x}, \boldsymbol{Z})$.

A likelihood function for complete data is given by:

$$p(\boldsymbol{x}, \boldsymbol{Z}|\boldsymbol{\psi}) = \prod_{i=1}^{N} \prod_{k=1}^{K} (p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\pi_k)^{I\{Z_i=k\}}$$

$I\{Z_i = k\}$ represents an indicator function that is used to find where the latent variable belongs.$I\{Z_i = k\} = 1$ if an observation belongs to the k^{th}component and $I\{Z_i = k\} = 0$ if otherwise.

The log-likelihood function since it is easier to compute and can be expressed in the following equation:

$$logp(\boldsymbol{x}, \boldsymbol{Z}|\boldsymbol{\psi}) = \sum_{i=1}^{N} \sum_{k=1}^{K} Z_{ik} log(p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\pi_k)$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} Z_{ik} (logp(\boldsymbol{x}_i|\boldsymbol{\theta}_k) + log\pi_k)$$

How the EM algorithm works

The EM algorithm performs iterations through the sample of X random variables and obtains the max-

imum likelihood estimates of $\hat{\psi}$. The iteration is performed through two steps namely the Expectation step and the Maximization step. The procedure produces a sequence of maximum likelihood estimates starting from $t = 0$ and it is repreated until the sequence reaches a point of convergence. A convergence criteria is decided by the investigator, the criteria is set to a value such that the difference between the maximum likelihood estimate at t and $t - 1$ is very small. The algorithm uses all its observation in the data set in each iteration.

The first step is to choose initial values for the complete set of parameters namely the means, co-variances, and mixing coefficients represented by $\hat{\boldsymbol{\psi}}(0)$. After initial values have been established, the E-step follows. The E-step uses conditional expectation of the complete log-likelihood function. This is computed using the observed data and recent estimates. The name of the expectation function is called the objective function. The objective function uses recently set or computed estimates to evaluate the posterior probabilities of the Gaussian mixture model.

The M-step follows after the E-step. Re-estimation the parameters of $\boldsymbol{\psi}(t)$ is computed in this step, namely the means, co-variances and mixing probabilities. This step involves the maximization of the objective function introduced in the E-step, provided that the constraints that all the weights are positive and that the weights sum to 1 is met.

The EM algorithm increases the maximum likelihood monotonically with each iteration. The objective function increases as the log-likelihood increases, the likelihood increases until a point of convergence. However, convergence to a global maxima is not guaranteed. The convergence rate depends on the type of initialization strategy employed.

The general EM algorithm

Below are the steps of how the EM algorithm is performed:

1. Initialize values for the parameters at $t = 0$ i.e $\hat{\boldsymbol{\psi}}(0)$

2. E-step:

Compute the objective function for $t \geq 0$ using the following objective function $Q(\boldsymbol{\psi}, \hat{\boldsymbol{\psi}}(t)) = E[logp(\boldsymbol{x}, \boldsymbol{Z}|\boldsymbol{\Theta})|\boldsymbol{x}, \hat{\boldsymbol{\psi}}(t))$

3. M-step:

Obtain the maximum likelihood estimates of the parameters by maximizing the objective function such that $\hat{\boldsymbol{\psi}}(t+1) = \underset{\Theta}{argmax} Q(\boldsymbol{\psi}, \hat{\boldsymbol{\psi}}(t))$ for $t \geq 0$

4. Repeat the E-step and M-step until the convergence to the global maxima.

E-step

Initial values for the parameters are set at $t = 0$. The initial values are used in the E-step to compute the conditional expectation of the log-likelihood $logp(\boldsymbol{x}, \boldsymbol{Z}|\boldsymbol{\Theta}))$ for $t \geq 0$. The objective or Q-function

can be expressed as follows for $t \geq 0$:

$Q(\boldsymbol{\psi}, \hat{\boldsymbol{\psi}}(t)) = E[logp(\boldsymbol{x}, \boldsymbol{Z}|\boldsymbol{\psi})|\boldsymbol{x}, \hat{\boldsymbol{\psi}}(t))$

where $\hat{\boldsymbol{\psi}}(t)$ represents the complete set of parameters provided by the maximum likelihood estimates at time t. The formula used to obtain the conditional expectation can be simplified since the log-likelihood fucntion of the complete set of parameters is a linear function of Z_{ik}, this simplification allows the user to make the computation easier and can be represented as follows:

$$E[Z_{ik}|\boldsymbol{x}, \hat{\boldsymbol{\psi}}(t)] = 1 \times P(Z_{ik} = 1|\boldsymbol{x}, \hat{\boldsymbol{\psi}}(t)) + 0 \times P(Z_{ik} = 0|\boldsymbol{x}, \hat{\boldsymbol{\psi}}(t))$$

$$= P(Z_{ik} = 1|\boldsymbol{x}, \hat{\boldsymbol{\psi}}(t))$$

$$= P(Z_{ik} = k|\boldsymbol{x}_i, \hat{\boldsymbol{\psi}}(t))$$

since $Z_{ik} = 1 \iff Z_i = k$.

With Z_{ik}, $i = 1, 2, ..., N$ and $k = 1, 2, ..., K$.

Equation **??** represents the posterior probability of the i^{th} observation is the probability that the observation i was generated by the k^{th} component of the density model. The E-step is used to estimate Z_{ik} i.e the labels of the observations for $t \geq 0$. An unbiased estimator posterior probability can be derived using Bayes' Theorem in the following way:

$$\hat{Z}_{ik}(t) = P(Z_i = k|\boldsymbol{x}_i, \hat{\boldsymbol{\psi}}(t))$$

$$= \frac{\hat{\pi}_k p(Z_{ik} = 1|\boldsymbol{x}, \hat{\boldsymbol{\theta}}_t(t))}{\sum_{j=1}^{K} \hat{\pi}_j(t) p(\boldsymbol{x}_i|\hat{\theta}_j(t))}$$

Equation **??** is called the responsibility which determines where component k for observation i belongs. From above it can be deduced that the E step performs soft assignment of observations to their respective component of the mixture model [12]. The E-step uses the latest estimates to compute the relative densities of observations for each component model [12].

The objective function can be derived in the following way:

$$Q(\boldsymbol{\psi}, \hat{\boldsymbol{\psi}}(t)) = E[log(\boldsymbol{x}, \boldsymbol{Z}|\boldsymbol{\psi})|\boldsymbol{x}, \hat{\boldsymbol{\psi}}(t)]$$

$$= \sum_{k=1}^{K} \sum_{i=1}^{N} \hat{Z}_{ik}(t)(logp(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}_k(t)) + log\hat{\pi}_k(t))$$

$$=\sum_{k=1}^{K}\sum_{i=1}^{N} P(Z_i = k|\boldsymbol{x}_i, \hat{\boldsymbol{\psi}}(t))(logp(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}(t)) + log\hat{\pi_k}(t))$$

$$=\sum_{k=1}^{K}\sum_{i=1}^{N} log\hat{\pi_k}(t)P(Z_i = k|\boldsymbol{x}_i, \hat{\boldsymbol{\psi}}(t))+ \sum_{k=1}^{K}\sum_{i=1}^{N} (logp(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}(t))P(Z_i = k|\boldsymbol{x}_i, \hat{\boldsymbol{\psi}}(t))$$

M-step

In this step estimates obtained from the E-step will now be used as estimates. Maximization of the objective function will provide new estimates for the unknown parameters. This step re-estimates new parameters for $\hat{\psi}$ at time $t+1$. The expression that will be used to perform maximization of the parameters at time $t+1$ is the following:

$$\hat{\boldsymbol{\psi}}(t+1) = \underset{\psi}{argmax} Q(\boldsymbol{\psi}, \hat{\boldsymbol{\psi}}(t)) \tag{7}$$

The weights and component densities of the Gaussian mixture model are independent. Maximization of equation 7 with respect to the weights and densities provide independent new estimates for the Gaussian mixture model. The new estimates $\hat{\pi_k}(t+1)$ and $\hat{\theta}(t+1)$ are independent because they were maximized independently[1]. This step computes the weights and the component densities of the Gaussian mixture model, and computation involves for k components that make up the Gaussian mixture model.

If the Z_{ik} label for the components are known then the weights of the components can be calculated in the following way:

$$\hat{\pi_k} = \sum_{i=1}^{N} \frac{Z_{ik}}{N}$$

for all k components in the Gaussian mixture model Z_{ik} is an indicator variable that is 1 if $Z_i = k$ and 0 if otherwise. $\hat{\pi_k}$ can then be interpreted as the proportion of observations that developed from the k^{th}component density of the mixture.

However, the components of the mixture models are unknown meaning that the formula above can not be applied. The estimate of the mixing proportions will then have to be approximated by an iterative process. $\hat{\pi_k}(t+1)$ can be obtained by solving the following equation given that the constrains that $\sum_{k=1}^{K} \hat{\pi_k} = 1$ hold:

$$\frac{\partial Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}(t))}{\partial \hat{\pi_k}(t)} = 0$$

Lagrange multiplier λ is used in order to take the constrains into account, consider the following equations [1]:

$$\frac{\partial}{\partial \hat{\pi_k}(t)} \left[\sum_{k=1}^{K} \sum_{i=1}^{N} \hat{Z}_{ik}(t)(logp(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}_k(t)) + log\hat{\pi_k}(t)) + \lambda \left(\sum_{k=1}^{K} \hat{\pi_k}(t) - 1 \right) \right] = 0$$

$$\therefore \sum_{i=1}^{N} \frac{1}{\hat{\pi_k}(t)} \hat{Z}_{ik}(t) + \lambda = 0$$

Adding both sides by k representing the number on components, $\lambda = -N$, the following equation is obtained:

$$\hat{\pi_k}(t) = \sum_{i=1}^{N} \frac{\hat{Z}_{ik}(t)}{N} \tag{8}$$

all k components.

8 will be computed iteratively, and the weights of the k components in the data set will be determined. The updated weights of the components will be obtained by using the following expression :

$$\hat{\pi_k}(t+1) = \sum_{i=1}^{N} \frac{\hat{Z}_{ik}(t)}{N}$$

$$= \sum_{i=1}^{N} \frac{P(Z_i = k|\boldsymbol{x}_i, \hat{\boldsymbol{\psi}}(t))}{N}$$

for all k components of the Gaussian mixture model.

The expression above represent the posterior probabilities that can be used obtain the K estimates of the mixing probabilities at time $t+1$ that has been calculated over all the N observations [12].

Since a maximized expression of $\hat{\pi_k}$ has been obtained, an expression for $\psi = (\theta_1, \theta_2, ..., \theta_K)$ that represents the component densities need to be determined. The objective function $\partial Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}(t))$ need to be maximized with respect to $\hat{\psi}(t)$. The expression for the maximized objective function can obtained by finding the solution of the equation:

$$\frac{\partial Q(\boldsymbol{\psi}, \hat{\boldsymbol{\psi}}(t))}{\partial \hat{\epsilon}(t)} = \frac{\partial}{\partial \hat{\epsilon}(t)} \left[\sum_{k=1}^{K} \sum_{i=1}^{N} \hat{Z}_{ik}(t) \left(logp(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}_k(t)) + log\hat{\pi_k}(t) \right) \right]$$

$$= \sum_{k=1}^{K} \sum_{i=1}^{N} \hat{Z}_{ik}(t) \frac{\partial logp\left(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}_k(t)\right)}{\partial \hat{\theta}(t)}$$

.

$$= 0$$

The solution of the maximized objective function of the Gaussian mixture model exists in a closed form. This makes it easier for the M-step to use the responsibilities obtained from the E-step to obtain new and updated estimates. The iteration of the E and the M-step is performed iteratively until the sequence of the log-likelihood converge, this happens when there is an insignificant change in the log-likelihood function from time t and $t+1$. This change is represented by a stopping criterion and the iteration will then stop when the criteria is satisfied.

Univariate Gaussian mixture models

The general principle of EM algorithm will be used to obtain k components that make up a univariate Gausssian mixture model. The multivariate expression of the Gaussian mixture model will be presented in the next section (as demonstrated in [12] and [13]).

Consider the data set $x = (x_1, x_2, ..., x_N)$ represents of independent and identically distributed unlabled observations of sample size N. The observed variable x originates from a univariate Gaussian mixture model that consists of K components. x_i can be obtained using the following conditional probability function:

$$p(x_i|\boldsymbol{\psi}) = \sum_{k=1}^{K} p(x_i|\theta_k)\pi_k \tag{9}$$

Equation 9 can be used to find ψ, which is the complete set of parameters which make up the univariate Gaussian mixture model. The complete set consists the means and variances of the k different components that make up the Guassian mixture model. In addition the weights the k component densities are all positive add up to 1. The components in the data set x are univariate Gaussian distributed i.e $X_i \sim N(\mu_k, \hat{\sigma}_k)$ for $i = 1, 2, ..., N$ and $k = 1, 2, ..., K$. The component-conditional densities for all the k components of the univariate Gaussian mixture model can be obtained in the following way:

$$p(x_i|\theta_k) = p(x_i|\mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi\sigma_k^2}} exp\left(-\frac{1}{2\pi\sigma_k^2}(x_i - \mu_k)^2\right)$$

E-step

The E-step calculates the responsibilites of each of the N observations. This is done for all k components that make up the Gaussian mixture model.

$$\hat{Z}_{ik}(t) = P(Z_i = k|x_i, \hat{\boldsymbol{\Theta}}(t))$$

$$= \frac{\hat{\pi_k}(t)p\left(x_i, \hat{\mathbf{\Theta}}(t)\right)}{\sum_{j=1}^{K} \hat{\pi}_j(t)p\left(x_i|\hat{\theta}_j(t)\right)}$$

$$= \frac{\hat{\pi_k}\left[\frac{1}{\sqrt{2\pi\hat{\sigma}_k^2}}exp\left(-\frac{1}{2\sigma_k^2}(x_i-\hat{\mu}_k)^2\right)\right]}{\sum_{j=1}^{K} \hat{\pi}_j(t)\left[\frac{1}{\sqrt{2\pi\hat{\sigma}_j^2}}exp\left(-\frac{1}{2\sigma_j^2}(x_i-\hat{\mu}_j)^2\right)\right]}$$

Equation ?? holds for $i = 1, 2, ..., N$ and $k = 1, 2, ..., K$ and $t \geq 0$.

M-step

The M-step maximizes the objective function. Partial derivatives of the objective function with respect to the parameters is computed in this step i.e partial derivatives of $\boldsymbol{\psi} = ((\mu_1, \sigma_1^2), (\mu_2, \sigma_2^2), ..., (\mu_K, \sigma_K^2), \pi_1, \pi_2, ..., \pi_K)$. The equations will be set to zero so that the roots of the equations are found. This stage is at time $t+1$ of the iteration, and the estimates of the mixing probabilities can be computed using the following equation:

$$\hat{\pi}_k(t+1) = \sum_{i=1}^{N} \frac{\hat{Z}_{ik}}{N}$$

$$= \sum_{i=1}^{N} \frac{P(Z_i = k|x_i, \hat{\boldsymbol{\psi}}(t))}{N}$$

This is valid for $i = 1, 2, ..., N$ and $k = 1, 2, ..., K$ and $t \geq 0$.

The M-step uses results obtained from the E-steps to obtain new or updated estimates. In order to find the parameters for the specific component densities at time $t+1$ of the iteration. The component parameters for each the component densities can be represent with $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_K) = ((\mu_1, \sigma_1^2), (\mu_2, \sigma_2^2), ..., (\mu_K, \sigma_K^2))$, in order to obtain the parameter estimates, the objective function needs to be differentiated partially with respect to the μ_i or σ_i^2 for $i = 1, 2, ..., k$. The partial derivatives of the objective function can be obtained by finding the roots of the following function:

$$\frac{\partial Q(\boldsymbol{\psi}, \hat{\boldsymbol{\psi}}(t))}{\partial \hat{\boldsymbol{\theta}}(t)} = \sum_{k=1}^{K}\sum_{i=1}^{N} \hat{Z}_{ik}(t)\frac{\partial logp\left(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}_k(t)\right)}{\partial \hat{\boldsymbol{\theta}}(t)} = 0 \tag{10}$$

An expression of the component-conditional density function is as follows:

$$logp\left(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}_k(t)\right) = logp\left(x_i|\hat{\mu}_k(t), \hat{\sigma}_k^2(t)\right)$$

$$= log\left[\frac{1}{\sqrt{2\pi}\hat{\sigma}_k^2(t)}exp\left(-\frac{1}{2\sigma_k^2(t)}(x_i-\hat{\mu}_k(t))^2\right)\right]$$

$$= log\left(\frac{1}{\sqrt{2\pi}}\right) + log\left(\frac{1}{\hat{\sigma}_k^2(t)}\right) - \frac{(x_i - \hat{\mu}_k(t))^2}{2\hat{\sigma}_k^2(t)} \tag{11}$$

The partial derivative of 11 with respect to $\hat{\mu}_k(t)$ and $\sigma_k^2(t)$ yield the following equations respectively:

$$\frac{\partial logp\left(\boldsymbol{x}|\hat{\boldsymbol{\theta}}_k(t)\right)}{\partial\hat{\mu}_k(t)} = \frac{(x_i - \hat{\mu}_k(t))^2}{\hat{\sigma}_k^2(t)}$$

and

$$\frac{\partial logp\left(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}_k(t)\right)}{\partial\hat{\sigma}_k^2(t)} = -\frac{1}{2\hat{\sigma}_k^2(t)} + \frac{(x_i - \hat{\mu}_k(t))^2}{2\hat{\sigma}_k^4(t)}$$

Using 11 i.e the component-conditional density function for partial derivative of $\partial\hat{\boldsymbol{\theta}}(t)$ can re-written in the following way:

$$\frac{\partial Q(\boldsymbol{\psi}, \hat{\boldsymbol{\psi}}(t))}{\partial\hat{\boldsymbol{\theta}}(t)} = \sum_{k=1}^{K}\sum_{i=1}^{N} \hat{Z}_{ik}(t)\frac{\partial logp\left(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}_k(t)\right)}{\partial\hat{\boldsymbol{\theta}}(t)}$$

$$= \sum_{k=1}^{K}\sum_{i=1}^{N} \hat{Z}_{ik}(t)\frac{\partial}{\partial\hat{\boldsymbol{\theta}}(t)}\left[log\left(\frac{1}{\sqrt{2\pi}}\right) + log\left(\frac{1}{\hat{\sigma}_k^2(t)}\right) - \frac{(x_i - \hat{\mu}_k(t))^2}{2\hat{\sigma}_k^2(t)}\right] \tag{12}$$

12 Can be used to solve the partial derivatives of $\hat{\mu}_k(t)$ and $\hat{\sigma}_k^2(t)$ are given below respectively:

$$\sum_{i=1}^{N} \hat{Z}_{ik}(t)\frac{\partial}{\partial\hat{\mu}_k(t)}\left[log\left(\frac{1}{\sqrt{2\pi}}\right) + log\left(\frac{1}{\hat{\sigma}_k^2(t)}\right) - \frac{(x_i - \hat{\mu}_k(t))^2}{2\hat{\sigma}_k^2(t)}\right] = 0$$

and

$$\sum_{i=1}^{N} \hat{Z}_{ik}(t)\frac{\partial}{\partial\hat{\sigma}_k^2(t)}\left[log\left(\frac{1}{\sqrt{2\pi}}\right) + log\left(\frac{1}{\hat{\sigma}_k^2(t)}\right) - \frac{(x_i - \hat{\mu}_k(t))^2}{2\hat{\sigma}_k^2(t)}\right] = 0$$

The partial derivative for $\hat{\mu}_k(t)$ can be obtained in the following way:

$$\sum_{i=1}^{N} \hat{Z}_{ik}(t)\frac{\partial}{\partial\hat{\mu}_k(t)}\left[log\left(\frac{1}{\sqrt{2\pi}}\right) + log\left(\frac{1}{\hat{\sigma}_k^2(t)}\right) - \frac{(x_i - \hat{\mu}_k(t))^2}{2\hat{\sigma}_k^2(t)}\right] = 0$$

$$. \quad \sum_{i=1}^{N} \hat{Z}_{ik}(t)\frac{(x_i - \hat{\mu}_k(t))^2}{2\hat{\sigma}_k^2(t)} = 0$$

$$\sum_{i=1}^{N} \hat{Z}_{ik}(t)x_i - \sum_{i=1}^{N} \hat{Z}_{ik}(t)\hat{\mu}_k(t) = 0$$

$$\sum_{i=1}^{N} \hat{Z}_{ik}(t)x_i = \sum_{i=1}^{N} \hat{Z}_{ik}(t)\hat{\mu}_k(t)$$

$$\hat{\mu}_k(t) = \frac{\sum_{i=1}^{N} \hat{Z}_{ik}(t)x_i}{\sum_{i=1}^{N} \hat{Z}_{ik}(t)} \tag{13}$$

The partial derivative for $\hat{\sigma}_k(t)$ can be obtained in the following way:

$$\sum_{i=1}^{N} \hat{Z}_{ik}(t) \frac{\partial}{\partial \hat{\sigma}_k^2(t)} \left[log\left(\frac{1}{\sqrt{2\pi}}\right) + log\left(\frac{1}{\hat{\sigma}_k^2(t)}\right) - \frac{(x_i - \hat{\mu}_k(t))^2}{2\hat{\sigma}_k^2(t)} \right] = 0$$

$$\sum_{i=1}^{N} \hat{Z}_{ik}(t) \left[-\frac{1}{2\hat{\sigma}_k^2(t)} + \frac{(x_i - \hat{\mu}_k(t))^2}{2\hat{\sigma}_k^4(t)} \right] = 0$$

$$-\frac{\sum_{i=1}^{N} \hat{Z}_{ik}(t)}{2\hat{\sigma}_k^2(t)} + \frac{\sum_{i=1}^{N} \hat{Z}_{ik}(t)(x_i - \hat{\mu}_k(t))^2}{2\hat{\sigma}_k^4(t)} = 0$$

$$\frac{\sum_{i=1}^{N} \hat{Z}_{ik}(t)}{2\hat{\sigma}_k^2(t)} = \frac{\sum_{i=1}^{N} \hat{Z}_{ik}(t)(x_i - \hat{\mu}_k(t))^2}{2\hat{\sigma}_k^4(t)}$$

$$\hat{\sigma}_k^2(t) = \frac{\sum_{i=1}^{N} \hat{Z}_{ik}(t)(x_i - \hat{\mu}_k(t))^2}{\sum_{i=1}^{N} \hat{Z}_{ik}(t)} \tag{14}$$

The expressions in 13 and 14 are used at the M-step of the iteration i.e time $t+1$, therefore the following expressions of the mean and variance are realized respectively:

$$\hat{\mu}_k(t+1) = \frac{\sum_{i=1}^{N} \hat{Z}_{ik}(t)x_i}{\sum_{i=1}^{N} \hat{Z}_{ik}(t)}$$

for all k components of the univariate Guassian mixture model and $t \geq 0$.

$$\hat{\sigma}_k^2(t+1) = \frac{\sum_{i=1}^{N} \hat{Z}_{ik}(t)(x_i - \hat{\mu}_k(t))^2}{\sum_{i=1}^{N} \hat{Z}_{ik}(t)}$$

for all k components of the univariate Guassian mixture model and $t \geq 0$.

Four steps of EM algorithm for Univariate Gaussian mixture models

1. Initialize values of the unknown parameters at $t = 0$ i.e initialize $\hat{\mu}_k(0)$ and $\hat{\sigma}_k^2(0)$ and $\hat{\pi}_k(0)$, for

all k components

2. E-step:

Responsibilities that will be used to estimate the posterior probabilities at time step t with $t \geq 0$ are computed in this step using

$$\hat{Z}_{ik}(t) = \frac{\hat{\pi}_k(t) p\left(x_i, \hat{\boldsymbol{\psi}}(t)\right)}{\sum_{j=1}^{K} \hat{\pi}_j(t) p\left(x_i | \hat{\theta}_j(t)\right)} = \frac{\hat{\pi}_k \left[\frac{1}{\sqrt{2\pi}\hat{\sigma}_k^2} exp\left(-\frac{1}{2\sigma_k^2}(x_i - \hat{\mu}_k)^2\right)\right]}{\sum_{j=1}^{K} \hat{\pi}_j(t) \left[\frac{1}{\sqrt{2\pi}\hat{\sigma}_j^2} exp\left(-\frac{1}{2\sigma_j^2}(x_i - \hat{\mu}_j)^2\right)\right]}$$

for all N observations and k components in the sample.

3. M-step:

Obtain the maximum likelihood estimates by maximizing the objective function at time step $t+1$ with $t \geq 0$ using

$$\hat{\pi}_k(t+1) = \sum_{i=1}^{N} \frac{\hat{Z}_{ik}}{N}$$

for all k components

$$\hat{\mu}_k(t+1) = \frac{\sum_{i=1}^{N} \hat{Z}_{ik}(t) x_i}{\sum_{i=1}^{N} \hat{Z}_{ik}(t)}$$

for all k components

and

$$\hat{\sigma}_k^2(t+1) = \frac{\sum_{i=1}^{N} \hat{Z}_{ik}(t)(x_i - \hat{\mu}_k(t))^2}{\sum_{i=1}^{N} \hat{Z}_{ik}(t)}$$

for all k components

4. Repeat the E-step and M-step until convergence to the global maxima.

Multivariate Gaussian mixture model

Let $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N) \in \Re$ represent an independent and identically distributed observations that are unlabeled of sample size N. The $\boldsymbol{x}_i$'s originates from a multivariate Gaussian mixture model that consists of K components.

The model's conditional density function is expressed as follows

$$p(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T (\boldsymbol{x} - \boldsymbol{\mu}_k)\right\}$$

Four steps of EM algorithm for Multivariate Gaussian mixture models

1. Initialize values of the unknown parameters at $t = 0$ i.e initialize $\hat{\boldsymbol{\mu}}_k(0)$ and $\hat{\boldsymbol{\sigma}}_k^2(0)$ and $\hat{\pi}_k(0)$, for all k components

2. E-step:

Responsibilities that will be used to estimate the posterior probabilities at time step t with $t \geq 0$ are computed in this step using

$$\hat{Z}_{ik}(t) = \frac{\hat{\pi}_k(t)p\left(x_i, \hat{\boldsymbol{\psi}}(t)\right)}{\sum_{j=1}^{K} \hat{\pi}_j(t)p\left(x_i|\hat{\theta}_j(t)\right)} = \frac{\hat{\pi}_k\left[\frac{1}{\sqrt{2\pi}\hat{\sigma}_k^2} exp\left(-\frac{1}{2\sigma_k^2}(x_i - \hat{\mu}_k)^2\right)\right]}{\sum_{j=1}^{K} \hat{\pi}_j(t)\left[\frac{1}{\sqrt{2\pi}\hat{\sigma}_j^2} exp\left(-\frac{1}{2\sigma_j^2}(x_i - \hat{\mu}_j)^2\right)\right]}$$

for all N observations and k components in the sample.

3. M-step:

Obtain the maximum likelihood estimates by maximizing the objective function at time step $t + 1$ with $t \geq 0$ using

$$\hat{\pi}_k(t+1) = \sum_{i=1}^{N} \frac{\hat{Z}_{ik}}{N}$$

for all k components

$$\hat{\mu}_k(t+1) = \frac{\sum_{i=1}^{N} \hat{Z}_{ik}(t)x_i}{\sum_{i=1}^{N} \hat{Z}_{ik}(t)}$$

for all k components

and

$$\hat{\boldsymbol{\Sigma}}_k(t)(t+1) = \frac{\sum_{i=1}^{N} \hat{Z}_{ik}(t)(\boldsymbol{x}_i - \hat{\mu}_k(t))^2}{\sum_{i=1}^{N} \hat{Z}_{ik}(t)}$$

for all k components

4. Repeat the E-step and M-step until convergence to the global maxima.

4 Application

In this section the parameters of the Gaussian mixture model will be estimated using two different initialization strategies namely: Hastie's method and the alternative method. The two distinct initialization strategies will be performed on a sample randomly generated by the *rannor* function in SAS. The generated samples consist of two clusters and the number of clusters are fixed i.e $k = 2$. The groups $\triangle$ apart, thus $\triangle$ represents the distance between the mean components of the two Gaussian components.

The first Gaussian component has been generated from the Gaussian distribution with parameters $\mu_1 = 50$ with known population variance $\sigma^2 = 16$, the second Gaussian component is $\triangle$ away with parameters $\mu_2 = \mu_1 + \triangle$ and the population variance known i.e $\sigma^2 = 16$. Two initialization strategies will be used to find parameters of a Gaussian mixture model with two components. The two initialization strategies will be employed. An investigation of the techniques will be undertaken. And inference will made based on the absolute bias of means and mixing proportions of the Gaussian mixture model.

4.1 Hastie's method

The number of components of the mixture of Gaussian model will be fixed at two ($k = 2$). Hastie's method chooses an initial starting values $\hat{\mu}_1$ and $\hat{\mu}_2$ at random from the sample. The sample variances σ_1^2 and σ_2^2 are set to the overall sample variance $\sum_{i=1}^{N} \frac{(x_i - \bar{x})^2}{N}$. The Hastie strategy will be evaluated under equal mixing proportions ($\hat{\pi}_1 = \hat{\pi}_2 = 0.5$) and unequal proportions mixing proportions ($\hat{\pi}_1 = 0.3$ and $\hat{\pi}_2 = 0.7$). In both instances where the mixing proportions are equal and unequal. The method by Hastie under different deltas, the deltas indicate how far or close the two components are. Once the EM algorithm is performed under different delta values absolute bias of the means, pies and
number of iterations it takes for the algorithm to reach convergence will be reported on.

4.1.1 Equal Pies ($\hat{\pi}_1 = \hat{\pi}_2 = 0.5$)

1000 simulations are performed under different delta values ($\triangle = 15$, $\triangle = 20$ and $\triangle = 25$), the simulations are performed under seven different sample sizes (n =20, 50, 100, 200, 300, 400, 500). The following results are observed.

Absolute bias of the two mean components for different delta values

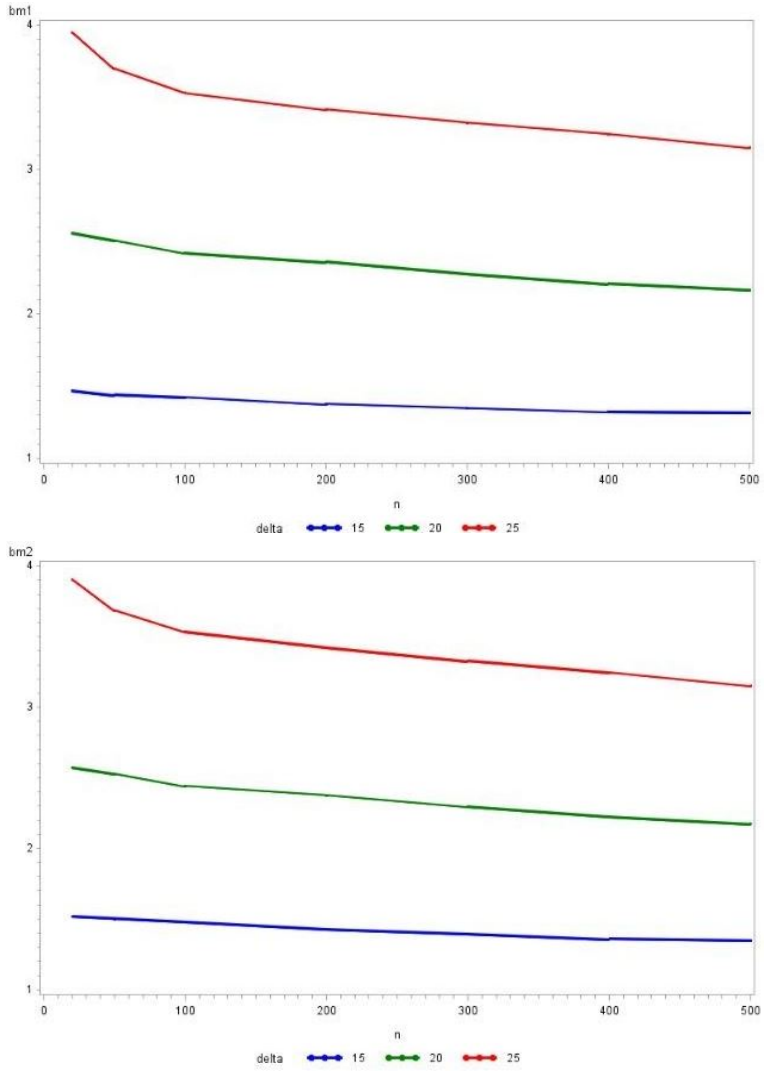


Figure 1: Bias of means for the two components under different means as sample size increases

The absolute bias of the two mean components are observed in Figure 1 for a Gaussian mixture model with two components. The absolute bias for both mean components are relatively the same under different $\triangle$values. Components with small $\triangle$ between them have the least absolute bias of the means while Gaussian components with high $\triangle$ are more bias, which is peculiar, because components closer to each other are expected to give more absolute bias means than objects much further from each other. It is not easy to separate components that are close to each other compared to components that are far from each other. An absolute bias of less about 5.3% is observed for both mean components of the Gaussian mixture model, and this value becomes smaller as the sample size increases.

Absolute bias of the two mixing proportions components for different delta values

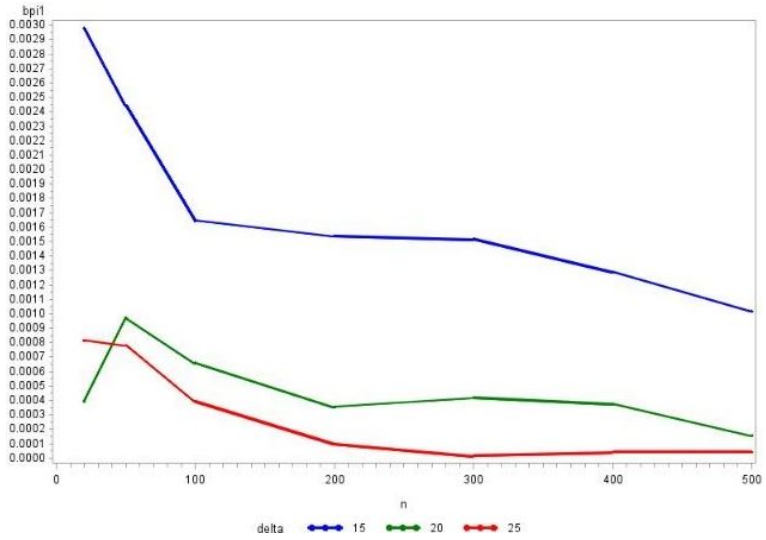


Figure 2: Bias of mixing proportions as sample size increases

Equal mixing proportions imply that the components of the Gaussian mixture model have the same responsibility in explaining observations in the sample. The absolute bias results in Figure 2 shows the observed results for different delta values. Components much further from each other give the least absolute bias estimates of the mixing proportion than components that are close to each other. The initialization strategy provides an absolute bias of the pies value is $\leq 0.6\%$ across all $\triangle$. The absolute bias of the pies gets smaller as the sample size increases.

Number of iterations for different delta values

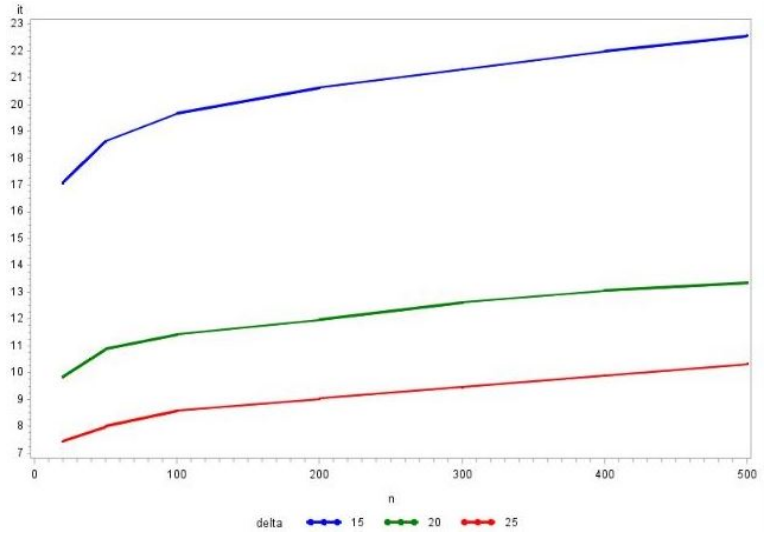


Figure 3: Number of iterations different deltas as sample size increases

The algorithm takes longer to converge to a solution when the two components are close. However the number of iterations before convergence increases as the sample size increase, implying that it the

algorithm takes longer to find estimates of a Gaussian mixture model with a large sample. This makes sense because more observations in a sample means that the EM algorithm has to go through more elements in each iteration.

4.1.2. Unequal Pies ($\hat{\pi}_1 = 0.3$ and $\hat{\pi}_2 = 0.7$)

The mixing probabilities of the Gaussian mixture model are different, meaning that each take different responsibilities in explaining the data set. In this case the second component has a higher responsibility than the first pie. Hastie's initialization technique is employed and the absolute bias of the mean components and mixing proportions is investigated.

Absolute bias of the two mean components for different delta values

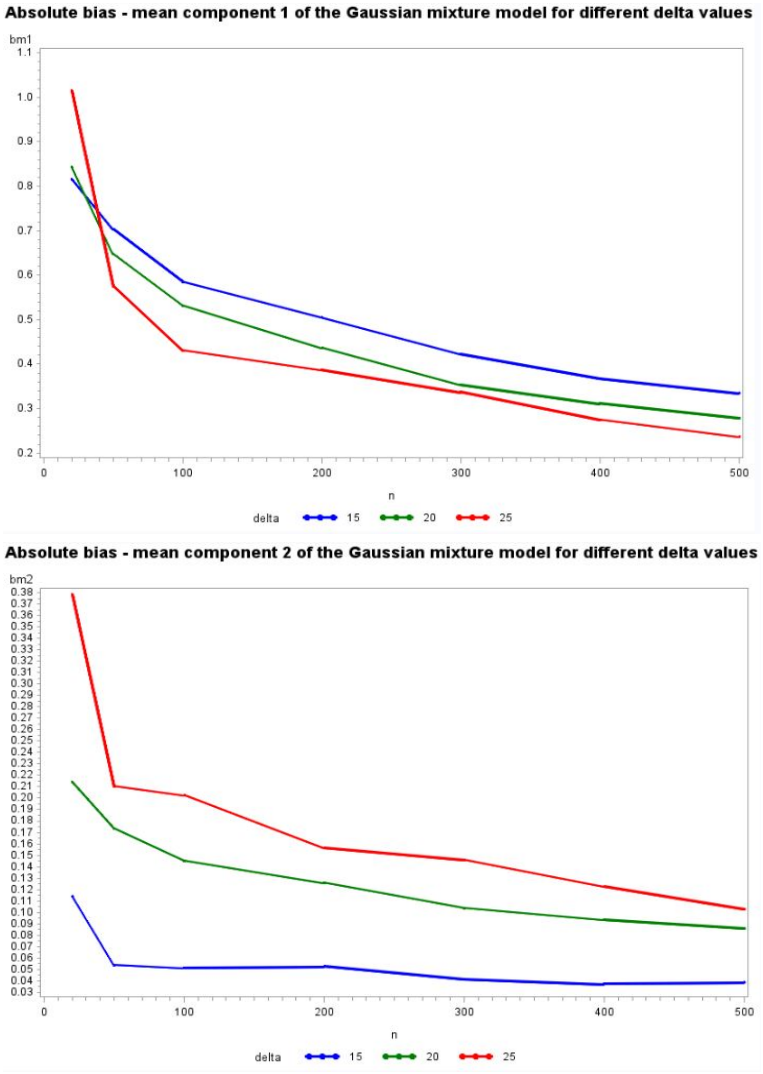


Figure 4: Bias of means for the two components under different means as sample size increases

The responsibilities that make up the Gaussian mixture model represent the proportion of the Gaussian mixture model each component that a Gaussian component explains in the sample. A high responsi-

bility associated with a component suggest that the Gaussian mixture component has a higher weight in the data set. From Figure 4 it is noted that the absolute bias of the mean components for the component with a smaller mixing proportion $\hat{\pi} = 0.3$ is more compared to the absolute bias of mean components with higher mixing proportions $\hat{\pi} = 0.7$. With $\hat{\pi} = 0.3$. The initilization strategy scores an absolute bias of less 2% while $\hat{\pi} = 0.7$ is 0.0051%. The absolute bias of the means decreases as the sample size increase.

Absolute bias of the two mixing proportions components for different delta values

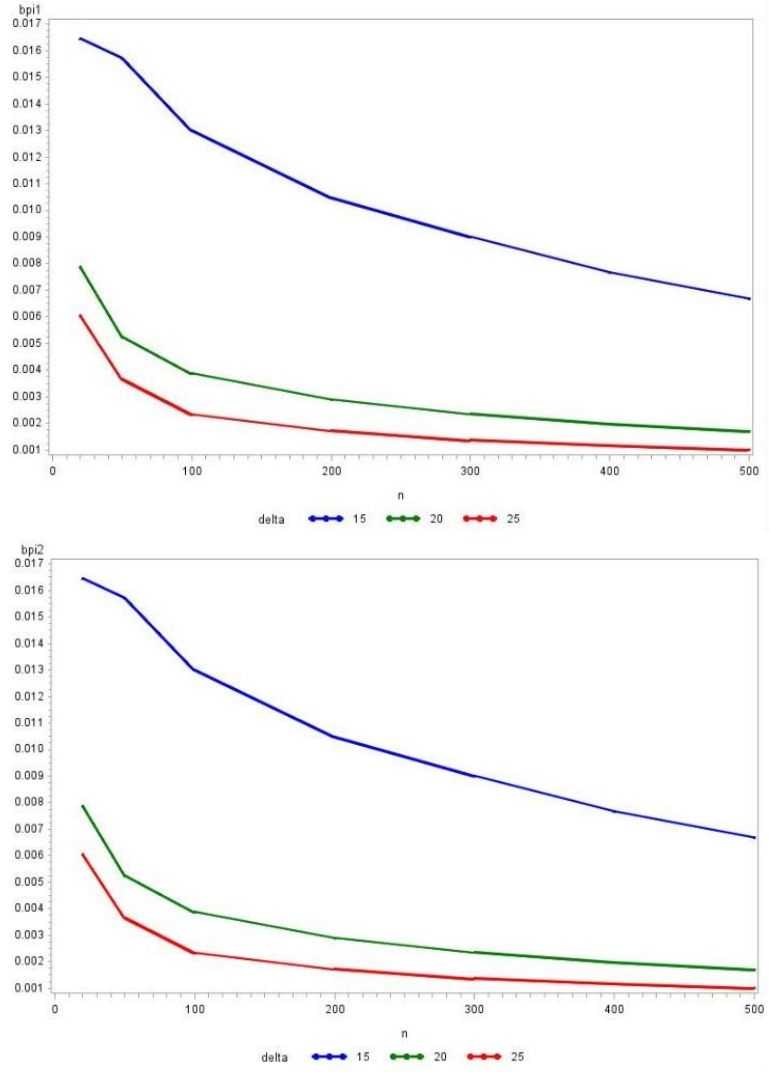


Figure 5: Bias of mixing proportions as sample size increases

The absolute bias of the pies aim to evaluate Hastie's initialization strategy to give correct estimates for mixing proportions of the Gaussian mixture model. Hastie's method give an absolute bias of the component with a smaller mixing proportion is 5.3% while the component with a higher mixing probability is 2.28%. Mixing proportions of components much further from each other give smaller absolute bias implying that the strategy gives bias values when the components are closer to each other. However, the

absolute bias of means decreases as the sample size increases.

Number of iterations for different delta values

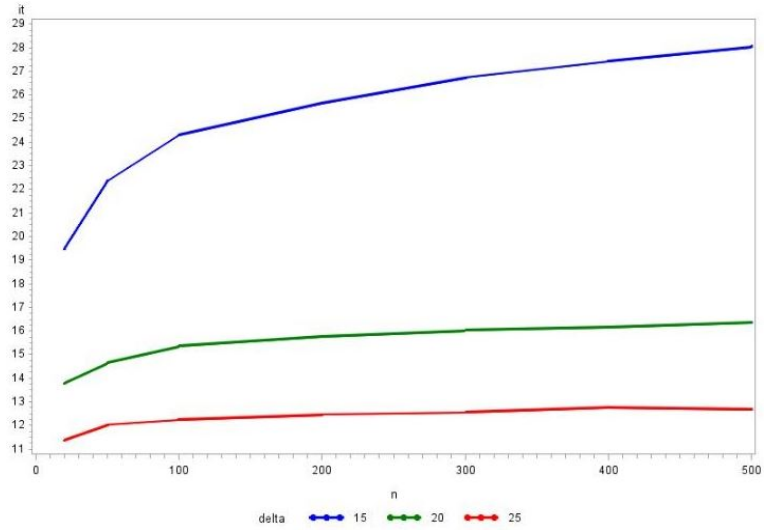


Figure 6: Number of iterations different deltas as sample size increases

Figure 6 shows the number of iterations for the algorithm to reach convergence for the different delta values. Gaussian mixture models with small delta values take longer to reach convergence. In addition, the convergence rate increase as the sample size increases.

4.2 Alternative method

The alternative method takes a randomly generated sample from a normal distribution, sorts the observations in ascending order. Then divides the data set into two halves with the median Q_2 separating the upper quartile Q_3 and the lower quartile Q_1. The initial are calculated in the following manner : $\hat{\mu}_1$ is set to be the average of values found in the first half and $\hat{\mu}_2$ is initialized to be values in the other half. The sample variances s_1^2 and s_2^2 are set to the overall sample variance $\sum_{i=1}^{N} \frac{(x_i - \bar{x})^2}{N}$. The initialization strategies will be evaluated under equal mixing proportions ($\hat{\pi}_1 = \hat{\pi}_2 = 0.5$) and unequal proportions mixing proportions ($\hat{\pi}_1 = 0.3$ and $\hat{\pi}_2 = 0.7$) . In both instances where the mixing proportions are equal and unequal, under different deltas, the deltas indicate how far or close the two components are. Once the EM algorithm is performed, the absolute bias of the means, pies and number of iterations it takes for the algorithm to reach convergence will be reported on.

4.2.1 Equal Pies ($\hat{\pi}_1 = \hat{\pi}_2 = 0.5$)

Bias of the two mean components for different delta values

The absolute bias of the mean components of the Gaussian mixture model for the alternative method is depicted in Figure 7. This strategy gives an absolute bias value of less than 1% for both mean components;

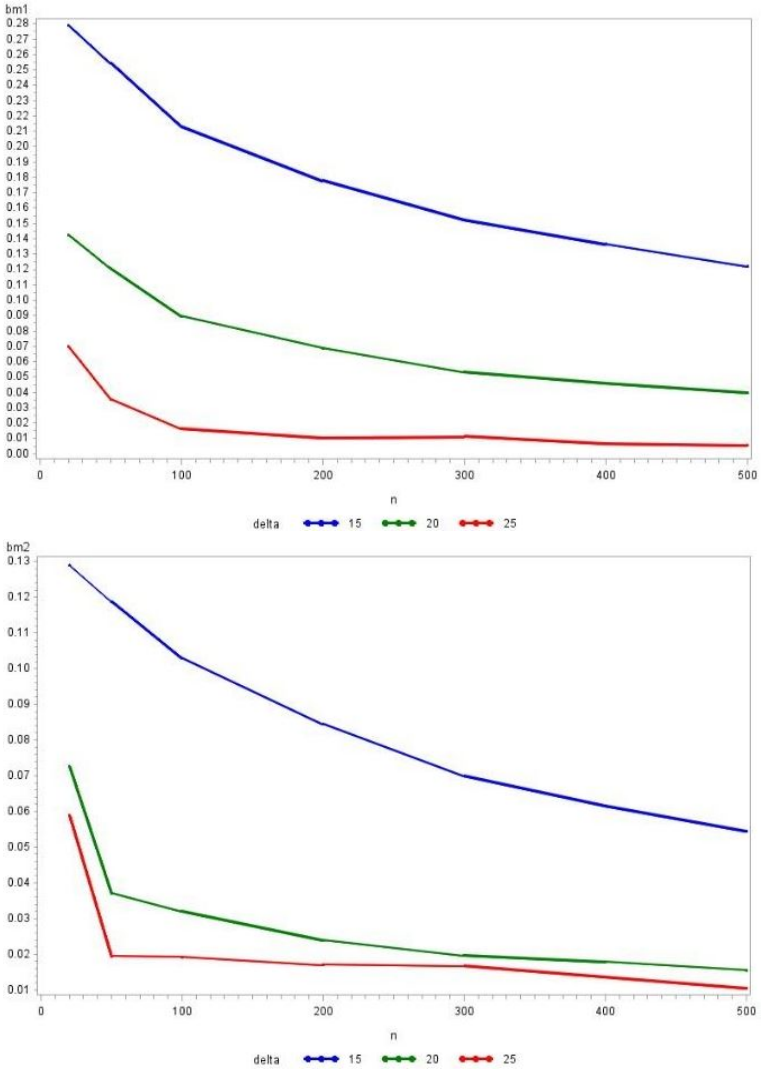


Figure 7: Bias of means for the two components under different means as sample size increases

it provides less absolute bias values for components that are far from each other. The absolute bias of the mean components decreases as the sample size decreases.

Bias of the two mixing proportions components for different delta values

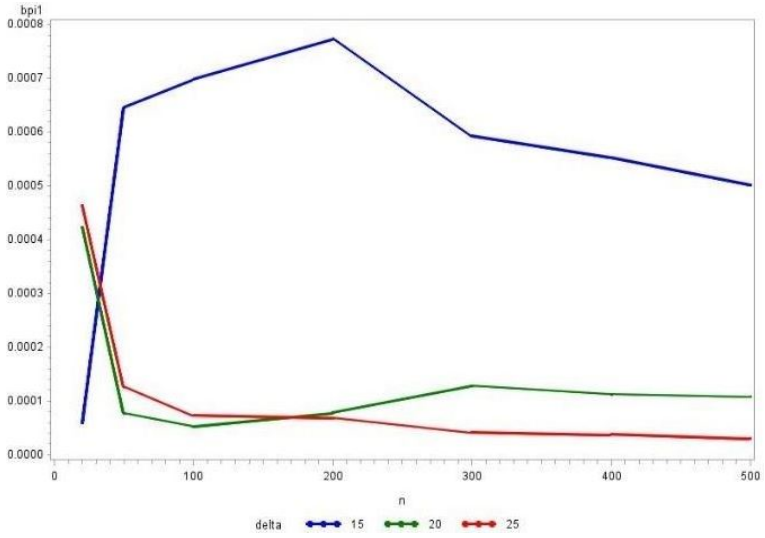


Figure 8: Bias of mixing proportions as sample size increases

The absolute bias of the mixing proportions under equal pies is 0.16%. The alternative method performs better for components of higher delta values. The absolute bias of the mixing proportions decrease as the sample increases.

Number of iterations for different delta values

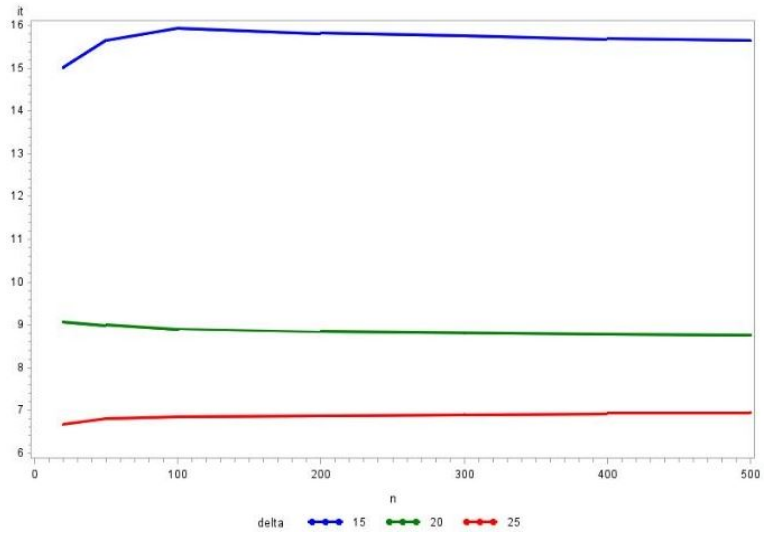


Figure 9: Number of iterations different deltas as sample size increases

Figure 9 shows the number of iterations for the algorithm to reach convergence for the different delta values. Convergence takes longer for components closer to each other and increases with an increase in the sample size.

4.2.2 Unequal Pies ($\hat{\pi}_1 = 0.3$ and $\hat{\pi}_2 = 0.7$)

Absolute bias of the two mean components for different delta values

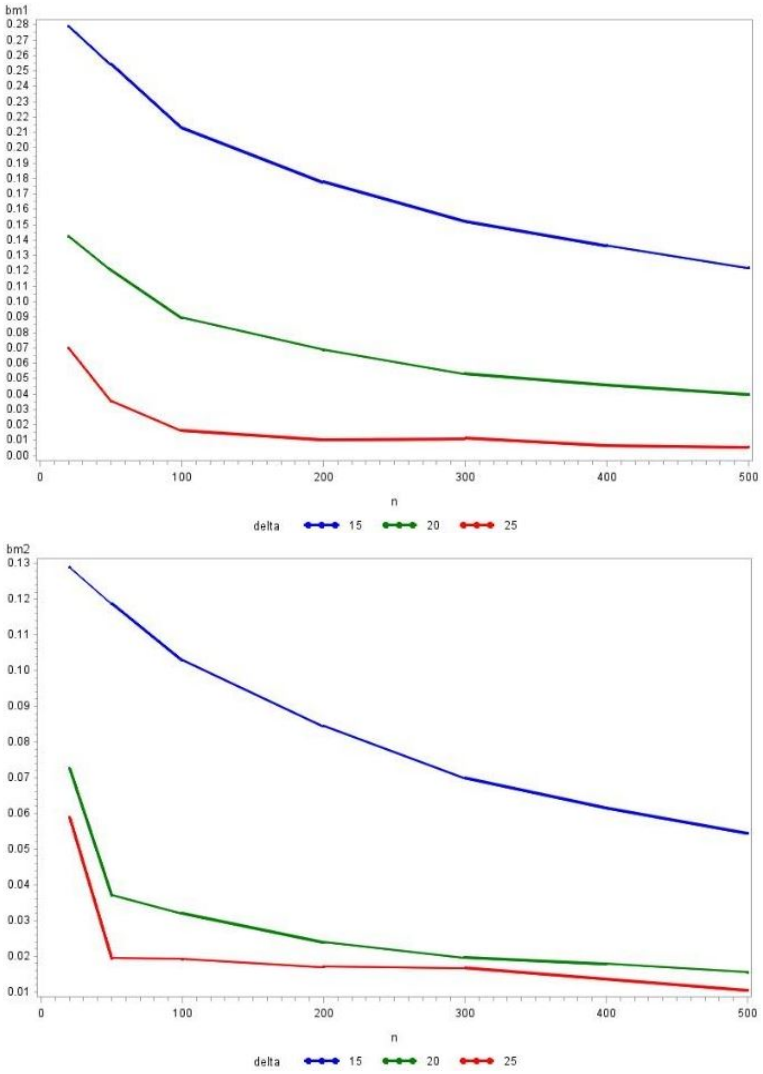


Figure 10: Bias of means for the two components under different means as sample size increases

The absolute bias of the mean of components with smaller mixing proportion is higher (0.43%) than the absolute bias of mean component belonging to larger mixing proportion (0.2%). This strategy performs better when mixture components are much further than each other. However the absolute bias of the mean components gets smaller as the sample size increases.

Absolute bias of the two mixing proportions components for different delta values

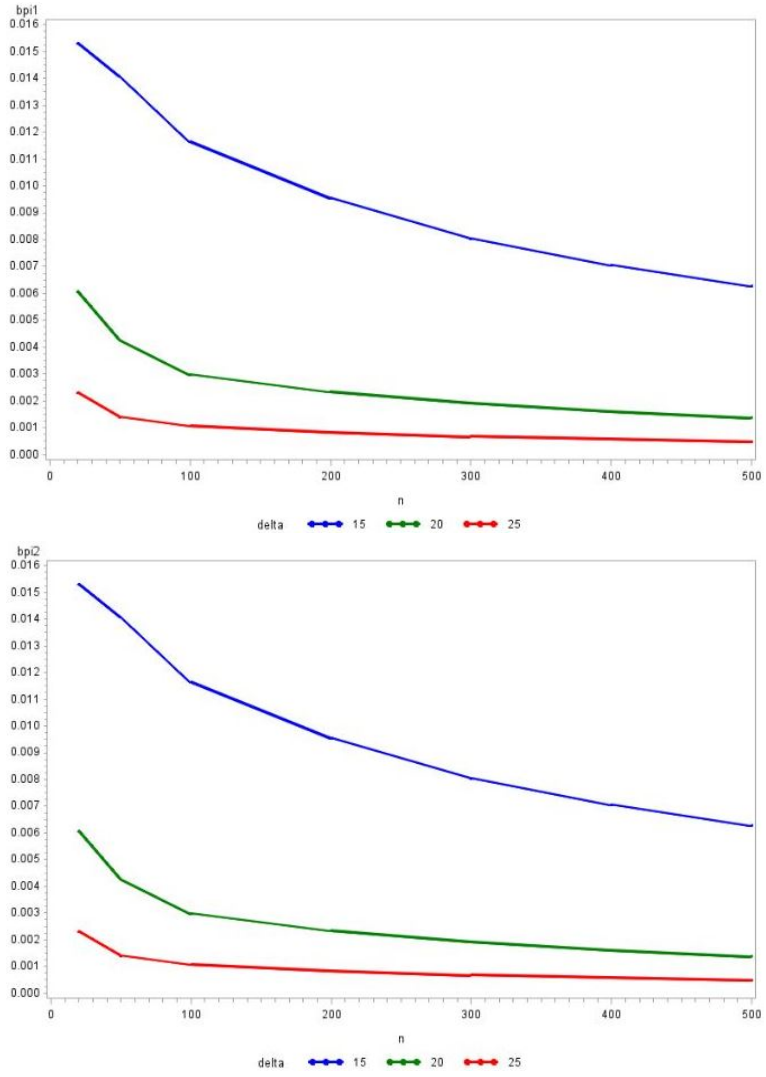


Figure 11: Bias of mixing proportions as sample size increases

The absolute bias of the smaller mixing proportion (5%) is smaller than the absolute bias of than the absolute bias of the bigger mixing proportion (21%). The alternative strategy gives less bias estimates when the two mixture components are much further from each other. However the absolute bias decreases for all values of delta as the sample size increases.

Number of iterations for different delta values

Figure 12 shows the number of iterations for the algorithm to reach convergence for the different delta values. The closer the components are, the more iterations the algorithm takes before reaching convergence, while components further from each other takes less to converge. It is also observed that the bigger the sample size, the longer the iteration will take before it reaches convergence.

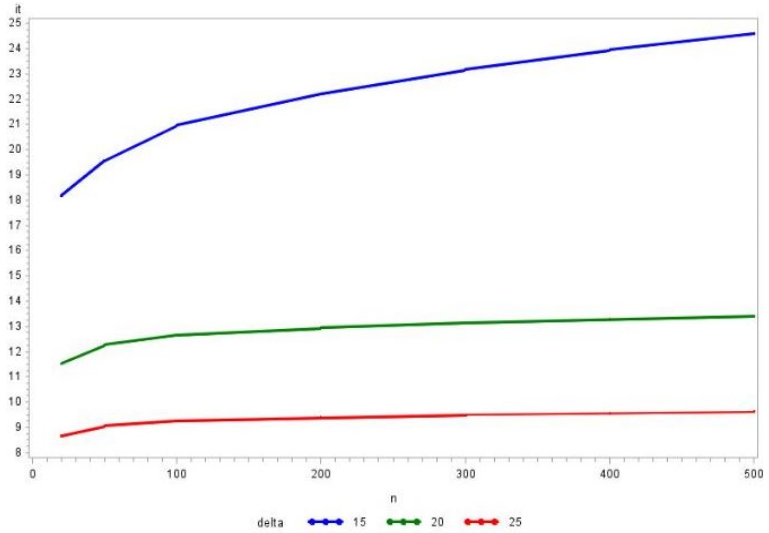


Figure 12: Number of iterations different deltas as sample size increases

4.3 Comparison of two strategies

Case 1: Equal Pies

The alternative strategy gives significantly smaller absolute bias results for both the mean and mixing proportions. Components means that are much further away from each other give smaller absolute bias results under the alternative method. For this reason the alternative initialization strategy is useful in estimating parameters of the Gaussian mixture model for larger delta values. Hastie's strategy might give bias parameter estimates it performs better in finding parameter estimates of mean components that have smaller delta values between them.

The alternative method reaches convergence faster than Hastie's method.

Case 2: Unequal Pies

In the case of unequal pies the alternative method and Hastie's initialization strategy give relatively the same performance for the absolute bias of the mean and mixing proportions. Both strategies perform better with mean components with larger delta values between them.

The alternative method converges at a faster rate than Hastie's method.

5 Conclusion

The alternative initialization strategy produces smaller absolute bias values for both the mean components and mixing proportions when the mixing proportions are equal and converges at a faster rate. However the initialization strategies perform relatively in the same way when the Pies are not equal, even though the alternative method converges at a faster rate than Hastie's method the performance of the two strategies produce satisfactory results.

References

[1] Jeff A Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

[2] Christopher M Bishop. Pattern Recognition. *Machine Learning*, 128:1–58, 2006.

[3] Cuesta-Albertos. Robust estimation in the normal mixture model based on robust clustering. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):779–802, 2008.

[4] Friedman. *The Elements of Statistical Learning*, volume 1. Springer series in statistics New York, 2001.

[5] Sylvia Frühwirth-Schnatter. *Finite Mixture and Markov Switching Models*. Springer Science & Business Media, 2006.

[6] Paolo Giudici and Silvia Figini. Market basket analysis. *Applied Data Mining for Business and Industry, Second Edition*, pages 175–191, 2009.

[7] Jean-Michel Marin. Bayesian modelling and inference on mixtures of distributions. *Handbook of statistics*, 25:459–507, 2005.

[8] Franck Picard. An introduction to mixture models. *Statistics for Systems Biology, Research Report*, (7), 2007.

[9] Richard A Redner and Homer F Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM review*, 26(2):195–239, 1984.

[10] Saldju Tadjudin and David A Landgrebe. Robust parameter estimation for mixture model. *IEEE Transactions on Geoscience and Remote Sensing*, 38(1):439–445, 2000.

[11] Verbeek. Self-organizing mixture models. *Neurocomputing*, 63:99–123, 2005.

[12] Simone Van Wyk. Clustering, self-organizing maps and mixtures of distributions. Master's thesis, University of Pretoria, 2016.

[13] Nanning Zheng and Jianru Xue. *Statistical Learning and Pattern Analysis for Image and Video Processing*. Springer Science & Business Media, 2009.

Appendix

Equal Pies: Hastie's method

SAS Code Used:

```
options ls=132 nodate pageno=1 ;


proc iml ;
pars = {1000 50 15  20 16 16 0.5 0.5 ,
1000 50 15  50 16 16 0.5 0.5 ,
        1000 50 15 100 16 16 0.5 0.5 ,
        1000 50 15 200 16 16 0.5 0.5 ,
        1000 50 15 300 16 16 0.5 0.5 ,
1000 50 15 400 16 16 0.5 0.5 ,
        1000 50 15 500 16 16 0.5 0.5  };


do kk =  1 to nrow(pars) ;
sim_size = pars[kk,1] ;
do jj= 1 to sim_size ;
pi1=pars[kk,7] ;
pi2=pars[kk,8] ;
delta=pars[kk,3] ;
m1=pars[kk,2] ;
m2=m1+delta ;
v1=pars[kk,5] ;
v2=pars[kk,6] ;
n=pars[kk,4] ;
n1=round(n*pi1) ;
n2=n-n1 ;
sd1 = J(n1,1,0) ;
sd2 = J(n2,1,0) ;
x1=rannor(sd1)*sqrt(v1)+m1 ;
x2=rannor(sd2)*sqrt(v2)+m2 ;
x = x1 // x2 ;
```

```
k=2 ;
call sort(x,{1}) ;
mm = x[sample(1:n,2,"NoReplace")] ;
        m1=min(mm) ;
m2=max(mm) ;
if m1=m2 then print "Equal mean values" ;
        v1=var(x) ;
v2=v1 ;
stopc = 0.001 ;
lli=-10000000000000000000 ;
diff=stopc+1 ;
do i = 1 to 50 while (diff>stopc);
s1=sqrt(v1) ;
s2=sqrt(v2) ;
n1 = pdf("Normal",x,m1,s1) ;
n2 = pdf("Normal",x,m2,s2) ;
gm = pi1*n1 || pi2*n2 ;
gm = gm / gm[,+];
m1 = sum(gm[,1]#x) / sum(gm[,1]) ;
m2 = sum(gm[,2]#x) / sum(gm[,2]) ;
v1 = sum(gm[,1] # (x-m1)##2)   / sum(gm[,1]) ;
v2 = sum(gm[,2] # (x-m2)##2)   / sum(gm[,2]) ;
pi1 = (gm[,1])[:];
pi2 = (gm[,2])[:];
ll = sum(log(pi1*n1+pi2*n2)) ;
results = results // (i || m1 || m2 || v1 || v2 || pi1 || pi2 || ll) ;
diff = abs(lli-ll) ;
lli=ll ;
end ;

varest = var(results[,2]) || var(results[,3]) ||
var(results[,4]) || var(results[,5]) ||
var(results[,6]) || var(results[,7])  ;
```

```
res_it = res_it // (jj || i-1 || m1 || m2 || v1 || v2 || pi1 || pi2 || ll || varest) ;
free results  ;

end ;

nm={"ss" "nr" "it" "m1" "m2" "v1" "v2" "pi1" "pi2" "ll"
    "varm1" "varm2" "varv1" "varv2" "varpi1" "varpi2"
    "sim_size" "tm1" "delta" "n" "tv1" "tv2"
    "tpi1" "tpi2" } ;
ares_it = ares_it // (kk || res_it[:,] || pars[kk,] );
end ;

print  ares_it[colname=nm] ;

create simResDelta15 from ares_it[colname=nm] ;
append from ares_it ;
close simResDelta15 ;

quit ;




data simRes15Delta ;
 set simResDelta15 ;
 bm1 = abs(m1-tm1) ;
 bm2 = abs(m2-(tm1+delta)) ;
 *bv1 = abs(v1-tv1);
 *bv2 = abs(v2-tv2) ;
 bpi1= abs(pi1-tpi1) ;
 bpi2= abs(pi2-tpi2) ;
run ;

symbol1 interpol=join width=2
```

```
  color=blue
        value=dot
        height=.5;
symbol2 interpol=join width=2
  color=green
        value=dot
        height=.5;
symbol3 interpol=join width=2
  color=red
        value=dot
        height=.5;



%macro plt(v1,v2,v3) ;
proc gplot data=simRes15Delta ;
plot &v1*n=delta ;
title " Absolute bias - &v3 component &v2" ;
run ;
%mend ;

*%plt(bm1,1,mean) ;
*%plt(bm2,2,mean) ;
*%plt(bv1,1,variance) ;
*%plt(bv2,2,variance) ;
*%plt(bpi1,1,pi) ;
*%plt(bpi2,2,pi) ;


*%macro plt1(v1,v2,v3) ;
*proc gplot data=simRes15Delta;
*plot &v1*n=delta ;
*title" Variance - &v3 component &v2" ;
*run ;
```

```
*%mend ;
*%plt1(varm1,1,mean) ;
*%plt1(varm2,2,mean) ;
*%plt1(varpi1,1,pi) ;

*proc gplot data=simRes15Delta ;
*plot it*n=delta ;
*title" Average number of iterations to solution" ;
*run;
*quit ;


/*Code to plot the graphs for different delta values*/
data Diffdeltas ;
set Simres15delta Simres20delta Simres25delta;
run;

%macro plt5(v1,v2,v3);
proc gplot data = Diffdeltas;
plot &v1*n=delta;
title " Absolute bias of &v3 component &v2 of the Gaussian mixture model for different delta values '
run;
%mend ;



%plt5(bm1,1,mean) ;
%plt5(bm2,2,mean) ;
%plt5(bpi1,1,pi) ;
%plt5(bpi2,2,pi) ;

%macro plt6(v1) ;
proc gplot data = Diffdeltas ;
plot &v1*n=delta ;
```

```
title "Number of iterations for the different Deltas" ;
run ;
%mend ;
%plt6(it) ;
```

```
/*data ab;
set Simres10delta Simres15delta;
run;

%macro plt5(v1,v2,v3) ;
proc gplot data=ab ;
plot &v1*n=delta ;
title" Absolute bias - &v3 component &v2" ;
run ;
%mend ;

%plt5(bm1,1,mean) ;
*/
```

Equal Pies: Alternative method

```
options ls=132 nodate pageno=1 ;


proc iml ;
pars = {1000 50 15  20 16 16 0.5 0.5 ,
1000 50 15  50 16 16 0.5 0.5 ,
        1000 50 15 100 16 16 0.5 0.5 ,
        1000 50 15 200 16 16 0.5 0.5 ,
        1000 50 15 300 16 16 0.5 0.5 ,
1000 50 15 400 16 16 0.5 0.5 ,
        1000 50 15 500 16 16 0.5 0.5 } ;


do kk =  1 to nrow(pars) ;
sim_size = pars[kk,1] ;
do jj= 1 to sim_size ;
pi1=pars[kk,7] ;
pi2=pars[kk,8] ;
delta=pars[kk,3] ;
m1=pars[kk,2] ;
m2=m1+delta ;
v1=pars[kk,5] ;
v2=pars[kk,6] ;
n=pars[kk,4] ;
n1=round(n*pi1) ;
n2=n-n1 ;
sd1 = J(n1,1,0) ;
sd2 = J(n2,1,0) ;
x1=rannor(sd1)*sqrt(v1)+m1 ;
x2=rannor(sd2)*sqrt(v2)+m2 ;
x = x1 // x2 ;
k=2 ;
pi1=0.5 ;
pi2=1-pi1 ;
call sort(x,{1}) ;
```

```
call qntl(q,x) ;
qpi = q[2] ;
mx = x#(x<=qpi) || x#(x>qpi) ;
m1 = loc(mx[,1]^=0) ;
m1 = (mx[,1])[m1] ;
m1 = m1[:] ;
m2 = loc(mx[,2]^=0) ;
m2 = (mx[,2])[m2] ;
m2 = m2[:] ;
v1 = var(x) ;
v2=v1 ;
stopc = 0.001 ;
lli=-10000000000000000000 ;
diff=stopc+1 ;
do i = 1 to 50 while (diff>stopc);
s1=sqrt(v1) ;
s2=sqrt(v2) ;
n1 = pdf("Normal",x,m1,s1) ;
n2 = pdf("Normal",x,m2,s2) ;
gm = pi1*n1 || pi2*n2 ;
gm = gm / gm[,+];
m1 = sum(gm[,1]#x) / sum(gm[,1]) ;
m2 = sum(gm[,2]#x) / sum(gm[,2]) ;
v1 = sum(gm[,1] # (x-m1)##2)   / sum(gm[,1]) ;
v2 = sum(gm[,2] # (x-m2)##2)   / sum(gm[,2]) ;
pi1 = (gm[,1])[:];
pi2 = (gm[,2])[:];
ll = sum(log(pi1*n1+pi2*n2)) ;
results = results // (i || m1 || m2 || v1 || v2 || pi1 || pi2 || ll) ;
diff = abs(lli-ll) ;
lli=ll ;
end ;

varest = var(results[,2]) || var(results[,3]) ||
```

```
var(results[,4]) || var(results[,5]) ||
var(results[,6]) || var(results[,7])  ;

res_it = res_it // (jj || i-1 || m1 || m2 || v1 || v2 || pi1 || pi2 || ll || varest) ;
free results  ;

end ;

nm={"ss" "nr" "it" "m1" "m2" "v1" "v2" "pi1" "pi2" "ll"
    "varm1" "varm2" "varv1" "varv2" "varpi1" "varpi2"
    "sim_size" "tm1" "delta" "n" "tv1" "tv2"
    "tpi1" "tpi2" } ;
ares_it = ares_it // (kk || res_it[:,] || pars[kk,] );
end ;

print  ares_it[colname=nm] ;

create simResDelta15 from ares_it[colname=nm] ;
append from ares_it ;
close simResDelta15 ;

quit ;




data simRes15Delta ;
 set simResDelta15 ;
 bm1 = abs(m1-tm1) ;
 bm2 = abs(m2-(tm1+delta)) ;
 bv1 = abs(v1-tv1);
 bv2 = abs(v2-tv2) ;
 bpi1= abs(pi1-tpi1) ;
 bpi2= abs(pi2-tpi2) ;
```

```
run ;

symbol1 interpol=join width=2
  color=blue
        value=dot
        height=.5;
symbol2 interpol=join width=2
  color=green
        value=dot
        height=.5;
symbol3 interpol=join width=2
  color=red
        value=dot
        height=.5;



%macro plt(v1,v2,v3) ;
proc gplot data=simRes15Delta ;
plot &v1*n=delta ;
title" Absolute bias - &v3 component &v2" ;
run ;
%mend ;

*%plt(bm1,1,mean) ;
*%plt(bm2,2,mean) ;
*%plt(bv1,1,variance) ;
*%plt(bv2,2,variance) ;
*%plt(bpi1,1,pi) ;
*%plt(bpi2,2,pi) ;


/*Code to plot the graphs for different delta values*/
data Diffdeltas ;
```

```
set Simres25delta Simres15delta Simres20delta;
run;

%macro plt5(v1,v2,v3);
proc gplot data = Diffdeltas;
plot &v1*n=delta;
title "Absolute bias - &v3 component &v2 of the Gaussian mixture model for different delta values";
run;
%mend ;

%plt5(bm1,1,mean);
%plt5(bm2,2,mean) ;
%plt5(bpi1,1,pi) ;
%plt5(bpi2,2,pi) ;

%macro plt6(v1);
proc gplot data = Diffdeltas;
plot &v1*n=delta;
title "Number of iterations for the different Deltas";
run;
%mend ;
%plt6(it) ;
```

Unequal Pies: Hastie's method

```
options ls=132 nodate pageno=1 ;
```

```
proc iml ;
pars = {1000 50 15  20 16 16 0.3 0.7 ,
1000 50 15  50 16 16 0.3 0.7 ,
        1000 50 15 100 16 16 0.3 0.7 ,
        1000 50 15 200 16 16 0.3 0.7 ,
        1000 50 15 300 16 16 0.3 0.7 ,
1000 50 15 400 16 16 0.3 0.7 ,
        1000 50 15 500 16 16 0.3 0.7 } ;

do kk =  1 to nrow(pars) ;
sim_size = pars[kk,1] ;
do jj= 1 to sim_size ;
pi1=pars[kk,7] ;
pi2=pars[kk,8] ;
delta=pars[kk,3] ;
m1=pars[kk,2] ;
m2=m1+delta ;
v1=pars[kk,5] ;
v2=pars[kk,6] ;
n=pars[kk,4] ;
n1=round(n*pi1) ;
n2=n-n1 ;
sd1 = J(n1,1,0) ;
sd2 = J(n2,1,0) ;
x1=rannor(sd1)*sqrt(v1)+m1 ;
x2=rannor(sd2)*sqrt(v2)+m2 ;
x = x1 // x2 ;
k=2 ;
pi1=0.3 ;
pi2=1-pi1 ;
call sort(x,{1}) ;
mm = x[sample(1:n,2,"NoReplace")] ;
        m1=min(mm) ;
m2=max(mm) ;
```

```
if m1=m2 then print "Equal mean values" ;
        v1=var(x) ;
v2=v1 ;
stopc = 0.001 ;
lli=-10000000000000000000 ;
diff=stopc+1 ;
do i = 1 to 50 while (diff>stopc);
s1=sqrt(v1) ;
s2=sqrt(v2) ;
n1 = pdf("Normal",x,m1,s1) ;
n2 = pdf("Normal",x,m2,s2) ;
gm = pi1*n1 || pi2*n2 ;
gm = gm / gm[,+];
m1 = sum(gm[,1]#x) / sum(gm[,1]) ;
m2 = sum(gm[,2]#x) / sum(gm[,2]) ;
v1 = sum(gm[,1] # (x-m1)##2)   / sum(gm[,1]) ;
v2 = sum(gm[,2] # (x-m2)##2)   / sum(gm[,2]) ;
pi1 = (gm[,1])[:];
pi2 = (gm[,2])[:];
ll = sum(log(pi1*n1+pi2*n2)) ;
results = results // (i || m1 || m2 || v1 || v2 || pi1 || pi2 || ll) ;
diff = abs(lli-ll) ;
lli=ll ;
end ;

varest = var(results[,2]) || var(results[,3]) ||
var(results[,4]) || var(results[,5]) ||
var(results[,6]) || var(results[,7])  ;

res_it = res_it // (jj || i-1 || m1 || m2 || v1 || v2 || pi1 || pi2 || ll || varest) ;
free results  ;

end ;
```

```
nm={"ss" "nr" "it" "m1" "m2" "v1" "v2" "pi1" "pi2" "ll"
    "varm1" "varm2" "varv1" "varv2" "varpi1" "varpi2"
    "sim_size" "tm1" "delta" "n" "tv1" "tv2"
    "tpi1" "tpi2" } ;
ares_it = ares_it // (kk || res_it[:,] || pars[kk,] );
end ;

print  ares_it[colname=nm] ;

create simResDelta15 from ares_it[colname=nm] ;
append from ares_it ;
close simResDelta15 ;

quit ;




data simRes15Delta ;
 set simResDelta15 ;
 bm1 = abs(m1-tm1) ;
 bm2 = abs(m2-(tm1+delta)) ;
 bv1 = abs(v1-tv1);
 bv2 = abs(v2-tv2) ;
 bpi1= abs(pi1-tpi1) ;
 bpi2= abs(pi2-tpi2) ;
run ;

symbol1 interpol=join width=2
  color=blue
        value=dot
        height=.5;
symbol2 interpol=join width=2
  color=green
```

```
        value=dot
        height=.5;
symbol3 interpol=join width=2
  color=red
        value=dot
        height=.5;



%macro plt(v1,v2,v3) ;
proc gplot data=simRes15Delta ;
plot &v1*n=delta ;
title" Absolute bias - &v3 component &v2 of the Gaussian mixture model for different delta values" ;
run ;
%mend ;

*%plt(bm1,1,mean) ;
*%plt(bm2,2,mean) ;
*%plt(bv1,1,variance) ;
*%plt(bv2,2,variance) ;
*%plt(bpi1,1,pi) ;
*%plt(bpi2,2,pi) ;


%macro plt1(v1,v2,v3) ;
proc gplot data=simRes15Delta;
plot &v1*n=delta ;
title" Variance - &v3 component &v2" ;
run ;
%mend ;
*%plt1(varm1,1,mean) ;
*%plt1(varm2,2,mean) ;
*%plt1(varpi1,1,pi) ;
```

```
*proc gplot data=simRes15Delta ;
*plot it*n=delta ;
*title" Average number of iterations to solution" ;
*run;
*quit ;

/*data ab;
set Simres10delta Simres15delta;
run;

%macro plt5(v1,v2,v3) ;
proc gplot data=ab ;
plot &v1*n=delta ;
title" Absolute bias - &v3 component &v2" ;
run ;
%mend ;

%plt5(bm1,1,mean) ;
*/

/*Code to plot the graphs for different delta values*/
data Diffdeltas ;
set Simres25delta Simres15delta Simres20delta;
run;

%macro plt5(v1,v2,v3);
proc gplot data = Diffdeltas;
plot &v1*n=delta;
title "Absolute bias - &v3 component &v2 of the Gaussian mixture model for different delta values";
run;
%mend ;

%plt5(bm1,1,mean);
%plt5(bm2,2,mean) ;
```

```
%plt5(bpi1,1,pi) ;
%plt5(bpi2,2,pi) ;

%macro plt6(v1);
proc gplot data = Diffdeltas;
plot &v1*n=delta;
title "Number of iterations for the different Deltas";
run;
%mend ;
%plt6(it) ;
```

Unequal Pies: Alternative method

```
options ls=132 nodate pageno=1 ;

proc iml ;
pars = {1000 50 15  20 16 16 0.3 0.7 ,
1000 50 15  50 16 16 0.3 0.7 ,
        1000 50 15 100 16 16 0.3 0.7 ,
        1000 50 15 200 16 16 0.3 0.7 ,
        1000 50 15 300 16 16 0.3 0.7 ,
1000 50 15 400 16 16 0.3 0.7 ,
        1000 50 15 500 16 16 0.3 0.7 } ;
do kk =  1 to nrow(pars) ;
sim_size = pars[kk,1] ;
do jj= 1 to sim_size ;
pi1=pars[kk,7] ;
pi2=pars[kk,8] ;
delta=pars[kk,3] ;
m1=pars[kk,2] ;
```

```
m2=m1+delta ;
v1=pars[kk,5] ;
v2=pars[kk,6] ;
n=pars[kk,4] ;
n1=round(n*pi1) ;
n2=n-n1 ;
sd1 = J(n1,1,0) ;
sd2 = J(n2,1,0) ;
x1=rannor(sd1)*sqrt(v1)+m1 ;
x2=rannor(sd2)*sqrt(v2)+m2 ;
x = x1 // x2 ;
k=2 ;
pi1=0.3 ;
pi2=1-pi1 ;
call sort(x,{1}) ;
call qntl(q,x) ;
qpi = q[2] ;
mx = x#(x<=qpi) || x#(x>qpi) ;
m1 = loc(mx[,1]^=0) ;
m1 = (mx[,1])[m1] ;
m1 = m1[:] ;
m2 = loc(mx[,2]^=0) ;
m2 = (mx[,2])[m2] ;
m2 = m2[:] ;
v1 = var(x) ;
v2=v1 ;
stopc = 0.001 ;
lli=-10000000000000000000 ;
diff=stopc+1 ;
do i = 1 to 50 while (diff>stopc);
s1=sqrt(v1) ;
s2=sqrt(v2) ;
n1 = pdf("Normal",x,m1,s1) ;
n2 = pdf("Normal",x,m2,s2) ;
```

```
gm = pi1*n1 || pi2*n2 ;
gm = gm / gm[,+];
m1 = sum(gm[,1]#x) / sum(gm[,1]) ;
m2 = sum(gm[,2]#x) / sum(gm[,2]) ;
v1 = sum(gm[,1] # (x-m1)##2)   / sum(gm[,1]) ;
v2 = sum(gm[,2] # (x-m2)##2)   / sum(gm[,2]) ;
pi1 = (gm[,1])[:];
pi2 = (gm[,2])[:];
ll = sum(log(pi1*n1+pi2*n2)) ;
results = results // (i || m1 || m2 || v1 || v2 || pi1 || pi2 || ll) ;
diff = abs(lli-ll) ;
lli=ll ;
end ;

varest = var(results[,2]) || var(results[,3]) ||
var(results[,4]) || var(results[,5]) ||
var(results[,6]) || var(results[,7])  ;

res_it = res_it // (jj || i-1 || m1 || m2 || v1 || v2 || pi1 || pi2 || ll || varest) ;
free results  ;

end ;

nm={"ss" "nr" "it" "m1" "m2" "v1" "v2" "pi1" "pi2" "ll"
    "varm1" "varm2" "varv1" "varv2" "varpi1" "varpi2"
    "sim_size" "tm1" "delta" "n" "tv1" "tv2"
    "tpi1" "tpi2" } ;
ares_it = ares_it // (kk || res_it[:,] || pars[kk,] );
end ;

print  ares_it[colname=nm] ;

create simResDelta15 from ares_it[colname=nm] ;
append from ares_it ;
```

```
close simResDelta15 ;

quit ;




data simRes15Delta ;
 set simResDelta15 ;
 bm1 = abs(m1-tm1) ;
 bm2 = abs(m2-(tm1+delta)) ;
 bv1 = abs(v1-tv1);
 bv2 = abs(v2-tv2) ;
 bpi1= abs(pi1-tpi1) ;
 bpi2= abs(pi2-tpi2) ;
run ;

symbol1 interpol=join width=2
  color=blue
        value=dot
        height=.5;
symbol2 interpol=join width=2
  color=green
        value=dot
        height=.5;
symbol3 interpol=join width=2
  color=red
        value=dot
        height=.5;



%macro plt(v1,v2,v3) ;
proc gplot data=simRes15Delta ;
```

```
plot &v1*n=delta ;
title" Absolute bias - &v3 component &v2" ;
run ;
%mend ;

*%plt(bm1,1,mean) ;
*%plt(bm2,2,mean) ;
*%plt(bv1,1,variance) ;
*%plt(bv2,2,variance) ;
*%plt(bpi1,1,pi) ;
*%plt(bpi2,2,pi) ;



%macro plt1(v1,v2,v3) ;
proc gplot data=simRes15Delta;
plot &v1*n=delta ;
title" Variance - &v3 component &v2" ;
run ;
%mend ;
*%plt1(varm1,1,mean) ;
*%plt1(varm2,2,mean) ;
*%plt1(varpi1,1,pi) ;

*proc gplot data=simRes15Delta ;
*plot it*n=delta ;
*title" Average number of iterations to solution" ;
*run;
*quit ;
/*Code to plot the graphs for different delta values*/
data Diffdeltas ;
set Simres25delta Simres15delta Simres20delta;
run;

%macro plt5(v1,v2,v3);
```

```
proc gplot data = Diffdeltas;
plot &v1*n=delta;
title "Absolute bias - &v3 component &v2 of the Gaussian mixture model for different delta values";
run;
%mend ;

%plt5(bm1,1,mean);
%plt5(bm2,2,mean) ;
%plt5(bpi1,1,pi) ;
%plt5(bpi2,2,pi) ;

%macro plt6(v1);
proc gplot data = Diffdeltas;
plot &v1*n=delta;
title "Number of iterations for the different Deltas";
run;
%mend ;
%plt6(it) ;
```

Comparing algorithms for the inverse gamma distribution

Tshiamo Motswiri 13099419

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor(s): Mr. J. Ferreira, Co-supervisor(s): Prof. A. Bekker

Department of Statistics, University of Pretoria

30 October 2017 (final)

Abstract

This study investigates different algorithms to estimate parameters of the inverse gamma distribution. Computational challenges of the estimation will be briefly addressed. The parameters will be tested by comparing the cumulative distribution function of each of the estimated parameters from the algorithms with the empirical distribution function of real data as well as the empirical distribution function of simulated data, using goodness-of-fit measures.

Declaration

I, *Tshiamo Brian Motswiri,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

Tshiamo Brian Motswiri

_____

Mr. Johan Ferreira

_____

Prof. Andriette Bekker

_____

30/10/2017

Date

Acknowledgements

Special thanks to Mr J. Ferreira and Prof. A. Bekker for the supervision and support they gave in the construction of this research report.

A special thanks also goes to the National Research Foundation (Reference: CPRR160403161466, Grant No: 105840) for the funding they provided.

Contents

List of Figures

List of Tables

1 Introduction

The gamma distribution is a skewed distribution (non-symmetrical) and has only positive parameters. It has two parameters, i.e. α and β where one is known as the scale parameter and the other known as the shape parameter, respectively. The probability density function (pdf) of a gamma distribution is as given by:

$$f(y) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} y^{\alpha-1} e^{-\beta y} \tag{1}$$

where $\alpha, \beta > 0$; $y > 0$, where the distribution is denoted by $Y \sim$Gamma(α, β), $\Gamma(.)$.

In this study, the results given and the methodology included in the estimation of parameters will be for the inverse gamma distribution. Thus if a random variable Y has pdf in equation (1), the random variable $X = \frac{1}{Y}$ has the inverse gamma distribution. Since this is a transformation of random variables, the first step to take is to determine the Jacobian of the transformation from Y to X before obtaining the pdf of the random variable X. The Jacobian can be determined as

$$J(Y \to X) = \mid \frac{\partial}{\partial x}(x^{-1}) \mid = x^{-2}.$$

The pdf of X, it is given by:

$$f(x) = f(y = x^{-1}) \cdot J(Y \to X). \tag{2}$$

Thus

$$\begin{aligned} f(x) &= \frac{\beta^{\alpha}}{\Gamma(\alpha)} (x^{-1})^{\alpha-1} e^{-\beta x^{-1}} \cdot x^{-2} \\ &= \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{-\alpha+1-2} e^{-\frac{\beta}{x}} \\ &= \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{-\alpha-1} e^{-\frac{\beta}{x}} \end{aligned} \tag{3}$$

for $x > 0$. The cumulative distribution function (cdf) of X can be obtained using (3) as:

$$\begin{aligned} F(x) &= \int_0^x f(t)dt \\ &= \int_0^x \frac{\beta^\alpha}{\Gamma(\alpha)} t^{-(\alpha+1)} e^{-\frac{\beta}{t}} dt \\ &= \frac{\beta^\alpha}{\Gamma(\alpha)} \int_0^x t^{-(\alpha+1)} e^{-\frac{\beta}{t}} dt \\ &= \frac{1}{\Gamma(\alpha)} \int_0^x \beta^\alpha t^{-(\alpha+1)} e^{-\frac{\beta}{t}} dt \\ &= \frac{\Gamma\left(\alpha, \frac{\beta}{x}\right)}{\Gamma(\alpha)} \end{aligned}$$

where $\Gamma(\alpha, \frac{\beta}{x}) = \int_0^x \beta^\alpha t^{-(\alpha+1)} e^{-\frac{\beta}{t}} dt$, and where the numerator in the last expression is the incomplete gamma function, and the denominator is a gamma function [3]. Note that the gamma and inverse gamma distributions have identical shape parameters and the scale parameter of the inverse gamma distribution is just the reciprocal of the gamma distribution's scale parameter. Thus the inverse gamma distribution is also a skewed distribution.

In the case of the random variable Y mentioned above, α is the scale parameter and β is the shape parameter; so the skewness of the gamma distribution is determined by the shape parameter and the scale parameter determines the statistical dispersion of the distribution. The larger the value of the scale parameter is, the more spread out the distribution will be, and the smaller the scale parameter is, the more concentrated the distribution will be. In this report, the focus is on estimation procedures used to estimate the parameters of the gamma distribution as well as the parameters of the inverse gamma distribution. This will be done using the algorithms proposed by [11] for the different estimation methods. Subsequently, the next step is to then move onto fitting these estimated inverse gamma distribution parameters to a given data set.

The following set of graphs indicate how the pdf (3) changes as the shape parameter increases (with a constant scale parameter $\alpha = 1$):

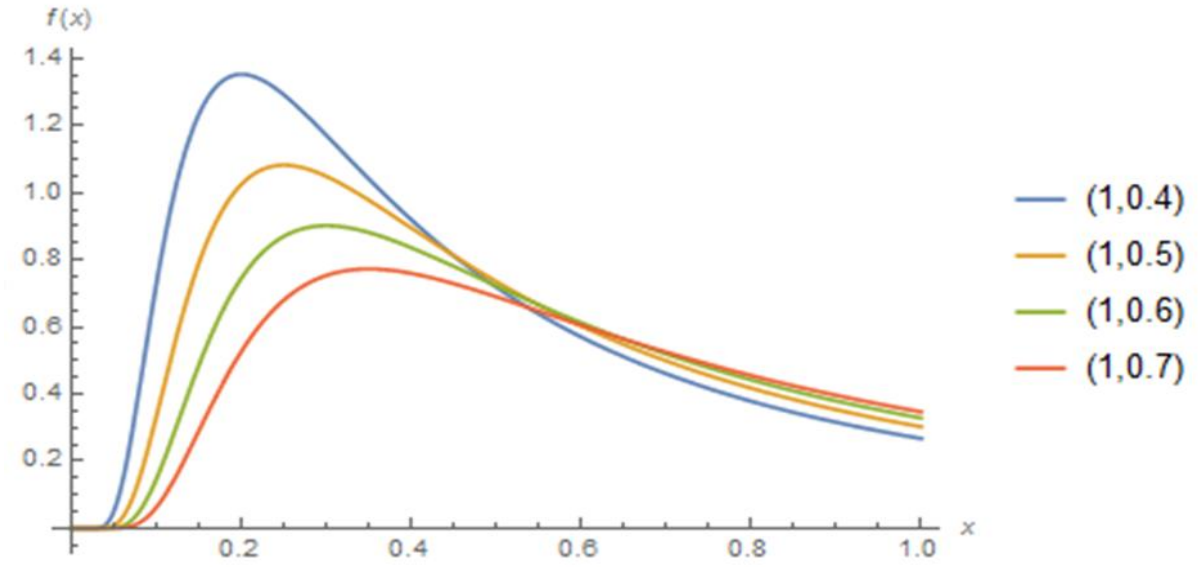


Figure 1: Increase in shape parameter PDF

To determine the effect of the scale parameter, the next set of graphs is used, with a constant shape parameter ($\beta = 1.5$) :

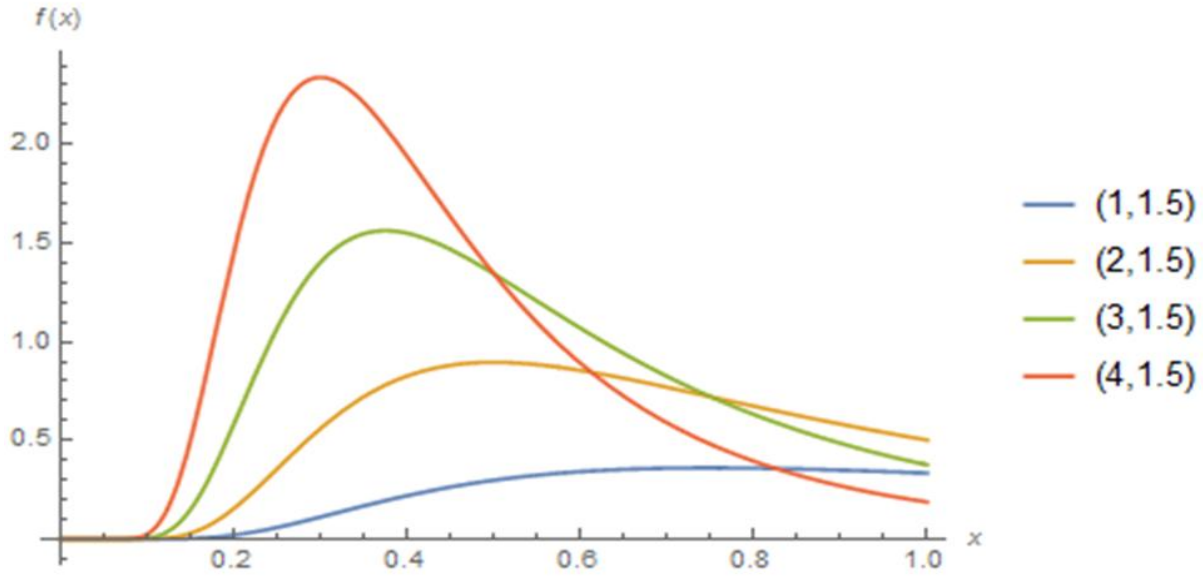


Figure 2: Increase in scale parameter PDF

Another important part of this theory is to realise what type of effect that an increase in either of the parameters has on the cdf of the inverse gamma distribution.

The following set of graphs indicate how the cdf curve changes as the scale parameter increases (for a

constant value $\beta = 0.1$):

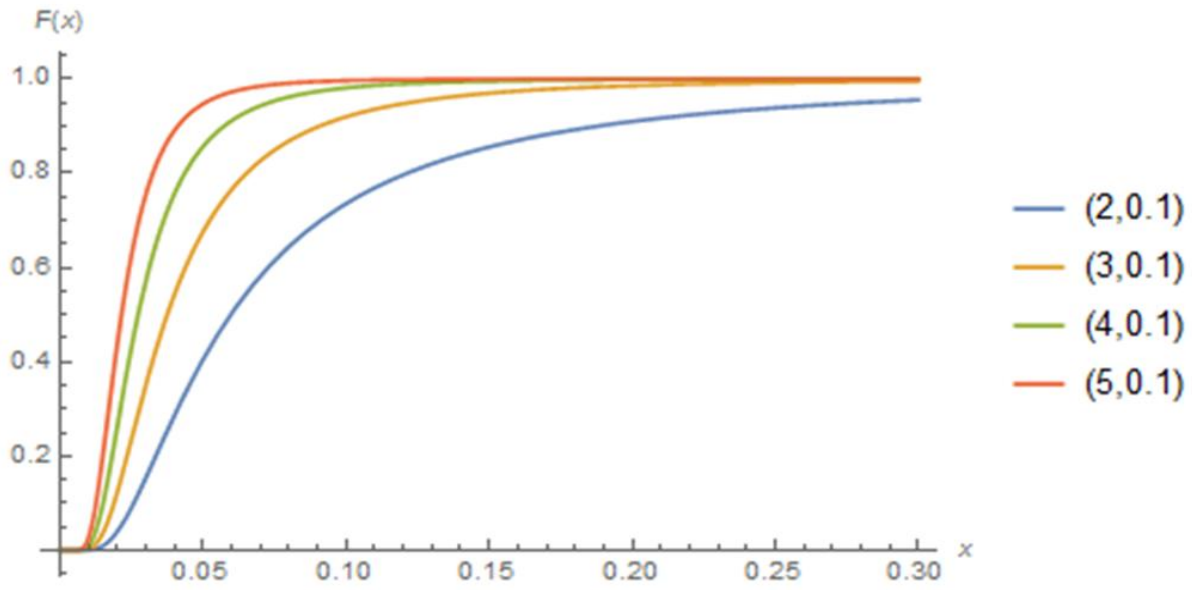


Figure 3: Increase in scale parameter for CDF

The next thing to be aware of is the effect of an increase in the shape parameter, which is what the following set of graphs indicate (for a constant value $\alpha = 1.4$):

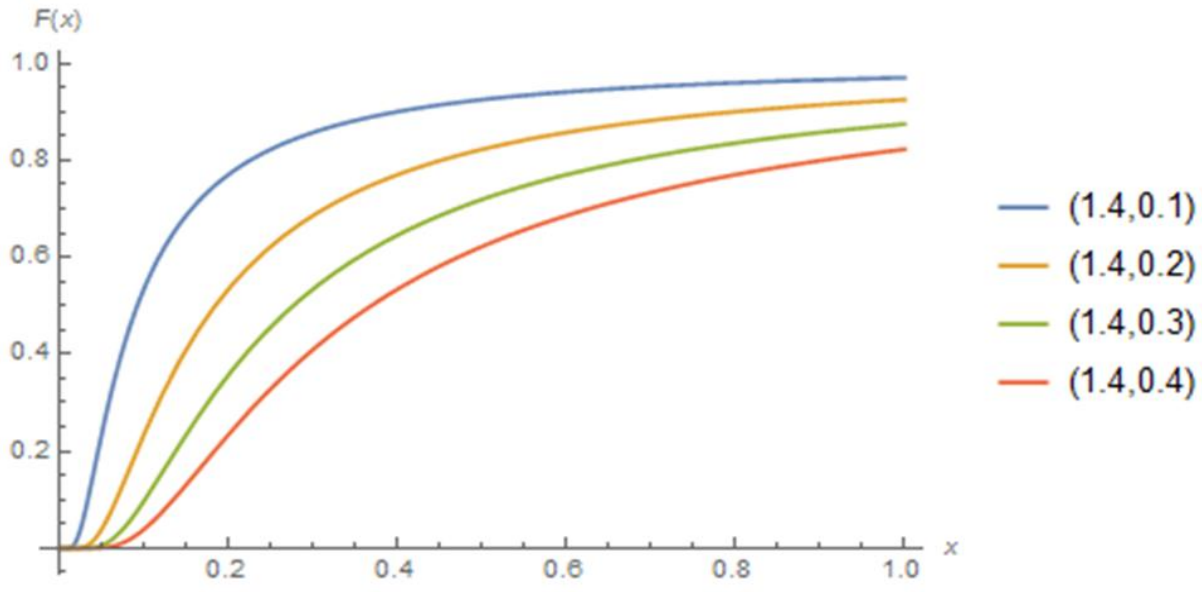


Figure 4: Increase in shape parameter for CDF

The first algorithm is derived using method of moments estimation, and the derivation of the two param-

eters under this estimation procedure are determined directly without the necessity of convergence. However, the second algorithm is derived using maximum likelihood estimation, and in this case the derivation of the rst parameter requires convergence and a special inequality. Once that process is completed, it is then used to obtain the other parameters for this distribution.

- **Method of Moments Estimation(MME)** : Derivations of equations that relate the population moments to the parameters that are being estimated
- **Maximum Likelihood Estimation(MLE)** : Stemming from the analytical difficulties in obtaining estimators under maximum likelihood estimation for the inverse gamma distribution.

In particular, [11] proposed algorithms for fitting the inverse gamma distribution to a data set. This study investigates two algorithms and considers the sensitivity of the estimated parameters between the above-mentioned estimation methods (MME and MLE).

2 Background Theory

2.1 Literature review

In 1969,[2] aimed at estimating the parameters of the gamma distribution, and in order to solve this problem,[2] used Maximum Likelihood Estimation to estimate both the parameters of the gamma distribution and their bias. [2] also investigated the bias of the parameters numerically, and to improve accuracy, they obtained a convenient table for the purpose of assessing the Maximum Likelihood Estimation method of the gamma distribution parameters. Thus the conclusion drawn from this task was only based on the MLE method results, and they concluded that the bias of both the parameter estimates of the gamma distribution using Maximum Likelihood are positive. In 2009, [6] focused on the quality of the MLEs for both the parameters of the gamma distribution; and the aim was to obtain analytical approximations for the bias of the maximum likelihood estimates for the parameters of the gamma distribution. But in doing this, [6] only used small samples. They introduced two methods that can be used in order to bias-adjust the estimators obtained from MLE, these two methods being the bootstrap and another method suggested by Cox and Snell in 1968. [6] needed to find out which of the two methods is more effective, and thus a better method to use. In discovering this, a simulation study was conducted, and it revealed that the methodology presented by [5] in 1968 was more effective than the bootstrap method; this simulation experiment was done through Monte Carlo. According to [6], the chosen method was then used to derive simpler expressions for bias of the MLEs for the gamma distribution parameters. It was therefore concluded that the methodology suggested

by [5] is the better method to use when we use maximum likelihood to estimate the parameters of the gamma distribution. Now the article written by [4] on the inverse gamma distribution in 2008 also confirms this, in this paper [4] derives the distribution of the inverse gamma distribution, calculates the moments of the distribution and proved that the inverse gamma distribution is a conjugate prior for an exponential likelihood function.

In 2002, [13] had the task of deriving an efficient algorithm that can be used to estimate both parameters of the gamma distribution for maximum likelihood estimation. In doing so, a log-likelihood function was used for the purpose of working with a simplified expression of maximum likelihood estimation. Also, [13] only sampled 100 points for this experiment; and only two algorithms are illustrated for maximising the log-likelihood function. The first one described iteratively maximizes the lower bound of the log-likelihood function, and the second one, which is considered to be faster, is obtained through generalized Newton.

In accordance to the above-mentioned lower bound, [13] then concludes that this approximation is very close to the true log-likelihood, verifying it's good performance. The gamma distribution was used by [7] to represent Africa's monthly rainfall in order to monitor drought within that region, [7] also developed models that they used as a tool to manage necessary resources such as water and food, and these models were generated by the gamma distribution. These models were useful in the sense that they helped with the evaluation of the likelihood of rainfalL occurences. In this study conducted by [7] ,the Kolmogorov-Smirnof (KS) test was used to compare the results obtained from using the gamma distribution with another distribution that is regularly used in rainfal occurences, and in this case, it was the Weibull distribution. However, in 1994 [9] discussed studies that utilized more statistical distributions with the goal of accurately fitting the precipitation data. But [9] specifically chose to use the gamma distribution to represent precipitation repeatedly because it equivalently produces a representation of different distributions all-together but only using two parameters, that being the shape and the scale parameters. Another interesting discovery that [8] brought to light is that the parameters of the gamma distribution could describe rainfall occurences for different timescales, regardless of the date. Now in order to fully utilize the gamma distribution for rainfal events, it is required to estimate it's parameters, and [14] used both the MLE method as well as the MME method to estimate these parameters. However the MME proved to be an inefficient method since it's estimates were inaccurate for small shape values, and it was thus concluded that it is a poor estimation procedure. Hence as a result of this, the MLE was used instead, in order to accommodate regions with small-scale parameters. Once the parameter estimation procedure is completed, these parameters have to be evaluated to ensure that they are accurate in accordance to the historical data before they are used to represent the modeled probability distribution of rainfall for a specified location.

From here, the next step taken by [7] was to use the KS test in order to carefully assess the relationship between the empirical distribution and the theoretical (gamma) distribution. To establish this, a hypothesis test took place, where the null hypothesis was the theoretical distribution giving satisfactory performance when modeling historical data, in other words, whether the gamma distribution is suitable or not and the level of rejection that was used is 0.1. This now implies that the null hypothesis will be rejected for locations with p-values that are less than 0.1. As was expected at the time, [7] found that 98.5% of the data had a p-value that is higher than 0.1, which implies that the null hypothesis is not rejected, leading to the conclusion that the gamma distribution is very suitable with regards to the approximation of the historical rainfall distribtuions.

In 2016 [10] introduced two Bayesian estimators for the purpose of having more information on the parameters of the gamma distribution. One of the algorithms uses an unnormalized conjugate prior for the shape of the gamma, and the second one uses a non-linear approximation to the likelihood and a prior on the shape that is conjugate to the approximated likelihood. In order to approximate the required expectations, Laplace is used for both algorithms. In concluding the experiment, [10] evaluated that the Bayesian algorithms have the same bias properties as the Maximum Likelihood bias properties.

Another good application of the inverse gamma distribution was performed by [1], and in this scenario it is used to model sea reflections. The very first procedure required in order to perform experiments such as this one in statistics is to estimate the parameters of the appropriate distribution that is to be used for the application, and the distribution used is the inverse gamma distribution. The reason behind the choice of this distribution is that modelling sea fluctuations is an example of of modelling Compound-Gaussian Clutter, which are much often used for heavy-tailed clutter distributions.

Therefore [1] used the Maximum Likelihood estimation procedure as well as the Method of Fractional Moments to estimate the parameters of the inverse gamma distributions.

The Kolmogorov-Smirnov test is a popular goodness-of-fit test that was thoroughly explained by [12] by means of tabled percentage points, the power function, the cumulative distribution's confidence intervals and examples thereof. It is essentially a test based on the maximum difference between the hypothesized cumulative distribution and the empirical distribution. This test is referred to as a goodness-of-fit test if the focus of the test is whether there is an agreement between the theoretical distribution and the distribution that follows from a set of values. If the sampling distribution is independent of the distribution of the population and/or independent of certain parameters, it is called a distribution-free (non-parametric) test. So [12] then discusses an alternative non-parametric goodness-of-fit test.

2.2 Methodology

2.2.1 Method of Moments

Suppose X follows the inverse gamma distribution with the pdf derived in equation (3), with $x > 0$; $\alpha > 0$, $\beta > 0$ where α is the shape parameter, and β the scale parameter, and where the function $\Gamma(.)$ is the gamma function. The expected value and variance of this distribution is given by,

$$E(X) = \frac{\beta}{\alpha - 1} \equiv \mu \tag{4}$$

and

$$VAR(X) = \frac{\beta^2}{(\alpha - 1)^2 (\alpha - 2)} \equiv \nu \tag{5}$$

(see [11]) respectively. In order to estimate the parameters α and β using the method of moments estimation method, explicit expressions of α and β in terms of the expected value and the variance are required.

From equation (4), it follows that α can be expressed as

$$\alpha = \frac{\beta + \mu}{\mu} \tag{6}$$

where μ denotes the expected value of X. Substituting equation (6) into equation (5) gives the result

$$\begin{aligned}
\nu &= \frac{\beta^2}{\left(\frac{\beta+\mu}{\mu} - 1\right)^2 \left(\frac{\beta+\mu}{\mu} - 2\right)} \\
&= \frac{\beta^2}{\left(\frac{\beta}{\mu}\right)^2 \left(\frac{\beta - \mu}{\mu}\right)} \\
&= \frac{\beta^2}{\frac{\beta^2(\beta-\mu)}{\mu^3}} \\
&= \frac{\mu^3}{\beta - \mu}
\end{aligned} \tag{7}$$

where ν denotes the variance of X. Now the expression obtained is in terms of only one parameter, β. The

next step now is to manipulate equation (7) to explicitly express β in terms of $\mu$and ν. So,

$$\begin{aligned}\nu\left(\hat{\beta}-\mu\right) &= \mu^3 \\ \hat{\beta} &= \frac{\mu^3+2\nu}{\nu} \\ &= \mu\left(\frac{\mu^2}{\nu}+2\right)\end{aligned} \tag{8}$$

Substituting equation (8) in (6) gives

$$\begin{aligned}\hat{\alpha} &= \frac{\mu^2+2\nu}{\nu} \\ &= \frac{\mu^2}{\nu}+2\end{aligned} \tag{9}$$

for $\nu > 0$. Therefore equation (8) and (9) represent the estimators of α and β under method of moments estimation.

2.2.2 Maximum Likelihood

Similarly, expressions for the parameters α and β under maximum likelihood estimation must be obtained. The first step is to write the pdf of the inverse gamma distribution given in equation (4) as a log-likelihood function of the vector x with observations $x = \{x_1, x_2, ..., x_n\}$, i.e.

$$\begin{aligned}f(x) &= \frac{\beta^\alpha}{\Gamma(\alpha)}x^{-(\alpha+1)}e^{-(\frac{\beta}{x})} \\ \prod_{i=1}^{n} f(x_i) &= \frac{\beta^{n\alpha}}{\Gamma(\alpha)^n}\prod_{i=1}^{n}(\mathbf{x}_i)e^{-\frac{\beta}{\sum_{i=1}^{n} x_i}} \\ log\prod_{i=1}^{n} f(x_i) &= log(\frac{\beta^{n\alpha}}{\Gamma(\alpha)^n}) + log(\prod_{i=1}^{n}\mathbf{x_i}^{-(\alpha+1)}) - \beta\sum_{i=1}^{n} x_i^{-1} \\ log\prod_{i=1}^{n} f(x_i) &= n\alpha log(\beta) - nlog(\Gamma(\alpha)) - n(\alpha+1)\overline{log(\mathbf{x})} - \beta\sum_{i=1}^{n} x_i^{-1}\end{aligned} \tag{10}$$

What is now required is to maximize equation (10) by differentiation with respect to β, and equate that to zero in order to find the maximum likelihood estimator, i.e.

$$\frac{\partial log\left(\prod_{i=1}^{n} f(x)\right)}{\partial\beta} = 0,$$

thus

$$\frac{n\alpha}{\beta} - \sum_{i=1}^{n} x_i^{-1} = 0$$

$$\hat{\beta} = \frac{n\alpha}{\sum_{i=1}^{n} x_i^{-1}} \tag{11}$$

To get the ML estimator of α, a different approach must be taken because the maximization of equation (10) directly with respect to α is not possible mathematically. Thus the first step taken is by substituting equation (11) into equation (10), which yields:

$$\begin{aligned} log\prod_{i=1}^{n} f(x) &= n\alpha log\left(\tfrac{n\alpha}{\sum_{i=1}^{n} x_i^{-1}}\right) - nlog\left(\Gamma(\alpha)\right) - n(\alpha+1)\overline{log(\mathbf{x})} - \tfrac{n\alpha}{\sum_{i=1}^{n} x_i^{-1}}\left(\sum_{i=1}^{n} x_i^{-1}\right) \\ &= n\alpha\left[log(n\alpha) - log\left(\sum_{i=1}^{n} x_i^{-1}\right)\right] - nlog\left(\Gamma(\alpha)\right) - n\alpha\overline{log(\mathbf{x})} - n\overline{log(\mathbf{x})} - n\alpha \\ &= n\alpha log(n\alpha) - n\alpha log\left(\sum_{i=1}^{n} x_i^{-1}\right) - nlog\left(\Gamma(\alpha)\right) - n\alpha\overline{log(\mathbf{x})} - n\overline{log(\mathbf{x})} - n\alpha \end{aligned} \tag{12}$$

Even after the previous substitution of equations it is not possible to directly maximize equation (12) with respect to α. Therefore, the alternative approach in maximizing equation (12) is by using the linear constrain

$$\alpha log(\alpha) \geq (1 + log(\alpha_0))(\alpha - \alpha_0) + \alpha_0 log(\alpha_0) \tag{13}$$

[11] and substituting equation (13) into (12), which yields

$$\begin{aligned} log\left(log\prod_{i=1}^{n} f(x)\right) &\geq -n\alpha\overline{log(\mathbf{x})} - n\overline{log(\mathbf{x})} - nlog\left(\Gamma(\alpha)\right) + n\alpha log(n) - n\alpha_0 + n\alpha log(\alpha_0) - n\alpha log\left(\sum_{i=1}^{n} x_i^{-1}\right) \\ &= \alpha\left[-n\overline{log(\mathbf{x})} + nlog(n) + nlog(\alpha_0) - nlog\left(\sum_{i=1}^{n} x_i^{-1}\right)\right] - n\overline{log(\mathbf{x})} - nlog\left(\Gamma(\alpha)\right) - n\alpha_0 \end{aligned} \tag{14}$$

Maximizing equation (14) (differentiate with respect to α), results in $\frac{\partial log\left(log\prod_{i=1}^{n} f(x)\right)}{\partial\alpha} = 0$:

$$0 = -n\overline{log(\mathbf{x})} + nlog(n) + nlog(\alpha_0) - nlog\left(\sum_{i=1}^{n} x_i^{-1}\right) - n\left(\frac{1}{\Psi(\alpha)}\right)$$

$$\frac{\partial log\left(log\prod_{i=1}^{n} f(x)\right)}{\partial\alpha} = 0 : -n\overline{log(\mathbf{x})} + nlog(n) + nlog(\alpha_0) - nlog\left(\sum_{i=1}^{n} x_i^{-1}\right) - n\left(\frac{1}{\Psi(\alpha)}\right) = 0$$

$$\Longleftrightarrow$$

$$\Psi^{-1}(\alpha) = log(n) + log(\alpha_0) - \overline{log(\mathbf{x})} - log\left(\sum_{i=1}^{n} x_i^{-1}\right)$$
$$\alpha = \Psi\left(log(n) + log\left(\alpha_0\right) - \overline{log(\mathbf{x})} - log\left(\sum_{i=1}^{n} x_i^{-1}\right)\right)$$

and therefore

$$\alpha = \Psi\left(log\left(n\alpha_0\right) - \overline{log(\mathbf{x})} - log\left(\sum_{i=1}^{n} x_i^{-1}\right)\right) \tag{15}$$

where $\Psi(.)$ denotes the digamma function. So no explicit expressions can be obtained for α and β using both MM estimation and ML estimation.

The next step from here is to find the best estimates for the parameters $\{\alpha, \beta\}$ using both estimation procedures (MM and ML). To find the estimators of the parameters using MM estimation is straight forward because there is no necessity to iterate any value. Hence equation (8) and (9) respresent the estimators for β and α respectively.

However, to find the estimators for the parameters under ML estimation is a bit more complicated. In this case, order is important since it is required to estimate α first in order to directly substitute that value into equation (11) to obtain the respective ML estimator for β. First α_0 is taken as the initial value, and then start an iterative process by continuously updating α_0 with α until it converges to the α in equation (9).

Once α has converged, that value can then be substituted into equation (11) to find the corresponding estimator for β.

2.3 Kolmogorov-Smirnov Goodness-of-fit test

The goodness-of-fit for any statistical model is simply a representation of how well the model itself fits a certain dataset. It can also be seen as the measured distance between the observed values and the values obtained theoretically from the particular model that is used. Measures of this nature are typically used for

testing statistical hypotheses.

The different hypothesis tests include testing for normailty, that is, testing if a set of observations are normally distributed, or testing whether a given number of samples follow a similar distribution.

For each type of hypothesis that is tested, there are specific statistical goodness-of-fit tests that are eligible to be used. For example, if an investigator decides to investigate whether a dataset follows a certain distribution then the following tests can be used:

- Kolmogorov-Smirnov test
- Cramer-von Mises criterion
- Anderson-Darling test
- Shapiro-Wilk test
- Chi-Squared test

In this investigation however, the Kolmogorov-Smirnov goodness-of-fit test is used, because it is most commonly used and relatively much more intuitive as compared to the other goodness-of-fit tests. The basis of this test is essentially upon the vertical maximum difference between the hypothetical distribution and the empirical distribution. The distribution of the Kolmogorov-Smirnov test statistic is independent of the underlying cumulative distribution function that is being tested.

The limitations of the Kolmogorov-Smirnov test include:

- Only applicable to continuous distributions
- Tends to be more sensitive closer to the center of the distribution than at the tails
- The distribution must be fully specified

Once the parameter pairs for each of the two algorithms are obtained, these parameter pairs will then be used to plot the CDF curves that follow from the inverse gamma distribution, and thus compare each of them to the empirical curve.

The reason for this is to determine which algorithm best fits the empirical distribution. But since it is not enough to conclude by a mere graphical observation, the Kolmogorov-Smirnov goodness-of-fit test is introduced as a measure to carefully assess which algorithm best fits the dataset.

3 Application

3.1 Description of Data sets

3.1.1 Real Data

This data set comprises of one variable and 1100 observations. Below is a table that shows the descriptive statistics for this data set:

N	Mean	Standard Deviation	Minumum	Maximum	Sum	Median
1100	0.3222379	0.2565929	0.0432783	2.2008115	354.4616449	0.2439673

Table 1: Descriptive Statistics-Real Data

3.1.2 Simulated Data

This data set also comprises of one variable, but has 1000 observations. The following displays the descriptive statsistics for the data set:

N	Mean	Standard Deviation	Minumum	Maximum	Sum	Median
1000	0.5310748	0.2668844	0.3297800	5.2945300	531.0748400	0.4665200

Table 2: Descriptive Statistics-Simulated Data

3.2 Algorithms

In this section, all the above-mentioned theory together with the relevant methodology will be represented in a more practical format. The estimation procedures (MME and MLE) that were discussed earlier will be interpreted as computational algorithms in order to obtain the estimators corresponding to each of the algorithms. Now these two algorithms will be expressed explicitly in order to fully explain how the estimators are obtained and to adequately put emphasis on the methodology behind every calculation.

3.2.1 Algorithm 1 - Method of Moments

This first algorithm demonstrates the computational procedure taken to obtain the estimators that were derived in equations (8) and (9). Suppose that the observations are given by:

$x = \{x_1, x_2, ..., x_n\}$, $x_i > 0$, then

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$
$$\nu = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$$
$$\hat{\alpha} = \frac{\mu^2}{\nu} + 2$$
$$\hat{\beta} = \mu\left(\left(\frac{\mu^2}{\nu}\right) + 1\right)$$

return $\hat{\alpha}$, $\hat{\beta}$, where μ represents the mean of the observed values and ν represents the variance of the observations. The MM estimators are given by $\hat{\alpha}$ and $\hat{\beta}$.

Algorithm 1 provides us with the estimators for both parameters α and β using Method of Moments Estimation and is the easiest algorithm compared to algorithm 2, because it requires no extra calculations when determining the estimates for α and β, nor does it require any iterations for any of the parameters. Thus by performing a basic direct substitution of the mean and the variance into the expressions for the estimators, we can obtain the values of $\hat{\alpha}$ and $\hat{\beta}$ explicitly.

3.2.2 Algorithm 2 - Maximum Likelihood

This algorithm demonstrates the computational derivation of the ML estimators. Also, if the observations are:

$x = \{x_1, x_2, ...x_n\}$,$x_i > 0$, then

$$\begin{aligned}
\mu &= \frac{1}{n}\sum_{i=1}^{n} x_i \\
\nu &= \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2 \\
\alpha &= \frac{\mu^2}{\nu} + 2 \\
C &= -log\left(\sum_{i=1}^{n} x_i^{-1}\right) - \frac{1}{n}\sum_{i=1}^{n} log(x_i) \\
&\text{repeat} \\
\frac{1}{\alpha} &= \frac{1}{\alpha} + \frac{C - \psi(\alpha) + log n\alpha}{\alpha^2\left(\frac{1}{\alpha} - \psi'(\alpha)\right)} \\
&\text{until} \quad \text{convergence} \\
\hat{\alpha} &= \alpha \\
\hat{\beta} &= \frac{n\hat{\alpha}}{\sum_{i=1}^{n} x_i^{-1}}
\end{aligned}$$

[11]return $\hat{\alpha}$ and $\hat{\beta}$, where μ is the mean of the observations, and ν represents the variance of this same set of observations. The α is also an initial value in this second algorithm, and the part in the algorithm that says 'repeat' refers to the continuous generation of a new reciprocal of α until α itself converges to a certain value. The estimator of α is then used to obtain the estimator of β.

Algorithm 2 also has the same mean and variance as that of the first algorithm. It also requires convergence in order to determine the estimators of the parameters, and since one of the parameters to be estimated is a function of the other parameter that also needs to be estimated, this implies that the order on which parameter should be estimated first matters.

3.3 Quantile Plots

The quantile plot is a graphical method for determining if data sets come from populations with a common distribution.

For this report, quantile plots are used for determining which of the algorithms are closest to the empirical distribution. The following quantile plots display each algorithm with the empirical distribution from real data.

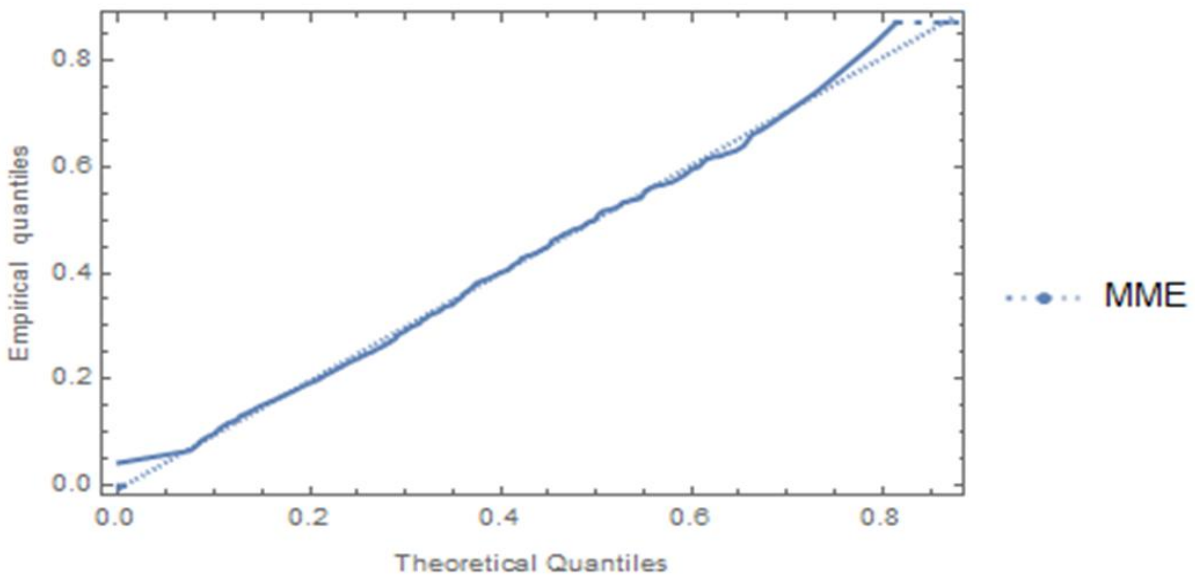


Figure 5: First Algorithm - Real data

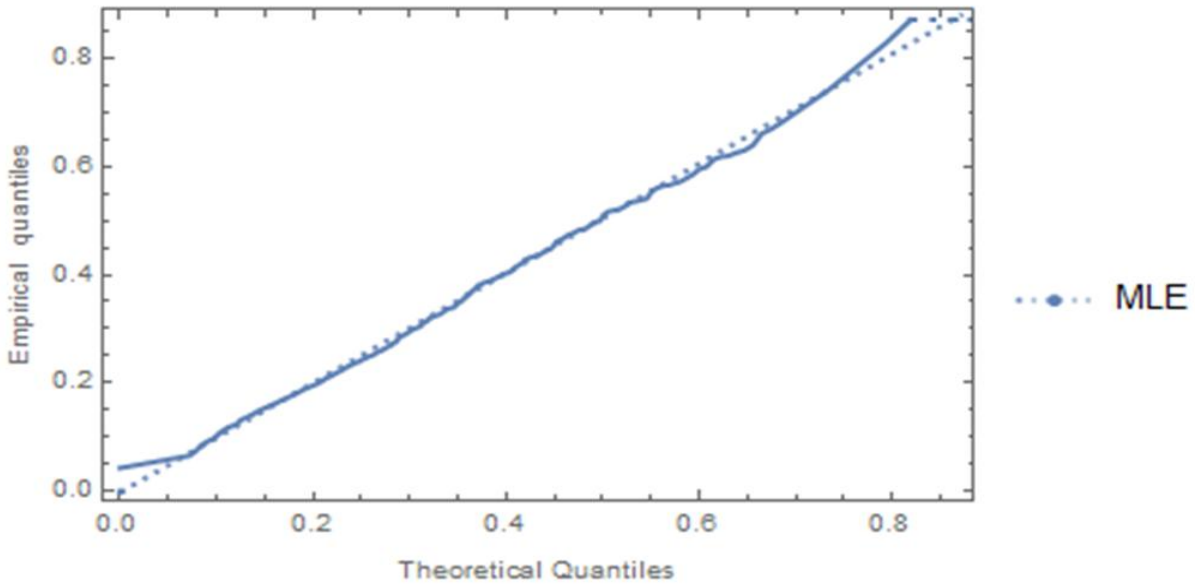


Figure 6: Second Algorithm - Real data

The quantile plots that follow here display each algorithm with the empirical distribution from the simulated data set:

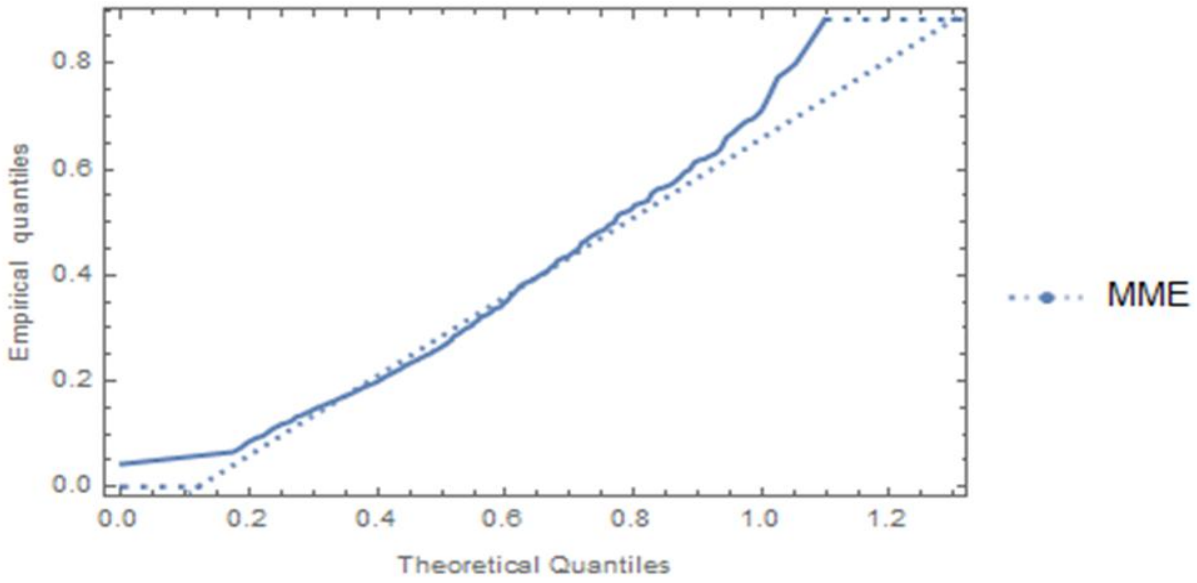


Figure 7: First Algorithm-Simulated Data

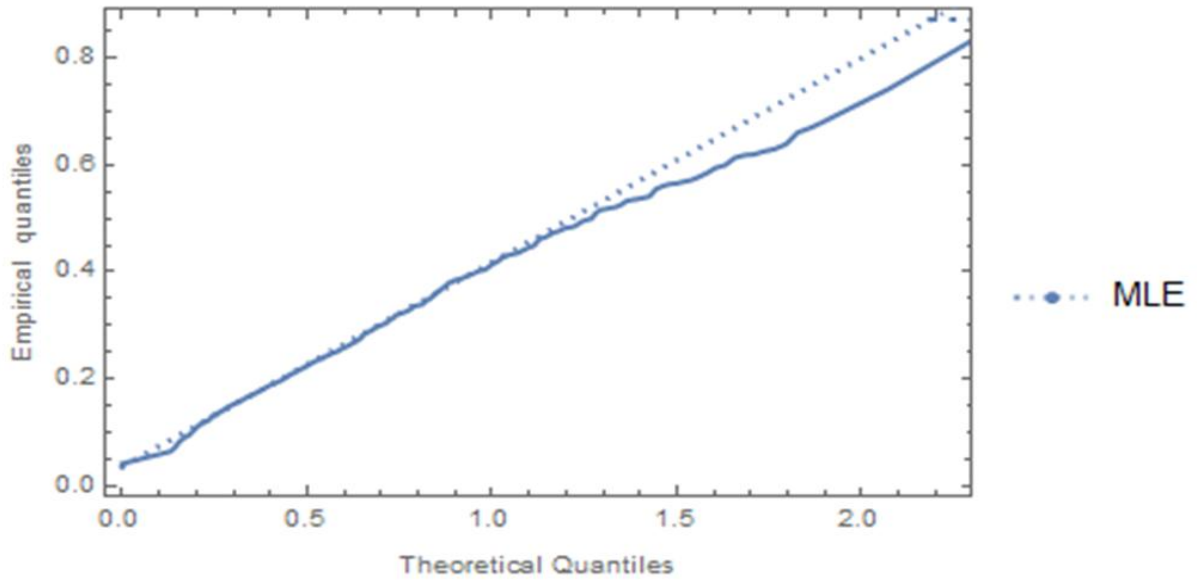


Figure 8: Second Algorithm-Simulated Data

3.4 SAS Coding for Algorithms

A specific dataset is used in order to obtain the estimates of the parameters for each of the two algorithms. This is done by first importing the data into SAS, and then run the two algorithms referring to the imported dataset. In this section, the exact same SAS code that is used to obtain the estimates for the parameters

will be shown explicitly, as well as the estimates obtained for the algorithms. The coding is included in the appendix.

3.5 CDF Comparisons for Real Data

In this section, the CDFs relating to each of the algorithms together with their estimates will be compared. The purpose of this comparison is to observe and determine which of the algortihms fits the data best, i.e. which of the CDF curves of the algorithms is closest to the empirical CDF, or has a stronger relationship with the empirical distribution. After which, an informed decision from the results will be made as to which algorithm performs the best in the estimation the parameters of the inverse gamma distribution, where the empirical distribution is the distribution that is to be used with the data observed for this investigation, and it requires no derivations, unlike what was done for the two algorithms. The manner in which this section is constructed is such that the CDFs for the two different pairs of estimates obtained from the two algorithms will be displayed on a different set of axes for adequate individual assessment, as well as the resulting CDF curve from the empirical distribution. After that, all three of the curves will be displayed on the same set of axes to therefore carefully assess which one of the two algorithms can be used to best estimate the inverse gamma parameters.

3.5.1 Relationship between the 1st algorithm and empirical distribution

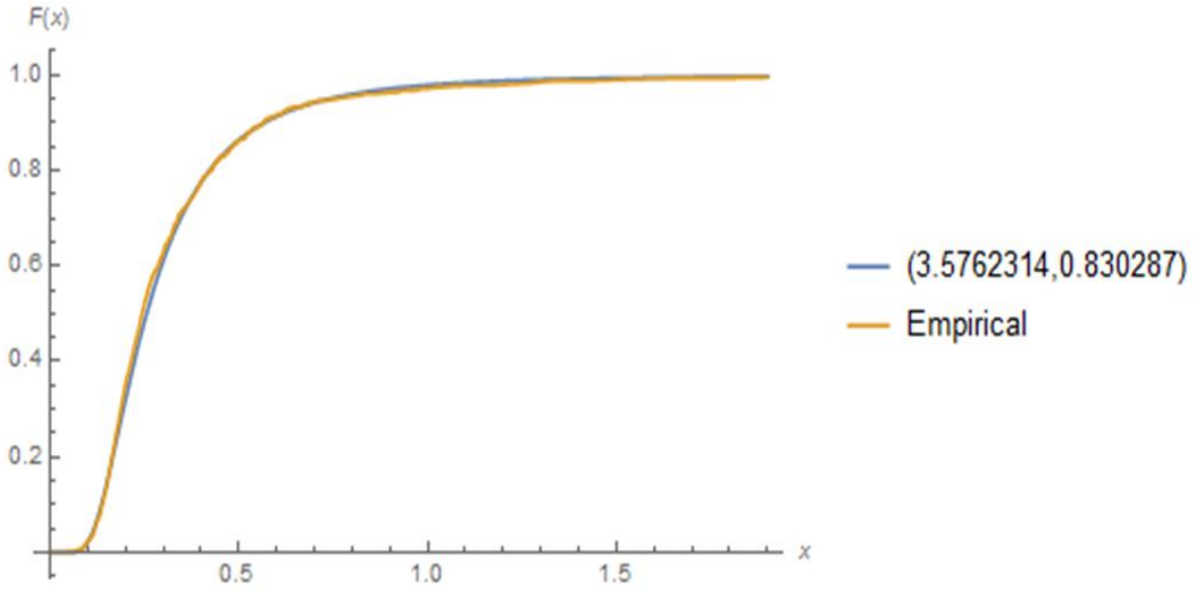


Figure 9: First Algorithm and Empirical Distribution

3.5.2 Relationship between the 2nd algorithm and empirical distribution

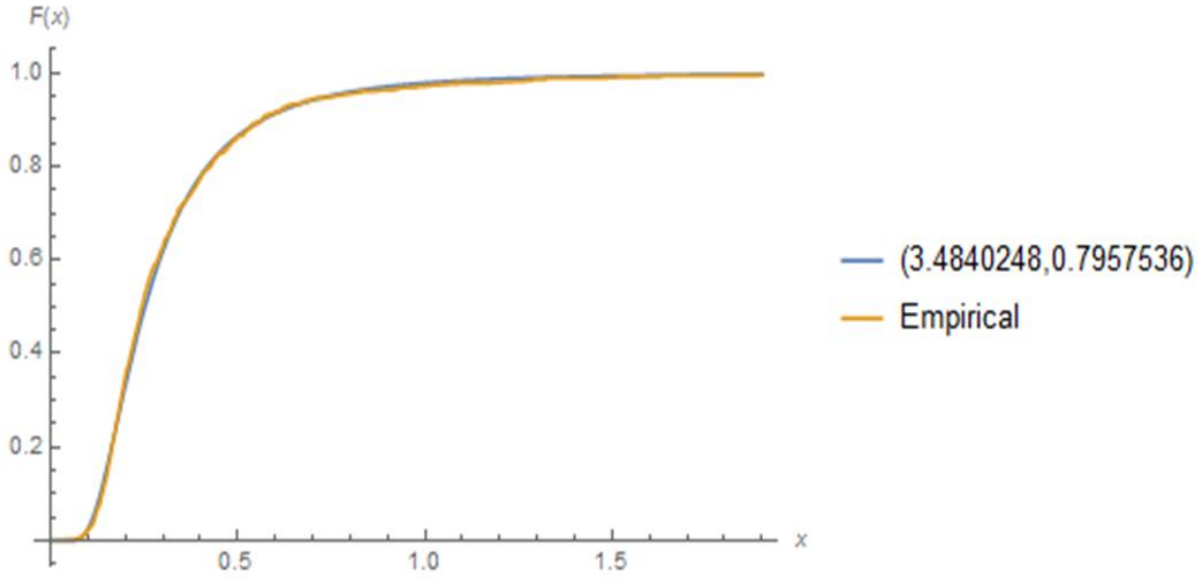


Figure 10: Second Algorithm and Empirical Distribution

3.5.3 Relationship between the 1st and 2nd algorithms with empirical distribution

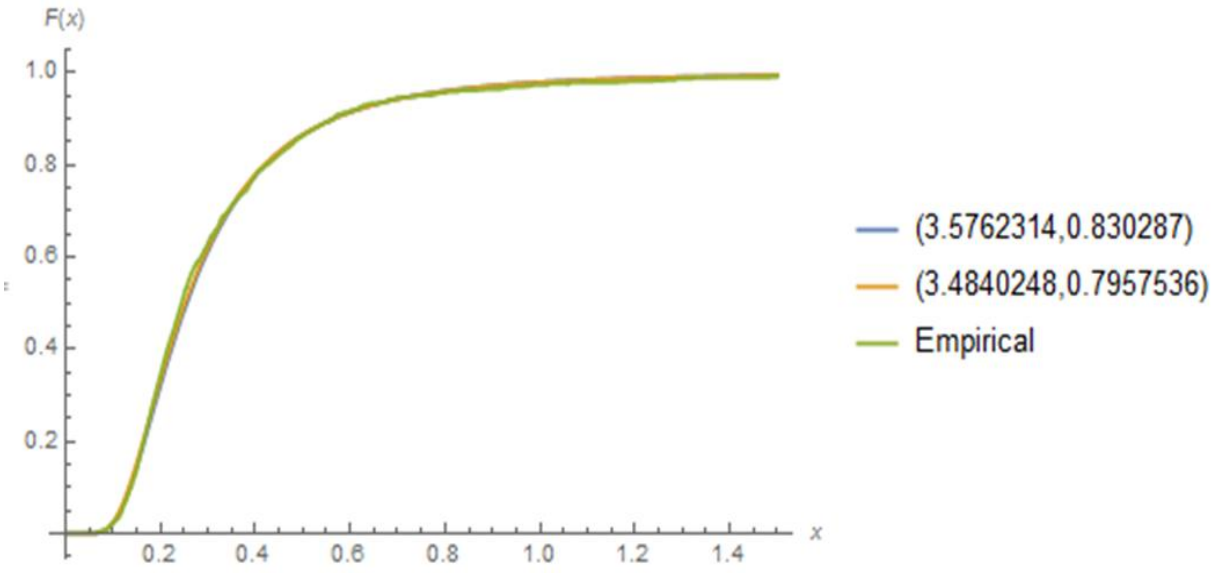


Figure 11: First and Second Algorithm with Empirical Distribution

3.6 CDF Comparisons for Simulated Data

In this section, a similar investigation as above will be inducted with a different dataset. The exact same algorithms that have been used thus far will be used to estimate the two pairs of the inverse gamma parameters using this new dataset.

This part of the investigation also involves determining the relationship between the CDF curves that result from the algorithms using this new dataset, and the CDF curve that results from the new emprical distribution.

The first graphical representation will display the relationship between the first algorithm and the empirical distribution, the one after that will display that of the second algorithm and the empirical distribution.

3.6.1 Relationship between the 1st algorithm and empirical distribution

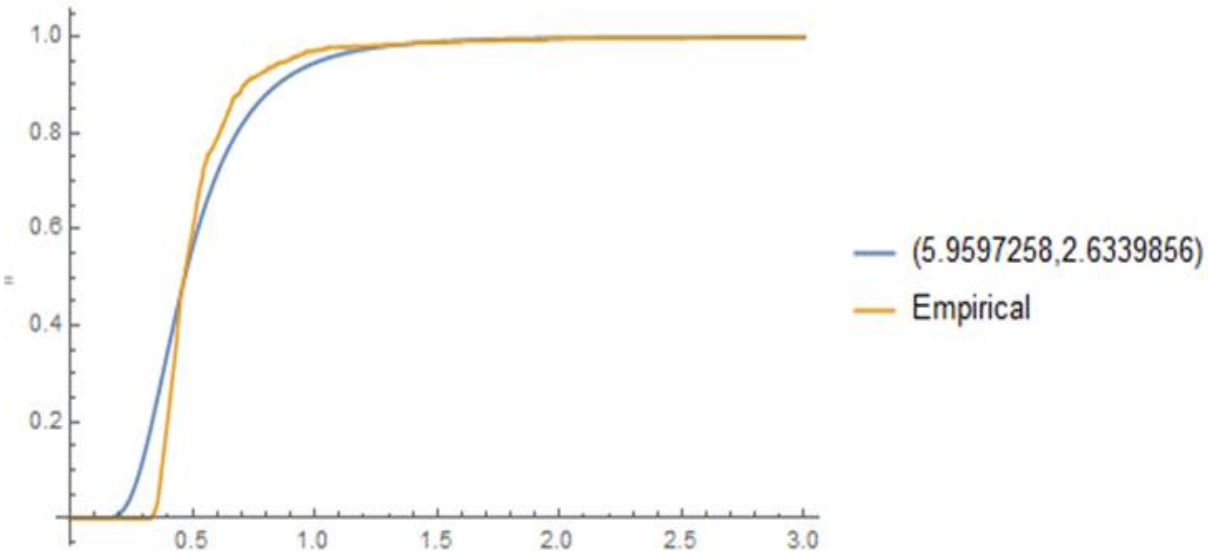


Figure 12: First Algorithm and Empirical Distribution

3.6.2 Relationship between the 2nd algorithm and empirical distribution

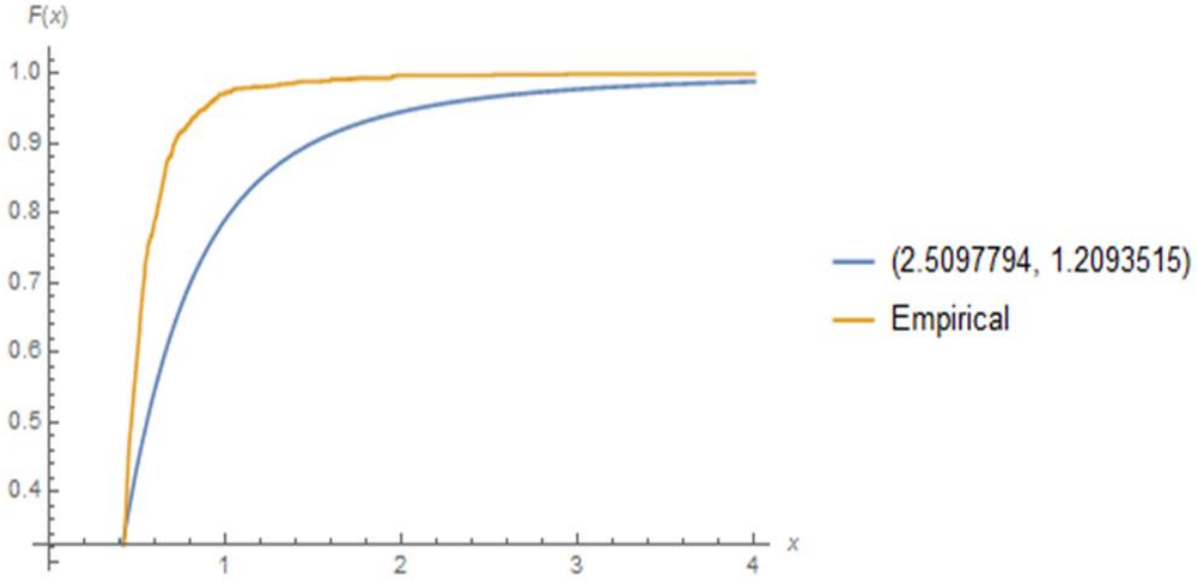


Figure 13: Second Algorithm and Empirical Distribution

3.6.3 Relationship between the 1st and 2nd algorithms with empirical distribution

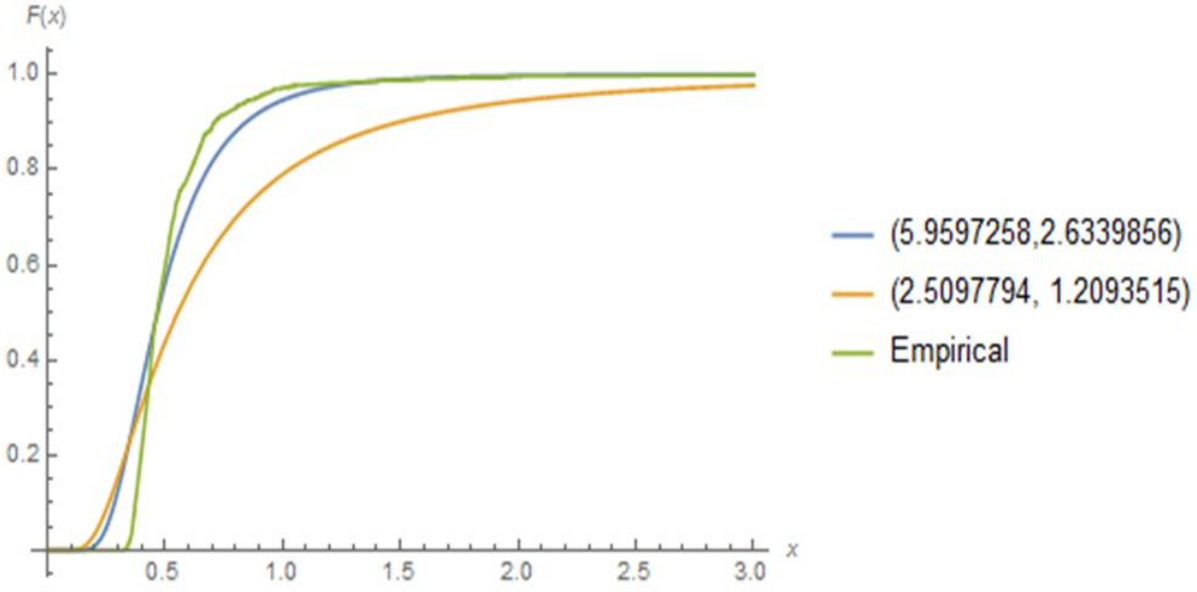


Figure 14: First and Second Algorithm with Empirical Distribution

3.7 Kolmogorov Smirnov Test

3.7.1 Real Data

As mentioned earlier, the Kolmogorov Smirnov test is the Goodness-of-fit test that will be used to determine which of the algorithms best fits the empirical distribution. The procedure that will be followed is one of which the comparison between each algorithm and the empirical distribution will be made, by calculating the maximum difference the CDF obtained from the algorithms and the empirical distribution. Once this has been completed for the two algorithms, the smallest maximum difference amongst the two algorithms is considered to be the closest to the empirical distribution. The maximum difference in this case is the Kolmogorov Smirnov test statistic. Which implies that two values will be compared before drawing up the conclusion.

Therefore from the programming that was performed, the results were as follows:

ALGORITHM	KOLMOGOROV SMIRNOV TEST STATISTIC
Algorithm 1	0.0405563
Algorithm 2	0.0214205

Table 3: Kolmogorv-Smirnov Test Results for the Real Data

from the results obtained, the correct conclusion is that the second algorithm fits the empirical distribution best since it has the smallest maximum difference. This implies that the MLE is more accurate than the MME for the real data.

3.7.2 Simulated Data

The exact same procedure as the one in section 3.7.1 will be followed, but for the simulated data.

Therefore for this data, the results that were obtained from the programming are given by:

ALGORITHM	KOLMOGOROV SMIRNOV TEST STATISTIC
Algorithm 1	0.507532
Algorithm 2	0.266674

Table 4: Kolmogorv-Smirnov Test Results for the Simulated Data

it therefore follows that the SECOND algorithm fits the empirical distribution best since it has the smallest maximum difference. Therefore similarly, the MME is more accurate than the MLE for the simulated data.

4 Conclusion

The methodology followed in this inverse gamma investigation comes from well-known statistical and mathematical principles, i.e. the derivation of the method of moments estimators as well as the derivation of the maximum likelihood estimation. The definitions used to proceed with this investigation also arise from the gamma and inverse gamma properties. The accuracy of each of the algorithms differ strictly depending on the structure of the empirical distribution, on the basis of the aim of the investigation. This also follows from the comparisons conducted between the CDF curves belonging to each individual algorithm together with the CDF curve that results from the empirical distribution. But just a mere observation of the algorithms' CDF curves with the empirical curve is not sufficient to make a decision on the accuracy of the algorithms, even though it might look convincing graphically.

After constructing the curves, there is a possibility that some of the curves representing the algorithms may be very far off as compared to the others, in comparison to the empirical curve. However, in both the scenarios (real data and simulated data), this is not the case. Which is the reason for introducing the kolmogorov smirnov goodness-of-fit test. After the computational procedures have been completed, an appropriate inference can then be made based on the results obtained. These computational methods include SAS and Mathematica.

For the real; data, both the parameters offer good fits, but judging by the ks test statistics it is therefore concluded that the MLE is a more accurate estimation method as compared to the MME simply because the algorithm that follows from the MLE has a smaller maximum difference when compared to the empirical distribution as was observed after conducting the Kolmorogorov-Smirnov goodness-of-fit test.

For the simulated data, the MLE is a more accurate estimation method as compared to the MME since the algorithm following the MLE produces a smaller maximum difference when compared to the empirical distribution as was also observed after conducting the Kolmogorov-Smirnov goodness-of-fit test.

References

[1] Allessio Balleri, Arye Nehorai, and Jian Wang. Maximum likelihood estimation for compound-gaussian clutter with inverse gamma texture. *IEEE Transactions on Aerospace and Electronic Systems*, 43(2), 2007.

[2] SC Choi and R Wette. Maximum likelihood estimation of the parameters of the gamma distribution and their bias. *Technometrics*, 11(4):683–690, 1969.

[3] WJ Cody. An overview of software development for special functions. In *Numerical Analysis*, pages 38–48. Springer, 1976.

[4] John D Cook. Inverse gamma distribution. *Technometrics*, 3(2), October 2008.

[5] David R Cox and E Joyce Snell. A general definition of residuals. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 248–275, 1968.

[6] David E Giles and Hui Feng. Bias of the maximum likelihood estimators of the two-parameter gamma distribution revisited. Technical report, Department of Economics, University of Victoria, 2009.

[7] Gregory J Husak, Joel Michaelsen, and Chris Funk. Use of the gamma distribution to represent monthly rainfall in africa for drought monitoring applications. *International Journal of Climatology*, 27(7):935–944, 2007.

[8] NT Ison, AM Feyerherm, and L Dean Bark. Wet period precipitation and the gamma distribution. *Journal of Applied Meteorology*, 10(4):658–665, 1971.

[9] Josip Juras. Some common features of probability distributions for precipitation. *Theoretical and Applied Climatology*, 49(2):69–76, June 1994.

[10] A Llera and CF Beckmann. Bayesian estimators of the gamma distribution. *arXiv preprint arXiv:1607.03302*, 2016.

[11] A Llera and CF Beckmann. Estimating an inverse gamma distribution. *arXiv preprint arXiv:1605.01019*, 2016.

[12] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[13] Thomas P Minka. Estimating a gamma distribution. *Microsoft Research, Cambridge, UK, Tech. Rep*, 2002.

[14] Daniel S Wilks. *Statistical methods in the atmospheric sciences*, volume 100. Academic press, 2011.

Appendix

Results

Gamma Function

For all complex and non-negative numbers, the gamma function can be defined as:

$$\Gamma(x) = \int_0^\infty y^{x-1} e^{-y} dy$$

If n is a postivive integer:

$$\Gamma(n) = (n-1)!$$

Incomplete Gamma Function

The incomplete gamma function is given by

$$\Gamma(x, b) = \frac{1}{\Gamma(b)} \int_0^x t^{b-1} e^{-t} dt$$

where $\Gamma(b)$ is the gamma function.

Digamma Function

In mathematics, the digamma function is defined as the logarithmic derivative of the gamma function, and is given by:

$$\psi(x) = \frac{d}{dx} ln\Big(\Gamma(x)\Big) = \frac{\Gamma'(x)}{\Gamma(x)}$$

SAS Coding for Algorithms

A specific dataset is used in order to obtain the estimates of the parameters for each of the two algorithms. This is done by first importing the data into SAS, and then run the two algorithms referring to the imported dataset. In this section, the exact same SAS code that is used to obtain the estimates for the parameters will be shown explicitly, as well as the estimates obtained for the algorithms. Thus in the following section, the SAS codes for the two algorithms is displayed.

Algorithm 1-MME

```
proc iml;
use WORK.INVERSEGAMMA;
read all var {x};
n=nrow(x);
mu=mean(x);
v=var(x);
alphaH=((mu**2)/v)+2;
betaH=mu*(((mu**2)/v)+1);
print alphaH betaH;
quit;
```

Algorithm 2-MLE

```
proc iml;
use WORK.INVERSEGAMMA;
read all var {x};
a = 1/x;
b = log(x);
n=nrow(x);
mu=mean(x);
v=var(x);
alpha_start=((mu**2)/v) + 2; *first best guess of what alpha is;
C=-log(sum(a))-mean(b);
d=1/alpha_start;
flag = 0;
do until (flag=0);
alpha_update=d+((C-digamma(alpha_start)+log(n*alpha_start))/(((alpha_start)**2)*
(d-trigamma(alpha_start)))); *new update alpha;
difference = abs(alpha_update-alpha_start);
if difference < 1e-6
then flag = 1; *has it converged yet?;
```

```
alpha_start = 1/d; *replace alpha with new alpha;
end;
*set final converged parameters;
alpha=1/alpha_update;
beta=(n*alpha)/(sum(a));
print alpha_start alpha beta;
quit;
```

Mathematica Coding for graphs

All the graphs that appear in this research were constructed using mathematica.

Introduction Graphs

This section contains the Mathematica code that was used to obtain the graphs that appear in the introduction part of this investigation.

Figure 1

```
Shape (pdf) = Plot[{(0.4^1*x^-2*Exp[-0.4/x])/Gamma[1],
(0.5^1*x^-2*Exp[-0.5/x])/Gamma[1],
(0.6^1*x^-2*Exp[-0.6/x])/Gamma[1],
(0.7^1*x^-2*Exp[-0.7/x])/Gamma[1]},
{x, 0, 1},PlotLegends -> {"(1,0.4)", "(1,0.5)", "(1,0.6)", "(1,0.7)"},
AxesLabel->{x, f[x]}]
```

Figure 2

```
Scale (pdf) = Plot[{(1.5^1*x^-2*Exp[-1.5/x])/Gamma[1],
(1.5^2*x^-3*Exp[-1.5/x])/Gamma[2],
(1.5^3*x^-4*Exp[-1.5/x])/Gamma[3],
(1.5^4*x^-5*Exp[-1.5/x])/Gamma[4]},
{x, 0, 1},PlotLegends -> {"(1,1.5)", "(2,1.5)", "(3,1.5)", "(4,1.5)"}
,AxesLabel->{x, f[x]}]
```

Figure 3

```
Shape (cdf) = Plot[{Gamma[1.4, 0.1/x]/Gamma[1.4],
Gamma[1.4,0.2/x]/Gamma[1.4],
Gamma[1.4, 0.3/x]/Gamma[1.4],
Gamma[1.4, 0.4/x]/Gamma[1.4]},
{x, 0, 1}, PlotLegends -> {"(1.4,0.1)", "(1.4,0.2)", "(1.4,0.3)","(1.4,0.4)"},
AxesLabel->{x, F[x]}]
```

Figure 4

```
Scale (cdf) = Plot[{Gamma[2,0.1/x]/Gamma[2],
Gamma[3,0.1/x]/Gamma[3],
Gamma[4, 0.1/x]/Gamma[4],
Gamma[5, 0.1/x]/Gamma[5]},
{x, 0, 1}, PlotLegends -> {"(2,0.1)", "(3,0.1)", "(4,0.1)", "(5,0.1)"},
AxesLabel->{x, F[x]}]
```

Code for Quantile Plots

Figure 5

```
Data = Import["C:\\Users\\Tmotswiri\\Downloads\\School\\INVERSEGAMMA.csv"];
Emp = EmpiricalDistribution[Data];
quantile1 = QuantilePlot[Emp, InverseGammaDistribution[3.5762314,0.830287],
PlotLegends -> {"Empirical","Inverse Gamma with Method of Moments"},
PlotLabel -> "Quantile plot",
FrameLabel -> {"Theoretical Quantiles", "Empirical quantiles"},
PlotStyle -> Thick];
```

Figure 6

```
Data = Import["C:\\Users\\Tmotswiri\\Downloads\\School\\INVERSEGAMMA.csv"];
Emp = EmpiricalDistribution[Data];
quantile1 = QuantilePlot[Emp, InverseGammaDistribution[3.4840248,0.7957536],
PlotLegends -> {"Empirical","Inverse Gamma with Method of Moments"},
PlotLabel -> "Quantile plot",
FrameLabel -> {"Theoretical Quantiles", "Empirical quantiles"},
```

```
PlotStyle -> Thick];
```

Figure 7

```
Data=Import["C:\\Users\\Tmotswiri\\Desktop\\sim.csv"];
Emp = EmpiricalDistribution[Data];
quantile1 = QuantilePlot[Emp, InverseGammaDistribution[5.9597258, 2.6339856],
PlotLegends -> {"Empirical","Inverse Gamma with Method of Moments"},
PlotLabel -> "Quantile plot",
FrameLabel -> {"Theoretical Quantiles", "Empirical quantiles"},
PlotStyle -> Thick]
```

Figure 8

```
Data=Import["C:\\Users\\Tmotswiri\\Desktop\\sim.csv"];
Emp = EmpiricalDistribution[Data];
quantile1 = QuantilePlot[Emp, InverseGammaDistribution[13.000567, 6.264397],
PlotLegends -> {"Empirical","Inverse Gamma with Method of Moments"},
PlotLabel -> "Quantile plot",
FrameLabel -> {"Theoretical Quantiles", "Empirical quantiles"},
PlotStyle -> Thick]
```

Section 3.5 & 3.6 Graphs

This part includes the coding that was used in mathematica to obtain all the graphs that appear in section 3.3 of this research, which contains the comparison of the three different curves that result from the algorithms with the curve resulting from the empirical distribution.

The first step that was crucial for this part was to use the dataset that determines the empirical distribution when making the comparisons, and in doing that, the dataset used has to be imported.

Importing The Dataset for Real Data

```
Data = Import["C:\\Users\\Tmotswiri\\Downloads\\School\\INVERSEGAMMA.csv"]
```

Figure 9

```
D = EmpiricalDistribution[Data];
```

```
Empirical = Plot[CDF[D, x], {x, 0, 3}];
Algorithm 1 = Plot[Gamma[3.5762314,0.830287/x]/Gamma[3.5762314], {x, 0, 3},
AxesLabel->{x, F[x]}]];
Plot[{Gamma[3.5762314,0.830287/x]/Gamma[3.5762314], CDF[D, x]},
{x, 0, 3},AxesLabel->{x, F[x]}], PlotLegends -> {"(3.5762314,0.830287)",
"Empirical"}]
```

Figure 10

```
D = EmpiricalDistribution[Data];
Empirical = Plot[CDF[D, x], {x, 0, 3},AxesLabel->{x, F[x]}]];
Algorithm 2 = Plot[Gamma[3.4840248,0.7957536/x]/Gamma[3.4840248],
{x, 0, 3},AxesLabel->{x, F[x]}]];
Plot[{Gamma[3.4840248,0.7957536/x]/Gamma[3.4840248], CDF[D, x]},
{x, 0, 3},AxesLabel->{x, F[x]}], PlotLegends -> {"(3.4840248,0.7957536)",
"Empirical"}]
```

Figure 11

```
Data = Import["C:\\Users\\Tmotswiri\\Downloads\\School\\INVERSEGAMMA.csv"];
D = EmpiricalDistribution[Data];
Empirical = Plot[CDF[EmpiricalDistribution[Data], x], {x, 0, 3}];
Algorithm1=Plot[Gamma[3.5762314, 0.830287/x]/Gamma[3.5762314],{x,0,5},
AxesLabel->{x, F[x]}];
Algorithm2=Plot[Gamma[3.4840248,0.7957536/x]/Gamma[3.4840248], {x, 0, 5},
AxesLabel->{x,F[x]}];
Show[Plot[{Gamma[3.5762314,0.830287/x]/Gamma[3.5762314],
Gamma[3.4840248,0.7957536/x]/Gamma[3.4840248],
CDF[EmpiricalDistribution[Data],x]},{x,0,5},
PlotLegends->{"(3.5762314,0.830287)","(3.4840248,0.7957536)","Empirical"},
AxesLabel -> {x, F[x]}]]
```

Kolmogorov Smirnov Test Code

This section displays the code that was used to obtain the Kolmogorov Smirnov test statistics for both the real data and as well as the simulated data.

The results are also displayed in tabel 1 and 2 respectively under sections 3.5.1 and 3.5.2.

Real Data

- First Algorithm

```
Data = Import["C:\\Users\\Tmotswiri\\Downloads\\School\\INVERSEGAMMA.csv"];
D = EmpiricalDistribution[Data];
Empirical = CDF[D,x];
KSTest = MaxValue[Abs[Empirical - CDF[InverseGammaDistribution[3.5762314,0.83028],
x]],x];
```

- Second Algorithm

```
Data = Import["C:\\Users\\Tmotswiri\\Downloads\\School\\INVERSEGAMMA.csv"];
D = EmpiricalDistribution[Data];
Empirical = CDF[D,x];
KSTest = MaxValue[Abs[Empirical - CDF[InverseGammaDistribution[3.4840248, 0.7957536],
x]], x];
```

Simulated Data

- First Algorithm

```
Data = Import["C:\\Users\\Tmotswiri\\Desktop\\sim.csv"];
D = EmpiricalDistribution[Data];
Empirical = CDF[D,x];
KSTest = MaxValue[Abs[Empirical - CDF[InverseGammaDistribution[5.9597258, 2.6339856],
x]], x];
```

- Second Algorithm

```
Data = Import["C:\\Users\\Tmotswiri\\Desktop\\sim.csv"];
D = EmpiricalDistribution[Data];
Empirical = CDF[D,x];
KSTest = MaxValue[Abs[Empirical - CDF[InverseGammaDistribution[2.5097794, 1.2093515],
x]], x];
```

Coding for the Simulated data

Obtaining Parameters for Algorithm 1

```
proc iml ;
use WORK.SIM;
read all var {x};
n=nrow(x);
mu=mean(x);
v=var(x);
alphaH=((mu**2)/v)+2;
betaH=mu*(((mu**2)/v)+1);
print n alphaH betaH;
quit;
```

Obtaining Parameters for Algorithm 2

```
proc iml;
use WORK.SIM; read all into x;
a = 1/x;
b = log(x);
n=nrow(x);
mu=mean(x);
v=var(x);
alpha_start=(mu**2)/v + 2; *first best guess of what alpha is - initial value;
C=-log(sum(a))-mean(b);
flag = 0;
do until (flag=1);
alpha_up_inv=1/alpha_start+((C-digamma(alpha_start)+log(n*alpha_start))/(alpha_start**2*
(1/alpha_start-trigamma(alpha_start)))); *take out abs - new update alpha;
difference = abs(alpha_up_inv-alpha_start);
    if difference < 1e-6    then flag = 1;  *has it converged yet?;
alpha_start = alpha_up_inv; *replace alpha with new alpha;
end;
```

```
*set final converged parameters;
alpha = 1/alpha_up_inv;
beta = n*alpha / sum(1/x);
print alpha beta;
quit;
```

Figure 12

```
Data = Import["C:\\Users\\Tmotswiri\\Desktop\\sim.csv"];
D = EmpiricalDistribution[Data];
Empirical = Plot[CDF[D,x],{x,0,3}];
Algorithm 1 =     Plot[Gamma[5.9597258, 2.6339856/x]/Gamma[5.9597258], {x, 0, 3},
AxesLabel->{x, F[x]}]];
Plot[{Gamma[5.9597258, 2.6339856/x]/Gamma[5.9597258],CDF[D,x]}, {x, 0, 3},
AxesLabel->{x, F[x]}], PlotLegends -> {"(5.9597258, 2.6339856)","Empirical"}]
```

Figure 13

```
Data=Import["C:\\Users\\Tmotswiri\\Desktop\\sim.csv"];
D=EmpiricalDistribution[Data];
Empirical = Plot[CDF[EmpiricalDistribution[Data], x], {x, 0, 3}],
AxesLabel->{x, F[x]}];
Algorithm2=Plot[{Gamma[2.5097794, 1.2093515/x]/Gamma[2.5097794]}, {x, 0, 3},
AxesLabel -> {x, F[x]}];
Show[Plot[{Gamma[2.5097794, 1.2093515/x]/Gamma[2.5097794],
CDF[EmpiricalDistribution[Data],x]},{x,0,4},
PlotLegends-> {"(2.5097794, 1.2093515)", "Empirical"},
AxesLabel -> {x, F[x]}]]
```

Figure 14

```
Data = Import["C:\\Users\\Tmotswiri\\Desktop\\sim.csv"];
D = EmpiricalDistribution[Data];
Empirical = Plot[CDF[D, x], {x, 0, 3}];
Algorithm 1 =  Plot[Gamma[5.9597258,2.6339856/x]/Gamma[5.9597258], {x, 0, 3},
```

```
AxesLabel->{x, F[x]}];
Algorith 2 =  Plot[Gamma[2.5097794, 1.2093515/x]/Gamma[2.5097794], {x, 0, 3}
,AxesLabel->{x, F[x]}];
Plot[{Gamma[5.9597258, 2.6339856/x]/Gamma[5.9597258],
Gamma[2.5097794, 1.2093515/x]/Gamma[2.5097794], CDF[D, x]},
{x, 0, 3},AxesLabel->{x, F[x]},
PlotLegends -> {"(5.9597258,2.6339856)", "(2.5097794, 1.2093515)", "Empirical"}]
```

The perceptions and awareness of Statistics as a profession and the role of Statistics amongst Mathematics teachers

Luwela Nodada 14433852

STK795 Research Report

Submitted in partial fulfillment of the degree BCom(Hons) Statistics

Supervisor: Mr A Swanepoel, Co-supervisor(s): Dr CJ Louw, Ms S Makgai, Dr I Fabris-Rotelli

Department of Statistics, University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

30 October 2017

Abstract

In this study, the perceptions and awareness of Statistics as a discipline amongst high school Mathematics teachers are investigated. A case study is used to determine the perceptions of Mathematics teachers about the role of Statistics in society and the career opportunities within this field. This case study is based on convenient and purposeful sampling from six research sites (the target population) consisting of the teachers invited to the Teachers' Awareness Event as organized by the Department of Statistics and teachers from five of the Kutlwanong Centers in Gauteng. A questionnaire, completed by Mathematics teachers is used as a data collection tool. Teachers who attended the awareness event also completed a follow-up questionnaire on the online platform Qualtrics. The data analysis of the collected information from the sites will be done by means of Excel and SAS software. Conclusions, identifying possible areas for future research and possible recommendations, based on the analyzed results, will flow from this study.

Declaration

I, *Luwela Nodada,* declare that this essay, submitted in partial fulfillment of the degree *BCom(Hons) Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

Luwela Nodada

Mr A Swanepoel

Dr CJ Louw

Ms S Makgai

Dr I Fabris-Rotelli

Date

Acknowledgements

The author would like to thank the supervisors Mr A Swanepoel, Dr CJ Louw and Ms S Makgai as well as Dr I Fabris-Rotelli and the University of Pretoria's Department of Statistics for academic support in the form of resources to conduct research. The author would also like to thank the Department of Education and the directors of the Kutlwanong Centers in Gauteng for their consent and making the study possible. Lastly, the author would like to thank the Center for Artificial Intelligence Research (CAIR) for financial support offered in the form of a study bursary.

Contents

List of Figures

List of Tables

1 Introduction

South Africa has faced many challenges with the education system from early childhood development, basic education to further education [7]. The National Development Plan (NDP), states that the bulk of these challenges is the result of South Africa's history and socio-economic struggles [7]. Due to these challenges, a new education curriculum was developed after the democratic government had been elected in 1994 [29]. Since then, the curriculum has been continuously assessed and adjusted with the intentions of raising the standard of education and increasing the outcomes of the education system. In South Africa, one of the key functions of education reforms is to redress the racial inequalities of the past and to address shortages in subject areas such as science, mathematics and technology [1].

"Being the single most important element of the education system, the quality of a country's teachers is intimately related with the quality of its education system" [26]. The role of teachers in the success of curriculum reform is of key importance. The effectiveness of teachers is affected by a number of factors, including their understanding of the content, their ability to apply the knowledge of the subject, their qualifications and their beliefs around teaching as a profession and Mathematics as a discipline [24]. Teachers' skills, attitudes and understanding of the content, all play a fundamental role in their acceptance of adjustments in a new curriculum. Furthermore, Gal *et al.* [12] believe that the attitudes and belief issues in statistics will become increasingly relevant as more students, at all educational levels, experience statistical education. Gal *et al.* [12] investigated the beliefs and attitudes of students towards statistics as a subject of learning, and though they acknowledge that attitudes and beliefs are influenced by personal ideas about challenges related to the subject matter, they conclude that various factors influence those ideas, of which, classroom interaction and teachers' beliefs and attitudes about statistics play an important role. Society's beliefs and attitudes about statistics as discipline and profession vary in accuracy and understanding. Hunter [14] describes a professional in statistics as, "A person whose everyday work consists of making sense of data and, equally as important, in the planning for information laden data".

Statistics is a very vibrant and valuable field of study and giving learners an introduction of it at school has been seen as a positive aspect. This research project aims to investigate and establish teachers' perceptions of statistics as a discipline and the possibilities of statistics as a profession. The following section states the primary and secondary research questions.

1.1 Research questions

The main research question is:

- What are teachers' perceptions of statistics as a discipline?

The secondary research questions are:

- How aware are teachers about statistics as a profession?
- How do teachers perceive the included statistical content within the mathematics curriculum?

1.2 Research Objectives

It would then follow from the above questions that the research objectives of this study would be to:

- Determine teachers' perceptions of statistics as a discipline;
- determine teachers' awareness of statistics as a profession;
- determine teachers' awareness of employment and career opportunities in the statistical field, and lastly,
- determine teachers' perception towards the inclusion of statistical content in the mathematics curriculum;

The Statistics discipline is broad and it offers career opportunities in various fields. Improving teachers' awareness and perceptions about statistics as a discipline and career choice could yield great benefits for both the learners and the South African economy. We need learners to pursue careers in statistics so as to interpret the large volumes of data, referred to as Big Data, that is now being collected by so many companies. South Africa faces various challenges with regards to unemployment [7], and an increased awareness about statistics as a profession and career choice could help to alleviate some of these challenges.

2 Literature Review

The role of Statistics in society has been the topic of many studies. Statistics offers various career choices in various fields of study. However, this does not translate positively to society's appreciation of

statisticians. To the community at large, a statistician can be someone who captures data, a sports number librarian, or someone running a survey [14]. Hunter [14] describes professional statisticians as persons who advance statistics through research, communicates its arts through journals and meetings, educates and defines statistics to society, serves society and serves its members. Statisticians spend their careers using mathematical and graphical tools to analyze and structure data. They employ processes of inferences to solve problems. Statisticians often work in service of other professionals and government offices, however, the contribution of statisticians to society is not one that is appreciated by society nor one that improves society's attitudes towards statistics as a discipline [14]. Hunter [14] suggests that a professional statistics accreditation could serve as a means to further increase the society's appreciation and perceptions of statistics as a profession.

2.1 Statistics as a Profession

A broad definition of statisticians includes statisticians in education and academia. These are the educators and the researchers who not only transfer the knowledge of statistics to society, but also work to improve the theory and science of modern statistics [14]. The American Bureau of Labor Statistics [1] in 2016, listed Scientific Research and Development Services as the industry with the highest levels of employment of statisticians. Statisticians who choose a career in research are not limited to the field of academics. The World of Statistics Organization describes researchers as providers of insight by means of surveys, case studies and technological advancements in areas of government, social sciences, education and medicine, just to name a few. These researchers are known as survey statisticians [2]. Hunter [14] describes statisticians in Research and Academics as the builders of statistical tools and theory. Statisticians form an integral part of varying industries. The American Bureau of Labor Statistics further listed the top industries with the highest concentration of employed statisticians as Monetary Authorities (Central Banks), Scientific Research and Development Services, Pharmaceutical and Medicine Manufacturing and Management, Scientific and Technical Consulting Services. These statisticians are collectively known as applied statisticians.
This emphasizes the diverse nature of a profession in statistics and the importance of statistics in society.

2.2 Statistics in the Mathematics Curriculum

Statistics education has faced various challenges at most levels of learning [12]. Both students and teachers have experienced challenges in awareness and understanding of statistics. A study conducted by Gal *et al.* [12] focusing on students' attitudes and beliefs of Statistics, found that most students were unaware of the benefits of statistics education and the opportunities associated with a statistics qualification. Further-

[1] https://www.bls.gov/oes/current/oes152041.htm
[2] http://www.worldofstatistics.org/statistics-as-a-career/statisticians-at-work/

more, students believed that statistics is a challenging subject. Similarly, there have been several studies investigating the teaching of mathematics. These studies focus on the abilities of teachers to effectively teach mathematics and whether enough is done to equip teachers with the necessary skills required to understand and teach content of a changing mathematics curriculum [12]. Course curriculum needs to be revised and adjusted to suit the needs of learners and teachers, however the implementation of a new curriculum and/or adjustments to subject content is a challenge [20]. There have been very few studies have focused on teachers' attitudes and beliefs of statistics. South Africa has seen many challenges with its educational reform programs due to factors such as the number of teachers in the fields of mathematics and science, the skill sets of teachers, performance of learners and limited resources. Outcomes based education (OBE) forms the basis of the new curriculum introduced in post-apartheid South Africa. The re-engineering of the education system in South Africa, according to Carnoy and Chisholm [3], was in three waves whose main focus in the curriculum was on the assessment of outcomes. The latest curriculum change, namely the National Curriculum and Assessment Policy Statements (CAPS) [7], was introduced in 2012.

To highlight the importance of statistics education in South Africa, Statistics South Africa (Stats SA), together with the South African Statistical Association (SASA) and the Association for Mathematics Education of South Africa (AMESA), co-hosted the 6th International Conference on the Teaching of Statistics (ICOTS-6) held in Cape Town in 2002 [3]. According to Statistics SA, the project "recognized the cross curricula need for data handling as an anticipated outcome, resulting in vast amounts of statistical material being included throughout the various phases of the new school curriculum". The conference led to the development of the maths4stats project which seeks to address the need for statistical development in South Africa. The primary goal of the maths4stats project is to assist towards Stats SA's objectives of developing national statistical capacity and promoting statistical literacy. The project has faced challenges in realizing its goals, most of which, Stats SA, attributes to the history of the South African education system. Furthermore, the Department of Science of Technology together with the Department of Education implemented programs to help teachers understand curriculum changes and new teaching practices [7], however limited results have been observed [1]. This raises the question "Why?" Why are there difficulties and challenges that arise with the introduction of new content outside of the training of development?

Quality teachers are the single most important variable which influences on pupil learning [26]. Teachers' perceptions of statistics could influence statistics education thus the implementation of curriculum reform must take into consideration the perceptions and attitudes of teachers. Mohammed and Jones [21] caution against thinking that teachers are without will of their own and can be manipulated. Thus, it is important to

[3] http://www.statssa.gov.za/?page_id=3500

find out what perceptions are constructed around curriculum reform and content implementation. Bantwini [1] found that teachers find curriculum reform to be high paperwork overload and the reform is viewed as a burden. The perception is born from the fact that teachers already feel overloaded by the learner to teacher ratio thus additional content changes are met with a lot of friction from the teachers [1]. Each teacher attached to curriculum reform, acts in his or her way to understand curriculum. Research conducted by Prescott and Cavanagha [23], revealed that teachers attitudes towards mathematics influence teaching practicing. An investigation conducted by Levpuscek and Zupancic [18] showed that teachers' beliefs contribute to developing attitudes about mathematics. Ayres and McCormick [16] found that teachers are generally unhappy with various content in mathematics which may result in the inability to cope with the demands of the mathematics curriculum. Garfield [13] found that teachers lack reasoning in statistics and are of the belief that statistics is not valuable in society. This again raises the question "why"? Why do teachers struggle with statistical concepts? Why do they struggle to teach statistical concepts and why do they view statistics as a subject of low importance in society?

2.3 The importance of Statistics

"Statistical thinking will one day be necessary for efficient citizenship as the ability to read and write" - H. G Wells

The claim by H.G Wells, towards the end of the nineteenth century, may have seemed erroneous at the time, but in recent times, what once appeared to be a false claim, can now be labeled a prophecy in the field of statistics. The evidence of which, lies in the increasing emphasis on statistics in the mathematics curriculum, the increasing demand for professional statisticians in various fields of employments and the increase in society's need to understand, interpret and implement solutions deduced from statistical information [27]. The trend of increased interest in statistical education was evident, first, in more developed societies such as the United States of America and Australia, whose curricula statements are largely based on the requirement for students to understand the social uses of statistics and the impact statistics has on society [27]. Similarly to H.G Wells, Gal [11], states that statistical literacy is an expected ability in all citizens in information-laden societies. Gal [11] further defines statistical literacy as people's ability to understand and express relevant opinions obtained from statistical information. As societies evolve, the need for statistical reasoning expresses itself in different forms. Jane M. Watson [27] states that everyday news media, which presents report on a wide range of subjects, from health to politics, further supports the growing necessity for statistical understanding in society. Statistics is a subject of importance, not only to those who choose it as a profession, but even more so, to regular citizens who are consumers of statistics.

3 Methodology

3.1 Research Design

This study adopted a mixed method design that entails both quantitative and qualitative research approaches. This research followed an exploratory case study design in the form of a questionnaire that was used to profile teachers. The exploratory case study design method refers to a type of case study that is best used in research that is aimed at exploring situations in which the phenomenon in question has no clear, single set of outcomes [30].

Quantitative approach

Creswell [9] defines a quantitative as an approach that tests objective theories by examining the relationship among variables. These variables can be measured numerically so that the data can be analyzed using statistical procedures [9]. In simple terms, quantitative research attempts to establish the measurements of something [8].

Qualitative approach

Qualitative research is often aimed at investigating research questions that seek to examine the meaning and existence of social phenomena. This research will be administered using a questionnaire to detect teachers' attitudes towards statistics, their understanding of the content, prior skills relating to basic statistics and current support available to them in the form of training and skills development.

3.2 Data Collection Method

The data collection method is in the form of a semi structured questionnaire. The questionnaire was piloted at the Teachers' Awareness Event that was hosted by the University of Pretoria on the 15th of March 2017. The questionnaires took an average of fifteen minutes to complete and were completely anonymous. The responses to the questionnaire revealed that some of the questions were ambiguous and required to be rephrased in order to avoid confusing the participants and to establish more accurate results. A follow up digital questionnaire on Qualtrics, an online survey software, was later forwarded to the participants of the awareness event to establish any changes in awareness and perception within the sample. Lastly, a revised version of the Statistics Awareness Event questionnaire, was drafted and sent to be filled out by teachers at the Kutlwanong Centers in Gauteng.

3.3 Research Participants

The Statistics Awareness Event consisted of twenty-four high school mathematics teachers from various schools of diverse socio-economic schools within the Gauteng region. The teachers were profiled in terms of

their ages and number of years teaching mathematics. This sample of teachers consisted of teachers between the ages of 31 and 60 years old. Two of the teachers were within the 31-35 years age group, three within the 41-45 group, five within the 46-50 group, ten within 51-55 group, three within 56-60 age group and only one questionnaire in which a respondent did not specify their age bracket. The questionnaire also profiled teachers according to the quintile their schools are ranked. Within the South African education system, schools are divided into five categories. These categories are known as quintiles and are ranked from poorest (quintile 1) to wealthiest (quintile 5) [2].

The second population consisted of high school mathematics teachers from the five Kutlwanong Centers in Gauteng. The sample from the Kutlwanong Centers consisted on thirty-one teachers. Thirteen of the teachers were within the 31-35 age group, six within the 41-45 group, two within the 51-55 group, three within 56-60 group and seven in the older than sixty age group. The overall population consisted of fifty-five mathematics teacher from the Gauteng region.

3.4 Research Instrument

Jane M. Watson [28] developed a teacher profiling tool with the aim of addressing issues of teacher competencies and curriculum changes, particularly for in mathematics education. The profiling tool exhibits flexibility that previous tools did not provide and for this reason, it has been adapted for many studies regarding Mathematics education [29]. Similarly, this study adopted and adapted the tool, in the form of a questionnaire, in order to profile Grade 10 - Grade 12 mathematics teachers.

The first research instrument used is a case study questionnaire divided into three sections, namely, biographical information, perceptions about statistics (from the teachers' point of view) and learner perceptions (as inferred by the teachers).

The second research instrument is a follow-up online questionnaire on the Qualtrics package. Similarly, the questionnaire is divided into the 3 sections as mentioned above.

3.5 Statistical Theory

3.5.1 The Chi-Squared (χ^2) Distribution

The Chi-Squared (χ^2) distribution *(see Figure 1)* dates as far back is the mid 1800's [15]. It is believed that is was derived by Bienaynme in 1838. Sheynin [25] theorized that it was first formulated by Ernst Karl Abbe in 1863 and that a general expression for the distribution was derived by Boltzman in 1881 but it was not until Karl Pearson published a seminal paper in 1900 which introduced the χ^2 not only as a distribution but also as a statistic and a statistical test, did the discovery of the χ^2 be impactful [15]. It is now, one of the most important and most widely used distributions in statistical theory and inference.

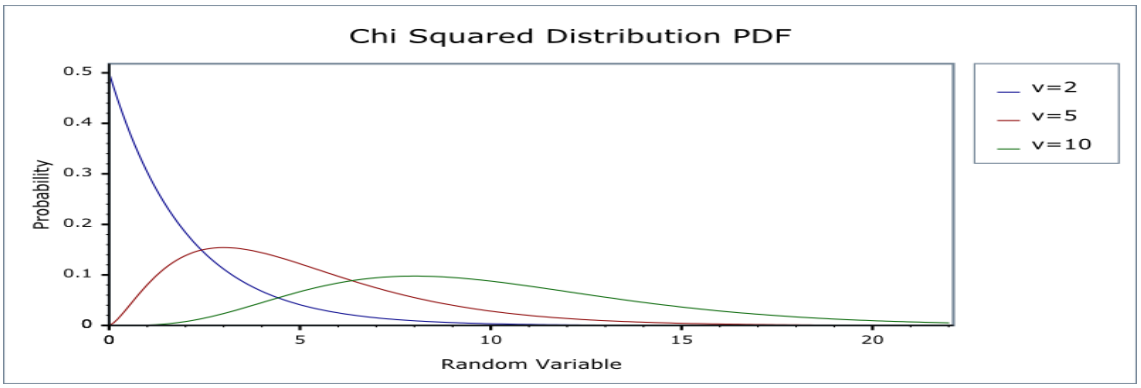


Figure 1: The χ^2distribution where v represents the degrees of freedom
Source:
http://www.boost.org/doc/libs/1_38_0/libs/math/doc/sf_and_dist/html/math_toolkit/dist/dist_ref/dists/chi_squared_dist.html

A random variable X is said to have a χ^2 distribution with n degree of freedom if it is absolutely continuous with density:

$$f(x) = \begin{cases} 0 & x \leq 0 \\ \Gamma\left(\frac{n}{2}\right)^{-1} 2^{-\frac{n}{2}} x^{\frac{n}{2}-1} e^{-\frac{x}{2}} & x > 0 \end{cases} \tag{1}$$

where $n \geq 1$, for cases where $\Gamma(.)$ represents the Gamma function and random variable $X \sim \chi^2(\cdot)$.
The χ^2 is closely related to both the Gamma distribution and the Normal Distribution. The Gamma distribution, with a mean of n and variance of $2n$, is equal to the χ^2 distribution with n degrees of freedom. Furthermore, the χ^2 distribution exhibits properties of a normal distribution. The sample variance, S^2, of a random sample from a normally distributed population has the χ^2 sample distribution. This means that if $X_1, \ldots, X_n$ are independent and identically distributed normal variables with the population variance σ^2, then

Remark 1.

$$\frac{n-1}{\sigma^2}.S^2 = \frac{1}{\sigma^2}((X_1 - \bar{X})^2 + \ldots + (X_n - \bar{X})^2) = \frac{1}{\sigma^2}(X_1^2 + X_2^2 + \ldots + X_n^2 - n\bar{X}^2)$$

has χ^2 distributed random variables with $n-1$ degrees of freedom. This result follows from general property of normality which is illustrated below.

Remark: Let $\mathbf{X} = (\mathrm{X}_1, \ldots, \mathrm{X}_\mathrm{n})$ be a normal vector with dimension n, such that$\mathrm{X}_1, \ldots, \mathrm{X}_\mathrm{n}$ $a$re normally distributed independent random variable with a mean of *0* and a variance of *1*, then

$$|\mathbf{X}|^2 = \mathrm{X}_1^2 + \ldots + \mathrm{X}_\mathrm{n}^2$$

has a χ^2 distribution with n degrees of freedom. From this result, if the mean values of the components of $\mathbf{X}$ are non-zero, then$|\mathbf{X}|^2$ has a non-central χ^2 distribution with n degrees of freedom and parameter, $\sqrt{\bar{\mathbf{X}}}$, that is also non-central. Thus the χ^2 distribution with a non-centrality parameter that is equal to 0 is known as the central χ^2 distribution.

Many tests have a χ^2distribution or an asymptotic χ^2 distribution. For example, the goodness of fit χ^2 tests are based on Pearson's χ^2 statistic which, under an appropriate null hypothesis, has a χ^2 distribution. The Friedman test statistic and likelihood ratio tests are also based on a test statistic that is asymptotically χ^2 distributed [15].

Non-Central χ^2 distributions are used for calculating the power function of tests based on the quadratic forms of the normal or asymptotic normal statistics.

3.5.2 Pearson's χ^2 Goodness of Fit Test: Improvements

The Goodness of fit test was first discovered in 1900 by Karl Pearson. Based on Pearson's publication, the limit distribution of the χ^2 statistic would be the same if the null hypothesis were replaced by estimates based on a sample. Pearson's Sum, also known as, Pearson's χ^2 Tests Statistic is written as follows:

$$\chi^2 = \chi_n^2(\theta) = \sum_{i=1}^{r} \frac{(v_i - np_i(\theta))^2}{np_i(\theta)} \tag{2}$$

where v_i is the observed frequency, $np_i(\theta)$ represents the expected frequency and r is the number of rows, such that Pearson's Sum (2) can be simplified as follows:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{Ei}$$

For the number of observations, $n \to \infty$, Pearson's Sum (2), for the simple null hypothesis specifying the true value of θ, will follow χ^2 distribution with $r - 1$ degrees of freedom. Pearson believed that the limit of the distribution of his χ^2 statistic would be the same if unknown parameters of the null hypothesis were replaced by estimates based on a sample [15]. This, of course, was an error, which inspired further development in the theory of the χ^2 test statistic. An article by Chernoff and Lehmann [5] published in The Annals of Mathematical Statistics Journal in 1954, investigated Pearson's test statistic. Their findings showed that the

test statistic does not have a limiting χ^2 distribution but that, it is in actual fact, statistically larger than it would be expected to be under the χ^2 theory. They showed that replacing the parameters in Pearson's sum (1), with maximum likelihood estimates based on non-grouped data would significantly alter the limit distribution [15]. This means that, the limit distribution will follow a distribution that generally depends on the unknown parameters and thus cannot be used for testing [15]. In a paper published by the Biometrika Journal 1977, Molinari [22] investigated the χ^2 test under the null hypothesis when parameters are estimated by the method of moments. He derived a limit derived the distribution of the limit of Pearson's Sum for moment type estimates that are based on raw data. Similarly to the case of maximum likelihood estimators, Molinari's limit distribution depends on unknown parameters [15]. Dahiya and Gurland [15] derived a modified version of Pearson's χ^2 test statistic so that the limit distribution does not depend on unknown parameters but instead on the null hypothesis. The modifications to Pearson's sum utilized estimators that are based on ungrouped data in deriving the test statistic as well as determining the class intervals [10]. Dahiya and Gurland [10] illustrate that, for continuous distributions with locator and scale parameters, the distribution under the null hypothesis of the modified test statistic does not depend on unknown parameters when estimated by the sample mean and the variance. They further developed a table of certain percentage points for the modified statistic in order to facilitate its use for normality testing [10].

Cochran [6], stated that χ^2 tests often do not indicate significant results when the null hypothesis is false. He further suggested that using a single degree of freedom, or a group of degrees of freedom from the total test statistic, will yield a more accurate and appropriate test [6]. However, it was Ronald Fisher, in 1925, who first showed that the number of degrees of freedom of the Pearson's χ^2 test must be decreased by the number of parameters estimated by the sample [15]. Fisher's result is only true if and only if the parameters are estimated by grouped data or by any asymptotically equivalent procedure. This resulted in what is known as the Pearson-Fisher χ^2 test. The Pearson-Fisher test is a Pearson test where the unknown parameters replaced by grouped data estimates are as follows:

$$\chi^2_n(\hat{\theta}_n) = \sum_{i=1}^{r} \frac{(v_i - np_i(\hat{\theta}_n))^2}{np_i(\hat{\theta}_n)} \tag{3}$$

where $\hat{\theta}_n$ represents the grouped data estimates.

3.5.3 Chi-Squared (χ^2) Test

The χ^2 test is the most widely used of all the non-parametric test of significance and is particularly of use for tests involving nominal data [8]. The χ^2 test, tests for significant differences between the observed distribution of data and the expected distribution as derived from the null hypothesis. The null hypothesis

is established from the expected frequency of data within a category. The deviations of the hypothesized frequencies from the actual frequencies are compared.

The null hypothesis states that there are no differences in expected and observed frequencies:

$H_0 : O_{ij} = E_{ij}$, where O_{ij} represents the number of cases categorized in the ijth cell and, E_{ij} is expected frequency of cases under the null hypothesis to be categorized in the ijth cell.

The null hypothesis is investigated by means of the following test statistic:

$$\chi^2 = \sum_{i}^{k} \sum_{j}^{k} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{4}$$

where k represents the number of categories.

The statistic expresses the extent of difference between the observed and expected frequencies and is thus calculated by means of a count or frequency as opposed to a percentage of cases in a categorized cell [8]. The greater the differences, that is, a large χ^2 value, the less the differences can be explained by "chance" [8]. The degrees of freedom associated with the test statistic determine the distribution of the χ^2. The degrees of freedom are calculated as follows:

$$df = (r-1)(c-1)$$

where r and c represent the number of rows and columns, respectively. The χ^2 has restrictions [8]. For $df = 1$, the expected frequency in each cell should be at least five, and for a case where $df > 1$, the χ^2 test should only be used if and only if, at most twenty percent of the expected frequencies are smaller than five. Thus, the χ^2 test is not suited for small sample sizes.

3.5.4 Fisher's Exact

Fisher's Exact is an augmented version of the χ^2 test that is more accurate for small sample sizes. Like the χ^2 test, Fisher's exact is used when two or more nominal variables are to be investigated for statistically significant differences. As the name suggests, Fisher's exact calculates an exact *p-value* the exact probability of the table of observed cell frequencies based on two main assumptions:

1. The null hypothesis of independence is true,
2. The number of rows and columns are fixed by the experiment.

The *p-value* is calculated using the exact probability of observed cell frequencies as follows:

	p	q	row total
x	a	b	a+b
y	c	d	c+d
Col Total	a+c	b+d	a+b+c+d=n

Table 1: Two by two contingency table

$$p - value = \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{n}{a+c}} \tag{5}$$

Equation 5 calculates an exact hypergeometric probability for contingency Table 1. The hypergeometric formula is a conditional probability due to the requirement that the row and column totals remain constant. Fisher's exact test is exact for as long as the row and column totals are fixed. For this reason, the test can be used regardless of the characteristics of the sample. Ludbrook [19] concluded that Fisher's exact test is the best method for analyzing conditional experimental designs. However, Fisher's exact does pose the problem of giving the exact answer to the wrong question[4]. Ludbrook [19] claimed that Fisher designed his exact test for a specific experiment, thus the conditional experimental design must be followed in order to effectively use Fisher's exact.

3.5.5 Chernoff Faces

Chernoff Faces were developed in response to a cluster analysis problem [15]. Cluster analysis includes a variety of methods of which the appropriate analysis method depends on the nature of the data. The method called "Chernoff faces" involves a computer program that draws a caricature of a face when given 18 numbers between 0 and 1. The grouping of faces serves as a preliminary method of clustering and recognizing which features are important in the clustering [4]. The numbers used in this method represent features of the face. Faces were used in this method as a way to comprehend data where roles of various factors are understood. Originally, Faces were designed in order to simplify and to further represent which variables are important and which variables interact with one another. The method of Faces is suited for high dimensions; however, it does not always handle a great number of faces well except in the case of a time series analysis in which the faces will appear in succession. The method of Faces can be further augmented to handle more than 18 variables by means of using a pair of faces to represent data. One of the main benefits of Chernoff Faces is the human brain's repose to caricatures and cartoon faces. It is apparent that the human brain recognizes faces in a different part of the brain than that which handles other forms of geometric data thus, small changes in facial features are easily recognized, understood and remembered, which is not the case with

[4]https://www.graphpad.com/guides/prism/7/statistics/stat_chi-square_or_fishers_test.htm?toc=0&printWindow

other conventional methods of graphical data analysis and representation. However, the method of faces is somewhat limited to data summarizing and representation and is not likely to be of much use for calculations [4].

4 Data Analysis

Qualitative research methodologies have been used in various fields and disciplines since as early as the 19th century [8]. The problem most researchers face is that results obtained from qualitative data cannot be generalized for a larger population. However, there are analysis methods and techniques that can be applied to qualitative data to yield conclusive and trustworthy results. Qualitative data analysis generally follows two distinct approaches, namely, grounded theory analysis and framework analysis.

Grounded theory analysis includes an inductive method that allows for social theories to be generated from data [17]. This means that concepts and relationships are developed directly from the data analysis providing a set of testable hypotheses that will form theories that provide further understanding of social phenomena. On the other hand, framework analysis was developed for the purposes of applied research which is research that is aimed at understanding and interpreting information to provide outcomes and suitable recommendations within a relatively short period of time [17].

Framework analysis identifies themes in data and further compares and contrasts the themes in search of patterns, associations and explanations as a means of interpreting the data. Framework analysis can otherwise be referred to as exploratory data analysis (EDA) which gives allowance to the researcher to respond to patterns that are revealed during the analysis process [8]. EDA emphasizes the use of visual representations and graphical techniques as a means to summarize and display data. There are various techniques that can be used to display data such as frequency tables, histograms and Chernoff faces. Cross tabulation, which is a technique that is useful for comparing two or more qualitative variables, is another means of examining data. Generally, it is applied in order to represent demographical variables against the study's target variables [8]. Furthermore, descriptive and inferential analysis methods can be effective in examining qualitative data *(see Figure 2)*. Data is used to answer the following questions:

1. Are teachers aware of statistics as a profession?
2. How do teachers perceive the inclusion of statistical content in the Mathematics curriculum?
3. Does the number of years of teaching Mathematics affect the perception of statistics?

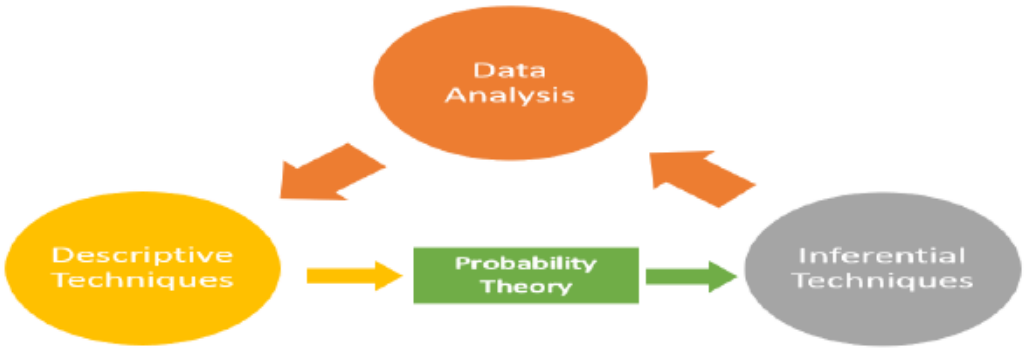


Figure 2: Graphical breakdown of the data analysis process

Descriptive Statistics

Cooper and Schindler [8] describe Exploratory Data Analysis (EDA) as a set of analysis techniques and as a perspective. EDA does not follow a rigid structure in terms of data analysis methods and can be further described as a flexible analysis technique. Flexibility is one of the most important attributes of Exploratory Data Analysis. In the context of EDA, Cooper and Schindler [8] emphasize the importance of visual representations and graphical techniques over summary statistics. For this reason, graphical methods are used to summarize and further analyzing patterns that emerged from the data.

Inferential Statistics

Under the topic of inferential statistical analysis, the method of hypothesis testing will be applied to address some of the research questions. According to Cooper and Schindler [8], hypothesis testing is used to determine the accuracy of a hypothesis that is linked to a sample of the data. The accuracy of a hypothesis test is evaluated by means determining the probability that the data will have true statistical differences and not just random sampling errors [8]. Classical Statistics techniques approach hypothesis testing as follows:

1. A claim about the data is established. This is known as the null hypothesis.
2. A contradictory claim to the null hypothesis is established. This is known as the alternative hypothesis.
3. The null Hypothesis is rejected or not rejected and a conclusion is drawn about the data.

The hypothesis tests will be performed at a 5% level of significance such that $\alpha = 0.05$.

Different hypothesis tests will be used to shed light and attempt to answer the research questions.

4.1 Question 1: Are teachers aware of statistics as a profession?

Question 1 is evaluated by means of three graphs which provide data that illustrates teachers' perceptions of statistics professionally. The graphs are generated from data provided by teachers with respect to awareness of different statistics and mathematics qualifications, types of employers that may employ statisticians, actuaries and mathematicians and lastly, the types of activities the different qualifications may have to perform in their careers. The data is provided graphically in Figure 3.

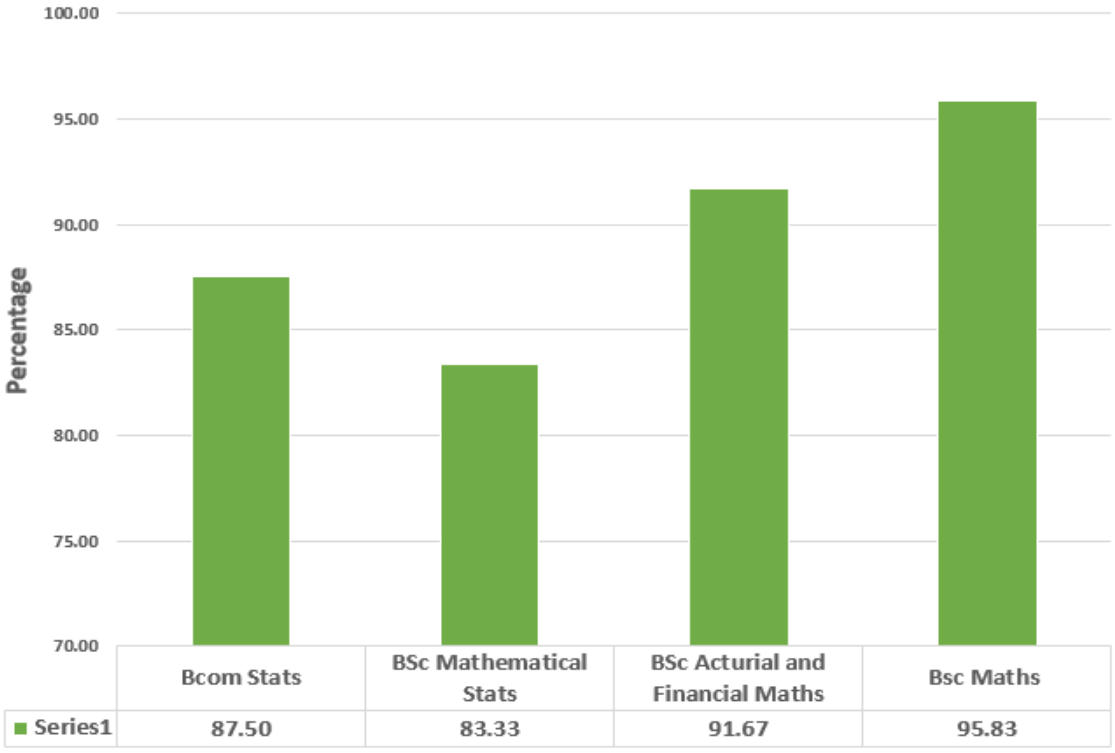


Figure 3: Teachers' awareness of BCom Stats, BSc Mathematical Stats, BSc Actuarial and Financial Mathematics and BSc Mathematics for teachers at the Awareness Event

Figure 3 summarizes responses from twenty-four of the teachers who attended the Teachers' Awareness Event. BSc in Mathematics and BSc in Actuarial and Financial Mathematics emerged as the two qualifications of which the teachers were most aware. The two Statistics qualifications, namely, BCom in Statistics and BSc in Mathematical Statistics, though at 87.5% and 83.33% awareness respectively, proved to be the qualifications that the teachers are least aware of especially in comparison to Actuarial and pure Mathematics qualifications. Table 2 illustrates what knowledge the teachers have regarding the type of employers that would typically hire statisticians, mathematicians and actuaries. The purpose is to establish whether teachers are aware of the differences and/or similarities in the careers and their roles in the professional market. Based on the displayed results, it is apparent that the teachers have a keen understanding as to

Teachers Knowledge of possible employers of statisticians, actuaries and mathematicians				
	Edgars/Foschini			
	statisticians	actuaries	mathematicians	N/A
%	91.67	25	33.33	0
	Facebook			
	statisticians	actuaries	mathematicians	N/A
%	91.67	16.67	33.33	4.17
	Sasol			
	statisticians	actuaries	mathematicians	N/A
%	83.33	54.17	62.50	0
	MTN			
	statisticians	actuaries	mathematicians	N/A
%	91.67	37.50	50	0
	OUTsurance			
	statisticians	actuaries	mathematicians	N/A
%	66.67	83.33	29.17	0
Capitec/ABSA				
	statisticians	actuaries	mathematicians	N/A
%	62.50	87.50	45.83	0

Table 2: Teachers Knowledge of possible employers of statisticians, actuaries and mathematicians

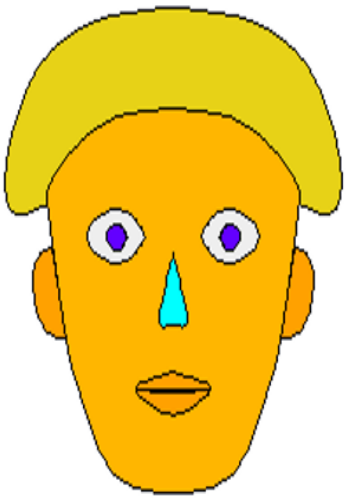
the role of statisticians professionally. The results show that there is knowledge of the type of employment that a statistician can pursue within the job market. 62.5% of the teachers at the Awareness Event revealed knowledge that banks and financial services are the biggest employers of statisticians whereas 91.67% of the teachers indicated that statisticians are employed by social media and retail industries. The information is summarized graphically in the form of Chernoff Faces in Figure 4.

There were thirty-one teachers who provided feedback at the Kutlwanong Centers. Their awareness of the different possible employers of statisticians, actuaries and mathematicians is summarized in Figure 5. Much like the teachers at the awareness event, the teachers at the Kutlwanong Centers indicated some understanding of the possible employers of statisticians with the insurance and banking industries getting the highest percentages of votes. Based on the style of hairstyles in Figure 4 anf Figure 5 , the teachers do not believe the automotive industry to be potential employers of statisticians, though this is also not true.

Question 1 was further investigated by assessing the teachers' awareness of the different activities performed by statisticians, actuaries and mathematicians. The teachers' ability to identify similarities within the qualifications was also used as a means of assessing the teachers' knowledge about statistics as a profession.

Table 3 illustrates teachers' knowledge of the different types of activities that statisticians, actuaries and mathematicians may perform in their careers. When asked about the different activities that are performed by statisticians, mathematicians and actuarial scientists in the work place, it can be deduced from Table 3, that on average, the teachers are aware of the day to day activities that the different careers offer, however, the

Height of face	Capitec or ABSA
Width of face	Edgars or Foschini
Structure of face	GEMS or Discovery Health
Height of mouth	Facebook
Width of mouth	Takealot
Smiling	Sasol
Height of eyes	Pfizer
Width of eyes	MTN
height of hair	News 24
Width of hair	OUTsurance
Style of hair	BMW



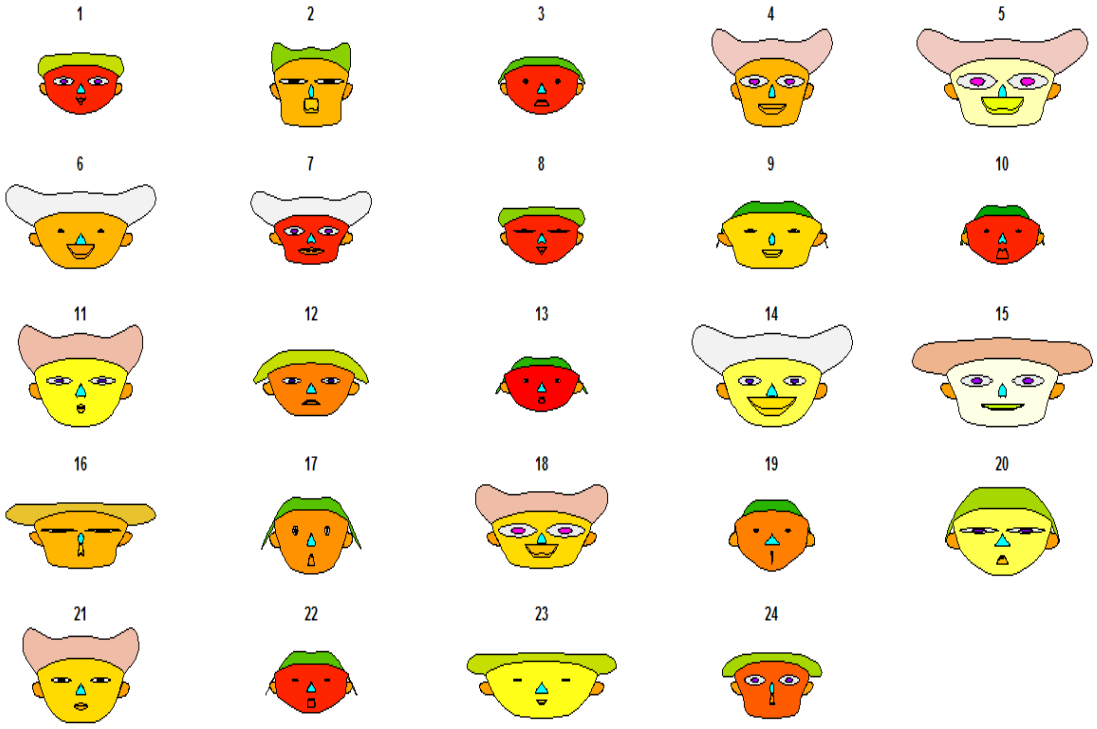
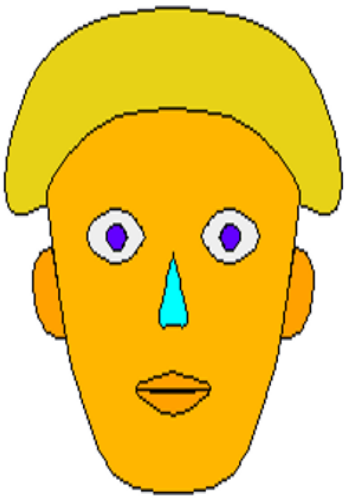


Figure 4: Chernoff Faces of possible employers of statisticians, actuaries and mathematicians as presented by teachers at the Awareness Event

Height of face	Capitec or ABSA
Width of face	Edgars or Foschini
Structure of face	GEMS or Discovery Health
Height of mouth	Facebook
Width of mouth	Takealot
Smiling	Sasol
Height of eyes	Pfizer
Width of eyes	MTN
height of hair	News 24
Width of hair	OUTsurance
Style of hair	BMW



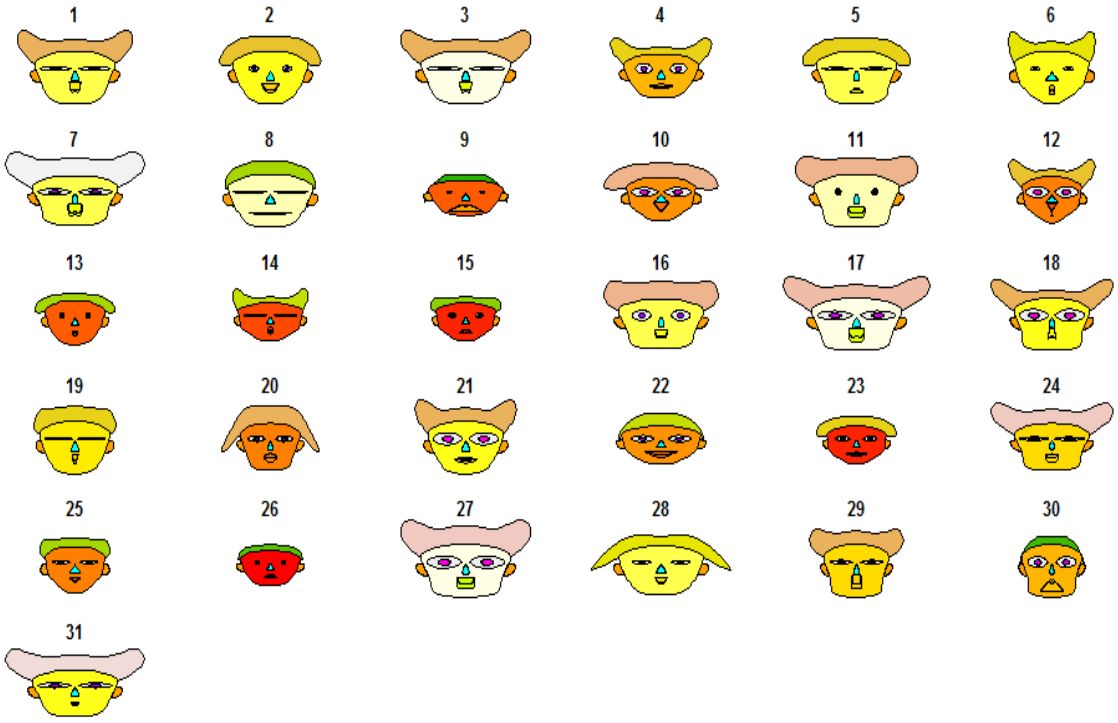


Figure 5: Chernoff Faces of possible employers of statisticians, actuaries and mathematicians as presented by teachers at the Kutlwanong Centers

Teachers' Knowledge of the professional activities of statisticians, actuaries and mathematicians				
	Identifying Sales Trends			
	statisticians	actuaries	mathematicians	N/A
%	79.17	54.17	25	4.17
	Model Rhino Poachers' Movements			
	statisticians	actuaries	mathematicians	N/A
%	83.33	4.17	33.33	4.17
	Develop Products for Insurance Companies			
	statisticians	actuaries	mathematicians	N/A
%	29.17	83.33	29.17	0
	Using Deterministic Models			
	statisticians	actuaries	mathematicians	N/A
%	75	37.5	54.17	4.17
	Working with Census Data			
	statisticians	actuaries	mathematicians	N/A
%	87.5	25	37.5	0

Table 3: Teachers' knowledge of the different activities that may be performed by statisticians, actuaries and mathematicians

responses also revealed that teachers are not aware of the differences and similarities between three different qualifications, particularly in the work place. The perceptions of the activities seem moderately split between the three different professions in question.

Figure 6 and 7 graphically illustrate the teachers' knowledge of the types of activities that are performed by statisticians, actuaries and mathematicians. Most of the teachers believed that modeling of life insurance packages is a task that is performed by statisticians. This can be seen from the very small size of the ears of most of the faces.

The teachers at the Kutlwanong Centers were also asked to rank statistics, actuarial Sciences and mathematics professions in terms of difficulty to obtain employment. The rank was done a scale of 1 - 5 with 1 = very difficult to employ and 5 = easily employed. Table 4 summarizes the percentage of teachers who ranked the highest and lowest levels of difficulty of employment for each profession. The results of which portray conflicting opinions. 29.03% of teachers believe that statisticians find employment with ease. The teachers believe that finding employment as an actuarial scientist and as a mathematician is easier than it is a statistician. More than 50% of the teachers indicated that an actuarial science qualification will be most easily employed. On the contrary however, the teachers ranked a qualification in actuarial science as the most difficult to obtain employment with nearly 20%. Similarly, statisticians, which were ranked lowest in terms of ease of securing employment, were again ranked the lowest in terms of difficulty obtaining employment. This revealed that there is a relative amount of confusion about the impact of a qualification in statistics and perhaps, an overall confusion regarding an actuarial sciences qualification and a statistics qualification.

Height of face	Working within the big data space
Width of face	Identify sales trends
Structure of face	Determine the price of an insurance product
Height of mouth	Forecast an election winner
Width of mouth	Model rhino poachers' movements
Smiling	Recommend fashion items online
Height of eyes	Detect fraudulent transactions
Width of eyes	Develop products for insurance companies
height of hair	Analyze images to predict maize crop size
Width of hair	Monitor corporate governance
Style of hair	Monitoring building quality of motor vehicles
Height of nose	Design experiments to assess the effect of drugs
Width of nose	Forecast disease outbreaks
Width of ear	Building deterministic models
Height of ear	Modeling of life insurance reserves

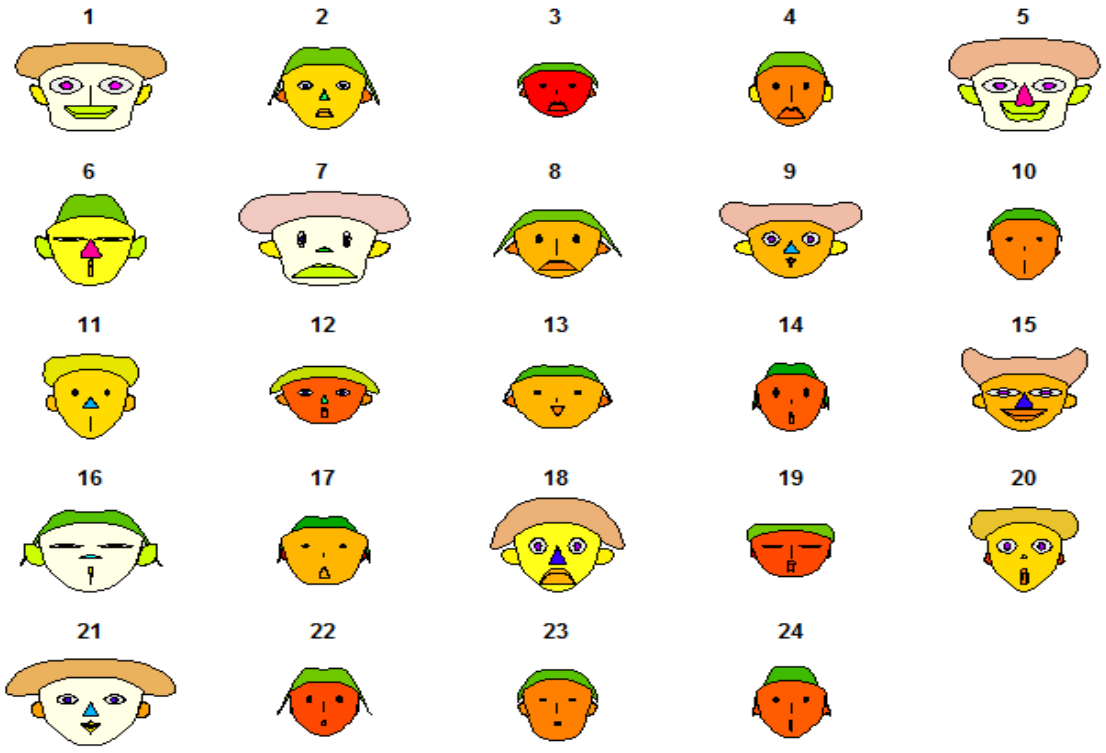


Figure 6: Chernoff Faces of Teachers' knowledge of the different activities that may be performed by statisticians, actuaries and mathematicians presented by teachers at the Statistics Awareness Event

Height of face	Working within the big data space
Width of face	Identify sales trends
Structure of face	Determine the price of an insurance product
Height of mouth	Forecast an election winner
Width of mouth	Model rhino poachers' movements
Smiling	Recommend fashion items online
Height of eyes	Detect fraudulent transactions
Width of eyes	Develop products for insurance companies
height of hair	Analyze images to predict maize crop size
Width of hair	Monitor corporate governance
Style of hair	Monitoring building quality of motor vehicles
Height of nose	Design experiments to assess the effect of drugs
Width of nose	Forecast disease outbreaks
Width of ear	Building deterministic models
Height of ear	Modeling of life insurance reserves

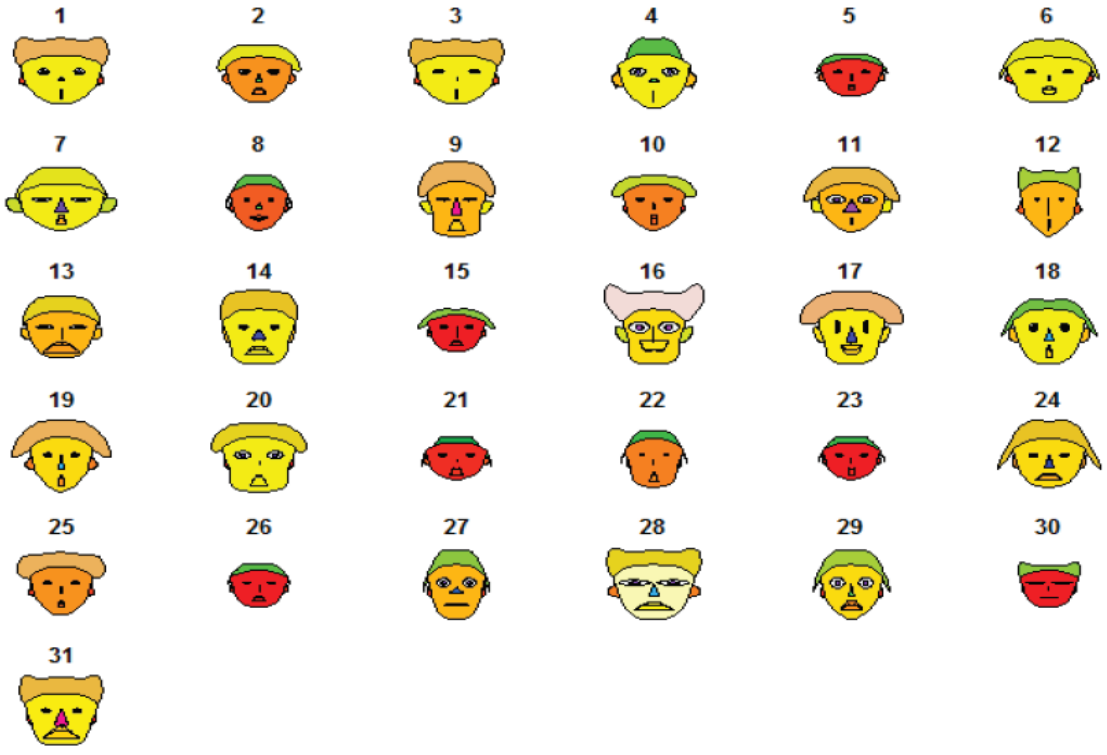


Figure 7: Chernoff Faces of Teachers' knowledge of the different activities that may be performed by statisticians, actuaries and mathematicians presented by teachers at the Kutlwanong Centers

	Actuary (%)	Statistician (%)	Mathematician (%)
5 = Easily Employed	54.84	29.03	38.71
1 = Very Difficult to get employed	19.35	6.45	9.68

Table 4: Teachers' opinions on difficulty of employment for actuaries, statisticians and actuaries

The responses also contradict current literature regarding statistics as a profession. As mentioned before, statisticians have a diverse employment opportunity profile and the with the banking industry being one of the top employers of statisticians.

There is a positive theme with regards to teachers' awareness of statistics as profession. The teachers are aware of the different statistics related qualifications but are not aware of what sets statistics apart from a career in actuarial sciences and/or pure mathematics. There is a clear mismatch in the understanding of the role and importance of statistics as a career, especially in comparison with actuarial sciences and pure mathematics.

4.2 Question 2: How do teachers perceive the inclusion of statistical content in the mathematics curriculum?

Question 2 was evaluated by a series of questions that attempt to establish teachers' attitudes towards the inclusion of statistics in the high school mathematics curriculum. This was done by means of perceptive questions with the purposes of viewing whether or not teachers understand the importance of statistics education at a high school level and the need for statistics education foundations. The data is provided graphically in Figure 8. Figure 8 illustrates the teachers' attitudes about the inclusion of statistics within the high school mathematics curriculum. All of the teachers agreed that the inclusion of statistics is important, of which, more than 66% felt strongly about the importance of the statistics content in the mathematics curriculum. Similarly, all the teachers agreed that statistics deserves to be included in the curriculum, with slightly more than 70% strongly agreeing that statistics education deserves to be included in the curriculum. Over 54% of all the teachers disagreed that the current statistics content in the curriculum is sufficient, while, roughly over 45% of the teachers felt that the statistics content is sufficient. These results illustrate that teachers are generally receptive and accepting of the inclusion of statistics and on average, would like to see more statistical content in the mathematics curriculum.

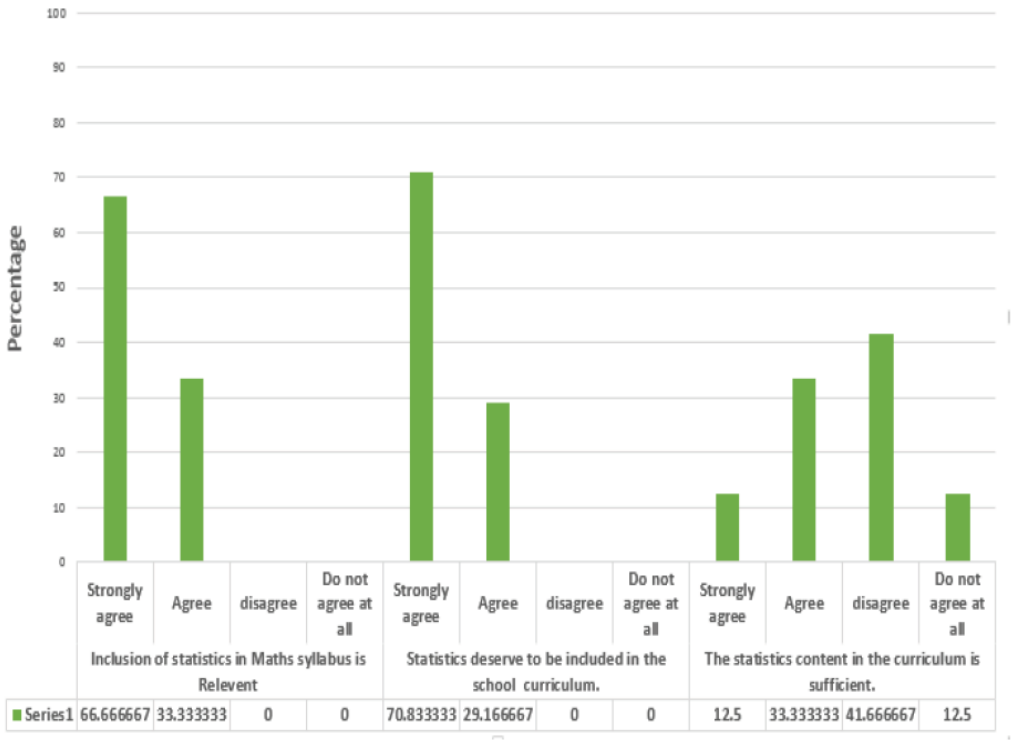


Figure 8: Teachers' perceptions on the inclusion of statistics within the high school mathematics curriculum

The inclusion of statistical concepts in mathematics syllabus is relevant	Table Probability	5.14E-04
	Pr <= P	0.4795
Statistics deserves to be included in the mathematics curriculum	Table Probability	4.98E-04
	Pr <= P	0.8441
The Statistical concepts in the curriculum are sufficient	Table Probability	3.59E-07
	Pr <= P	0.7869

Table 5: Fisher's Exact Test for teachers' perceptions on the inclusion of statistics within the high school mathematics curriculum

For the purposes of Fisher's Exact test, the teachers were grouped in terms of the number of years teaching Grade 12. This was selected as the group differentiator in the hopes of analyzing whether there is a difference in perception between teachers because of the number of years teaching.

Null Hypothesis for the inclusion of Statistical concepts in the mathematics syllabus:

H_o : The proportion of teachers' perceptions about the inclusion of statistical concepts in the Mathematics curriculum is the same for all groups

Alternative Hypothesis:

H_a : The proportion of teachers' perception per group differs significantly

p-value: $0.4795 > 0.05$ *(see Table 5)*

Thus, the null hypothesis is not rejected meaning that there is not a significant difference in proportion size of

	The inclusion of statistical concepts in mathematics syllabus is relevant			
	Years of teaching Gr12	**I agree**	**I strongly agree**	**Total**
Frequency	**Group A : (0-5)**	11	7	18
Percent		20	12.73	32.73
Row Product		61.11	38.89	
Column Product		42.31	24.14	
	Group B: (6-10)	5	4	9
		9.09	7.27	16.36
		55.56	44.44	
		19.23	13.79	
	Group C: (11-15)	1	4	5
		1.82	7.27	9.09
		20	80	
		3.85	13.79	
	Group D: (16-20)	3	9	12
		5.45	16.36	21.82
		25	75	
		11.54	31.03	
	Group E: (21-25)	2	2	4
		3.64	3.64	7.27
		50	50	
		7.69	6.9	
	Group F: (26-30)	4	3	7
		7.27	5.45	12.73
		57.14	42.86	
		15.38	10.34	
	Total	26	29	55
		47.27	**52.73**	100

Table 6: Frequency table for the teachers' perceptions on the relevance of statistical concepts in the mathematics syllabus

teachers in each age group with regards to their perceptions of the inclusion of statistics in the mathematics curriculum. To further investigate the perceptions of teachers with regards to the inclusion of statistics in the mathematics curriculum, a frequency table is analyses. Table 6 shows all the teachers, in all the different groups agree that statistics is relevant in the high school mathematics curriculum. Group A emerged with the highest percentage (32.73%) of teachers who agree to the relevance of statistics in the curriculum. In total, more than 50% of the teachers strongly feel that statistics is a relevant concept within high school mathematics.

Null Hypothesis to assess teachers' perceptions regarding whether or not statistics should be included in the mathematics curriculum:

H_o : The proportion of teachers' perceptions regarding whether statistics deserves to be included in the mathematics curriculum is the same for all groups

Alternative Hypothesis:

	Statistics deserves to be included in the school curriculum				
	Years of teaching Gr12	**I disagree**	**I agree**	**I strongly agree**	**Total**
Frequency	**Group A : (0-5)**	1	4	12	17
Percent		1.96	7.84	23.53	33.33
Row Product		5.88	23.53	70.59	
Column Product		100	25	35.29	
	Group B: (6-10)	0	3	4	7
		0	5.88	7.84	13.73
		0	42.86	57.14	
		0	18.75	11.76	
	Group C: (11-15)	0	1	4	5
		0	1.96	7.84	9.8
		0	20	80	
		0	6.25	11.76	
	Group D: (16-20)	0	3	8	11
		0	5.88	15.69	21.57
		0	27.27	72.73	
		0	18.75	23.53	
	Group E: (21-25)	0	1	3	4
		0	1.96	5.88	7.84
		0	25	75	
		0	6.25	8.82	
	Group F: (26-30)	0	4	3	7
		0	7.84	5.88	13.73
		0	57.14	42.86	
		0	25	8.82	
	Total	1	16	34	51
		1.96	**31.37**	**66.67**	**100**

Table 7: Frequency table for teachers' perceptions of whether or not statistics deserves to be included in the school curriculum

H_a : The proportion of teachers' perception per group differs significantly

p-value: $0.8442 > 0.05$ *(see Table 5)*

Thus, the null hypothesis is not rejected. The different groups in terms of years of teaching grade 12 have similar believes about whether statistics deserves to be included in the mathematics curriculum. To further investigate the perceptions of the teachers, the responses per group of years of teaching are analyzed in a frequency table. Table 7 illustrates that less than 2% of all the teachers indicated that they believe that statistics should not be included on the high school mathematics curriculum.

Null Hypothesis to assess teachers' perceptions regarding their beliefs about the statistical concepts in the high school mathematics curriculum is as follows:

H_o : The proportion of teachers' perceptions regarding statistical concepts in the mathematics curriculum is the same for all groups

Alternative Hypothesis:

H_a : The proportion of teachers' perception per group differs significantly

	The statistical concepts in the curriculum are sufficient					
	Years teaching Gr12	**I strongly disagree**	**I disagree**	**I agree**	**I strongly agree**	**Total**
Frequency	**Group A : (0-5)**	3	5	8	1	**17**
Percent		5.88	9.8	15.69	1.96	**33.3**
Row Product		17.65	29.41	47.06	5.88	
Column Product		42.86	26.32	36.36	33.3	
	Group B: (6-10)	2	4	1	0	**7**
		3.92	7.84	1.96	0	**13.7**
		28.57	57.14	14.29	0	
		28.57	21.05	4.55	0	
	Group C: (11-15)	1	1	2	1	**5**
		1.96	1.96	3.92	1.96	**9.8**
		20	20	40	20	
		14.29	5.26	9.09	33.33	
	Group D: (16-20)	1	5	4	1	**11**
		1.96	9.8	7.84	1.96	**21.6**
		9.09	45.45	36.36	9.09	
		14.29	26.32	18.18	33.33	
	Group E: (21-25)	0	2	2	0	**4**
		0	3.92	3.92	0	**7.84**
		0	50	50	0	
		0	10.53	9.09	0	
	Group F: (26-30)	0	2	5	0	**7**
		0	3.92	9.8	0	**13.7**
		0	28.57	71.43	0	
		0	10.53	22.73	0	
	Total	7	19	22	3	51
		13.73	**37.25**	**43.14**	**5.88**	**100**

Table 8: Frequency table of teachers' perceptions of whether or not the statistical concepts in the curriculum are sufficient

p-value: $0.7869 > 0.05$ *(see Table 5)*

Thus, the null hypothesis is not rejected. The different groups in terms of years of teaching grade 12 have similar believes about the statistical concepts in the mathematics curriculum. To further investigate the perceptions of the teachers, the responses per group of years of teaching are analyses in a frequency table. Table 8 illustrates that overall, there is an even split with 50.98% objecting to the fact that the statistics curriculum is sufficient. Even within the different groups of years if teaching, the split between disagreeing and agreeing with the statistics content is even with the exception of group B, which are teachers who have been teaching for 6-10 years. Within group B six out of the seven teachers do not agree with that the statistical content is sufficient. The rest of the groups A, C, D and E, seem to be satisfied with the statistical content in the mathematics curriculum.

The teachers at the Kutlwanong Centers were asked to rank different topics in mathematics curriculum according to importance *(see Figure 9)*.

The teachers are positively receptive and accepting of the inclusion of statistics however there are split opinions on the importance of statistics relative to other concepts within the high school mathematics curriculum.

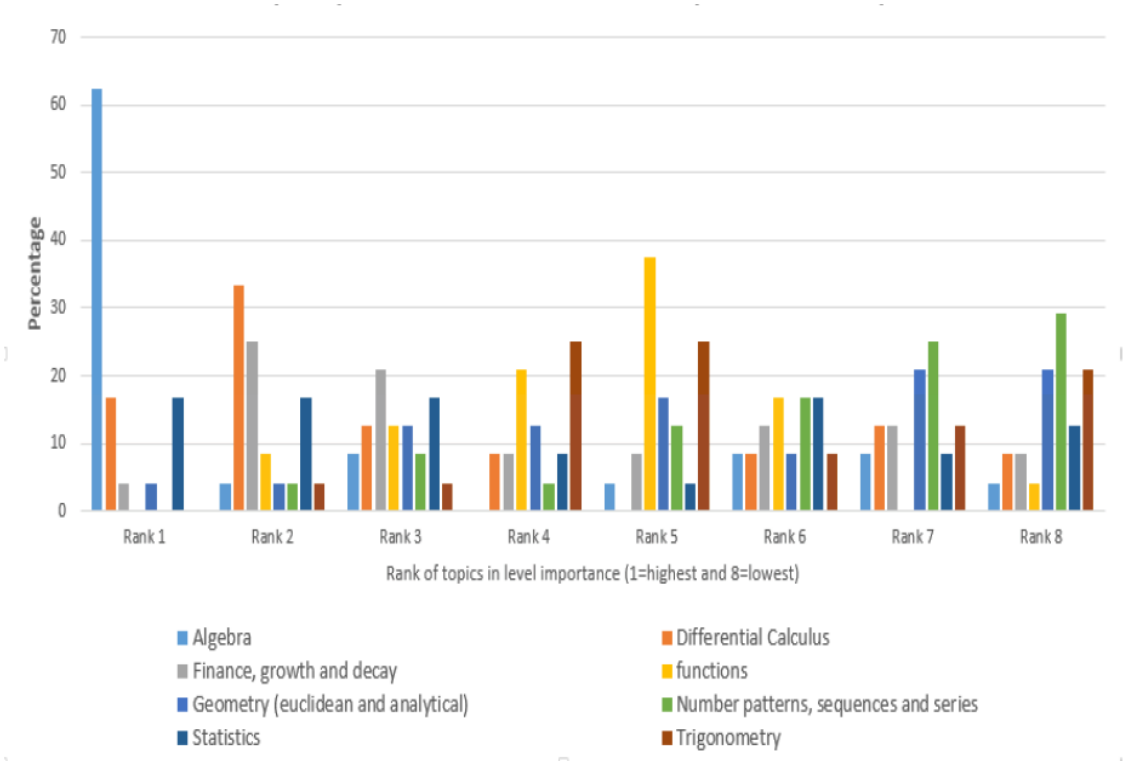


Figure 9: Teachers' perceptions of which topics are most important in the mathematics syllabus

Figure 9 illustrates teachers' beliefs about which topics in the mathematics curriculum are most important. The teachers were asked to rank eight topics, including statistics, on a scale of 1 to 8, with 1 being the most important and 8 being the least important. The results revealed that more than 60% of the teachers believe algebra to be the most important topic within the curriculum. Just under 50% of the teachers ranked statistical concepts within the top three positions. Under 40% of the teacher ranked statistics to be within the bottom three. Statistics also appeared in the lowest rank at just over 12%. There is evidence that teachers perceive the inclusion of statistics in the mathematics curriculum to be a positive change but there are conflicting view on the importance of statistics relative to other concepts within the mathematics curriculum

4.3 Question 3: Does the number of years of teaching mathematics affect the perception of statistics?

The Fisher's exact tests performed in section 4.2 where performed using the years of teaching grade 12 as a differentiator for the different groups of teachers. From Table 5 in section 4.2, it is evident that teachers of different groups have similar perceptions about the inclusion of statistics in the high school mathematics curriculum.

Further investigations were done using Fisher's Exact to analyze whether teachers of the different groups had different opinions about the definition of statistics and the difficulty of statistics as a topic of teaching. Similarly, to section 4.2, the teachers are split into groups of years of teaching grade 12.

Null Hypothesis to assess teachers' opinions on the definition of statistics:

H_o : The proportion of teachers' perceptions regarding definition of statistics is the same for all groups

Alternative Hypothesis:

H_a : The proportion of teachers' perceptions per group differs significantly

p-value: $1 > 0.05$ *(see Table 9)*

Thus, the null hypothesis is not rejected. The perceptions of the teachers do not differ significantly is the same for all groups.

Null Hypothesis to assess teachers' perceptions regarding the level of difficulty of statistics as a topic of teaching:

H_o : The proportion of teachers' perceptions regarding the difficulty of statistics as a topic of teaching is the same for all groups

Alternative Hypothesis:

H_a : The proportion of teachers' perception per group differs significantly

p-value: $0.746 > 0.05$ *(see Table 9)*

Thus, the null hypothesis is not rejected. The perception of the teachers is not significantly different for all groups.

Both tests further support the fact that there are no significant differences in the perceptions of teachers due to years of teaching. The years of teaching do not significantly affect the teachers' perceptions of statistics.

	Fisher's Exact Test	
Which one of the phrases best describes your understanding of what statistics is	Table probability	0.1243
	Pr <= P	1
Rate teaching of statistics as a topic in terms of difficulty	Table Probability	3.88E-04
	Pr <= P	0.746

Table 9: Fisher's Exact Test for Teachers' perceptions the definition of statistics and level of difficulty of statistics as a teaching topic

5 Conclusion

The purpose of this research was to evaluate the perceptions and awareness of Statistics as a discipline amongst high school mathematics teachers. This was done by means of a questionnaire that was given to mathematics teachers that attended an awareness event at that was hosted by the University of Pretoria, and mathematics teachers at the Kutlwanong Centers in Gauteng. The method of Chernoff faces, graphs and tables were used for a descriptive analysis of the data. Fisher's Exact Test was used to establish relationships between the different groups of teachers.

The research revealed that teachers are aware of statistics qualifications. However, they are more aware of the importance of pure mathematics and actuarial sciences qualifications. The teachers also showed a keen understanding of statistics as a profession but struggled to identify the daily activities of statisticians and the career opportunities available to a professional statistician. The teachers further struggled to identify the main differences and similarities between statisticians, mathematicians and actuarial Scientist. Overall, the teachers showed similar perceptions about statistics within the high school mathematics curriculum though there is a clear conflict in terms of the importance of statistics relative to the other concepts in the mathematics curriculum.

6 Recommendations

The study faced various challenges, the main challenge being the limited data size of only fifty-five teachers. This restricted the types of analysis methods that could be applied to the data. Furthermore, the teachers were not as active in answering the online follow-up questionnaires which were meant to provide insight about the impact of the Statistics Awareness Event that was hosted at the University of Pretoria. Further studies of this nature need a more extensive sample sizes so as to obtain more accurate data. Comparisons regarding the effects of the quintiles of the schools and the number of learners per teacher per class could

provide further insight on teachers' perceptions and awareness of statistics in the mathematics curriculum and perhaps bring to light the challenges that high school teachers of schools of varying backgrounds face in the teaching of statistics.

References

[1] Bongani D Bantwini. How teachers perceive the new curriculum reform: Lessons from a school district in the Eastern Cape, South Africa. *International Journal of Education Development*, 30(1):83–90, 2010.

[2] Tony Bush and Jan Heystek. School governance in the new south africa. *Compare: A Journal of Comparative and International Education*, 33(2):127–138, 2003.

[3] Martin Carnoy and Linda Chisholm. Towards understanding student academic performance in south africa: A pilot study of grade 6 mathematics lessons in gauteng province. Technical report, The Human Sciences Research Council (HSRC) with Stanford Universty, 2008.

[4] Herman Chernoff. *Chernoff Faces*, pages 243–244. Springer, Berlin, Heidelberg, 2011.

[5] Herman Chernoff and E. L. Lehmann. The use of maximum likelihood estimates in χ^2 tests for goodness of fit. *The Annals of Mathematical Statistics*, 25(3):579–586, September 1954.

[6] William G. Cochran. Some methods for strengthening the common $\chi^2 tests. Biometrics, 10(4) : 417 - -451, 1954.$

[7] National Planning Commission. Our future - make it work. In *National Development Plan 2030*, 2011.

[8] Donald R. Cooper and Pamela S. Schindler. *Business Research Methods.* McGraw Hill Irwin, 2011.

[9] John W. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches.* SAGE Publications, Inc, 2013.

[10] Ram C. Dahiya and John Gurland. Pearson chi-squared test of fit with random intervals. *Biometrika*, 59(1):147–153, April 1972.

[11] Iddo Gal. Adults' statistical literacy: Meanings, components, responsibilities. *International Statistical Review*, 70(1):1–51, 2002.

[12] Iddo Gal, Lynda Ginsburg, and Candace Schau. *The Assessment Challenge in Statistics Education.* IOS PR, 1997.

[13] Joan B. Garfield. Assessing statistical reasoning. *Statistics Education Research Journal*, 2(1):22–38, May 2003.

[14] J. Stuart Hunter. Statistics as a profession. *Journal of the American Statistical Association*, 89(425):1–6, 1994.

[15] Miljenko Huzak. *Chi-Square Distribution*, pages 245–246. Springer, Berlin, Heidelberg, 2011.

[16] Paul l. Ayres and John M. McCormick. Grade 12 mathematics teachers' views on curriculum reform in new south wales. In *29th annual conference of the Mathematics Education Research Group of Australasia*, volume 1, pages 54–59. Mathematics Education Research Group of Australasia, 2006.

[17] Anne Lacey and Donna Luff. Qualitative research analysis. *The NIHR RDS for the East Midlands / Yorkshire & the Humber*, 2007.

[18] Melita Puklek Levpuscek and Maja Zupancic. Math achievement in early adolescence: The role of parental involvement, teachers' behavior, and students' motivational beliefs about math. *The Journal of Early Adolescence*, 29(4):541–570, August 2009.

[19] John Ludbrook. Analysing 2 by 2 contingency tables: Which test is best? *Clinical and Experimental Pharmacology and Physiology*, 40(3):177–180, 2013.

[20] Vukani Cleopas Maphumulo. A study of teachers' attitudes towards selected challenges in the teaching mathematics in Maphumulo circuit. Master's thesis, University of Zululand, 2015.

[21] R. F Mohammed and B Harlech-Jones. The fault in in ourselves: looking at failures in implentation. *Compare: a journal of comparative and international education*, 38(1):39–51, 2008.

[22] Luciano Molinari. Distribution of the chi-squared test in nonstandard situations. *Biometrika*, 64(1):115–121, April 1977.

[23] Anne Prescott and Michael Cavanagh. An investigation of pre-service secondary mathematics teachersâ beliefs as they begin their teacher training. *Mathematics Education Research Group of Australasia*, 2:424–431, 2006.

[24] Mamokgethi Setati, Jill Adler, Yvonne Reed, and Abdool Bapoo. Incomplete journeys: Code-switching and other language practices in Mathematics, Science and English Language Classrooms in South Africa. *Language and Education*, 16(2):128–149, jun 2002.

[25] O. B. Sheynin. Studies in the history of probability and statistics. xxv. on the history of some statistical laws of distribution. *Biometrika*, 58(1):234–236, 1971.

[26] Nicholas Spaull. South Africa's education crisis: The quality of education in South Africa 1994-2011. *Center of Development and Enterprise*, October 2013.

[27] Jane M. Watson. Assessing statistical thinking using the media. In Iddo Gal and Joan B. Garfield, editors, *The Assessment Challenges in Statistics Education*, chapter 9, pages 107–121. International Statistical Institute, 1997.

[28] Jane M Watson. Profiling teachers' competence and confidence to teach particular mathematics topics: The case of chance data. *Journal of Mathematics Teacher Education*, 4:305–337, 2001.

[29] Helena Wessels. Statistics in the South African School Curriculum. In *Teaching Statistics in School Mathematics-Challenges for Teaching and Teacher Education*, pages 21–25. Springer Netherlands, 2011.

[30] Robert K. Yin. *Case Study Research: Design and Methods (Applied Social Research Methods)*. SAGE Publications, Inc, 2002.

7 Appendix

SAS CODE:

```
data Resp;
input Group Age $ YoT10 $ YoT11 $ YoT12 $ relevant should sufficient BcomS
BscMS BscAFM BscM Q14S Q14A Q14M Q161 Q162 Q163 ;
datalines;
1 g f f f 4 4 2 1 1 1 1 1 1 1 2 2 2
1 e e e d 3 3 2 0 0 0 1 0 1 0 1 2 2
1 d d d d 4 4 2 0 1 1 1 1 1 1 3 3 3
1 f f f f 3 3 3 1 1 1 1 1 1 1 2 2 2
1 f d d d 3 3 3 1 1 1 1 1 1 1 2 2 2
1 f f f e 4 4 2 1 1 1 1 1 1 1 2 2 2
1 f d f f 4 4 3 1 0 1 1 0 1 1 1 1 1
1 f e e e 4 4 2 1 1 1 1 1 1 1 2 2 2
1 e b d d 4 4 1 1 1 1 1 1 1 1 2 2 3
1 f d d d 4 4 2 1 1 1 1 1 1 1 2 2 2
1 e d d d 4 4 4 1 1 1 1 1 1 1 2 3 3
1 f a a a 4 4 1 0 0 1 1 0 1 1 2 3 3
1 f e e e 3 4 3 1 1 1 1 1 1 1 2 2 2
1 g c b c 3 3 3 1 1 1 1 0 1 1 1 2 2
1 f e f f 4 4 3 1 1 1 1 1 1 1 2 2 3
1 e c c b 4 4 3 1 1 1 1 1 1 1 2 2 2
1 f a c c 4 4 1 1 1 1 1 1 1 1 1 2 3
1 f f f f 3 3 3 1 1 1 1 0 1 1 2 2 2
1 a b a a 3 3 3 1 1 1 1 0 1 1 3 3 3
1 f b e f 3 3 2 1 1 2 1 0 1 1 2 3 3
1 d a a a 4 4 4 1 1 1 1 1 1 1 2 3 3
1 d d d d 4 4 2 1 1 1 1 1 1 1 3 3 3
1 d b c a 4 4 2 0 0 1 1 0 1 1 3 3 3
1 b b b b 3 3 2 1 1 1 1 1 1 1 2 2 2
2 e e d d 3 3 3 0 1 1 1 1 1 1 2 2 2
2 c b b b 3 4 2 0 1 1 1 0 1 1 3 3 3
2 f a c a 4 4 2 0 0 1 1 1 1 1 2 2 2
2 f e e d 4 4 3 0 0 1 1 1 1 1 3 3 3
2 b a a a 4 4 2 1 1 1 1 1 1 1 2 2 2
2 d b b b 4 4 2 1 1 1 1 1 1 1 3 3 3
2 d c d d 4 4 2 0 1 1 1 1 1 1 2 2 2
2 d d d d 4 4 3 0 1 1 1 1 0 1 2 2 2
2 e e e e 3 3 3 1 1 1 1 1 1 1 3 3 3
2 f b b b 3 3 1 1 1 1 1 1 1 1 2 3 3
2 a a a a 3 4 1 1 1 1 1 1 1 1 2 2 2
2 c c b c 4 4 4 1 1 1 1 1 0 1 3 3 3
2 c c c c 4 4 2 1 1 1 1 1 1 0 3 3 3
2 b b a a 4 4 2 1 1 1 1 1 1 1 2 3 3
2 d 0 b c 4 4 3 0 1 1 1 0 1 0 2 2 2
2 a 0 0 a 3 3 3 0 1 1 1 0 0 1 3 3 3
2 f a a b 3 3 0 1 1 1 1 1 1 1 3 3 3
2 b 0 0 a 3 3 3 1 1 1 1 0 0 1 2 2 2
2 a a 0 a 3 4 3 1 1 1 1 1 1 1 3 2 3
2 f b b a 4 4 0 1 1 1 1 1 1 1 3 2 2
2 d b b b 4 4 3 1 1 1 1 1 1 1 2 3 3
2 a b b a 3 2 3 1 1 1 1 2 2 2 3 3 3
2 c b b b 3 4 1 1 1 1 1 1 1 1 3 3 3
2 a a a a 3 4 2 1 1 1 1 0 1 1 2 3 2
2 b b a a 3 4 3 1 0 1 1 0 0 1 3 3 3
2 f e f f 3 3 3 0 1 0 1 1 1 0 2 1 3
2 d d d b 4 4 2 1 1 1 1 1 1 1 2 2 2
2 b a b a 3 3 3 1 1 0 1 0 0 1 3 3 3
2 a a a a 3 3 3 1 1 1 1 1 1 1 1 2 2
2 a a a a 3 4 3 2 1 0 1 1 1 1 2 2 3
2 e 0 d d 4 4 3 1 1 1 1 1 1 1 2 2 2
```

```
;
proc print data = resp; run;
ods html body = 'D:\Analysisfinal.xls';
proc freq data = resp;
tables group*age / fisher;
run;
proc freq data = resp;
tables YoT12*relevant/fisher;
run;
proc freq data = resp;
tables YoT12*should/fisher;
run;
proc freq data = resp;
tables YoT12*sufficient/fisher;
run;
proc freq data = resp;
tables YoT12*BcomS/fisher;
run;
proc freq data = resp;
tables YoT12*BscMS/Fisher;
run;
proc freq data = resp;
tables YoT12*BscAFM/Fisher;
run;

proc freq data = resp;
tables Yot12*BscM/Fisher;
run;

proc freq data = resp;
tables YoT12*Q14S/Fisher;
run;
proc freq data = resp;
Tables YoT12*Q14A/Fisher;
run;
proc freq data = resp;
Tables YoT12*Q14M /Fisher;
run;
proc freq data = resp;
tables Yot12*Q161/fisher;
run;
proc freq data = resp;
tables YoT12*Q162/fisher;
run;
proc freq data = resp;
tables YoT12*Q163/fisher;
run;
ods html close;
```

R CODE:

#Luwela Nodada 14433852

#23 September 2017

#Group 2

#Code for Faces for Daily Work Routines of statisticians, actuaries and mathematicians
install.packages("aplpack")

Group2 <- read.csv("D://Book1.csv")

```
library(aplpack)

faces(Group2[,2:19])

###############################################################################################

#Luwela Nodada 14433852

#23 September 2017

#Group 1

#Code for Faces for Daily Work Routines of statisticians, actuaries and mathematicians
install.packages("aplpack")

Group <- read.csv("C://Users/Luwela' maza/Desktop/Research/ResearchPaper/Data
analysis/ChernoffAFaces/facesG1.csv")

Group[1:55,]

library(aplpack)

faces(Group[,2:18])
```

Department of Statistics
Awareness project

The aim of the questionnaire is to:

- establish awareness about statistics as a profession; and
- explore perceptions about the role of statistics in real life.

Take Note:
This questionnaire is opinion based and therefore there are no incorrect answers. At each question, indicate the most appropriate answer(s) or fill in where applicable.

0	Respondent number

BIOGRAPHICAL INFORMATION

1	**<30**	**31-34**	**35-40**	**40-44**	**45-50**	**51-54**	**55-60**	**61-65**	**>65**
Your age in years is	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

2	**NQ1**	**NQ2**	**NQ3**	**NQ4**	**NQ5**
The quintile of your school is	**1**	**2**	**3**	**4**	**5**

3

Complete the table to reflect your qualifications	**Year obtained**	**Major subjects**
Diploma		
B-degree		
Postgraduate diploma		
Hons		
Masters		

4	**Gr 10**	**Gr 11**	**Gr 12**
How many years have you been teaching **MATHEMATICS** in the different grades			
How many years have you been teaching **MATHEMATICS LITERACY** in the different grades			
How many learners do you have currently in each grade in **MATHEMATICS**			
How many learners do you have currently in each grade in **MATHEMATICS LIETERACY**			

PERCEPTIONS ABOUT STATISTICS

5 Indicate your level of agreement with the following statements.	I do agree at all	I Disagree	I Agree	I Strongly agree
a) The inclusion of Statistics in the Mathematics syllabus is relevant.	**1**	**2**	**3**	**4**
b) Statistics deserve to be included in the school curriculum.	**1**	**2**	**3**	**4**
c) The statistics content in the curriculum is sufficient.	**1**	**2**	**3**	**4**

6

Are you aware of the following degree qualifications? Mark all.	**Yes**	**No**
BCom Statistics	**1**	**2**
BSc Mathematical Statistics	**1**	**2**
BSc Actuarial and Financial Mathematics	**1**	**2**
BSc Mathematics	**1**	**2**

7 Which of the following activities might form part of the typical daily work routine of the different professionals. Indicate your opinion by marking the appropriate block(s) with an X. **You may choose more than one professional per activity**.

Activities	**Statisticians**	**Actuaries**	**Mathematicians**
Working within the big data space	**1**	**2**	**3**
Identify sales trends	**1**	**2**	**3**
Determine the price of an insurance product	**1**	**2**	**3**
Forecast an election winner	**1**	**2**	**3**
Model rhino poachers' movements	**1**	**2**	**3**
Recommend fashion items online	**1**	**2**	**3**
Detect fraudulent transactions	**1**	**2**	**3**
Develop products for insurance companies	**1**	**2**	**3**
Analyze images to predict maze crop size	**1**	**2**	**3**
Monitor corporate governance	**1**	**2**	**3**
Monitoring building quality of motor vehicles	**1**	**2**	**3**
Design experiments to assess the effect of drugs	**1**	**2**	**3**

Activities	Statisticians	Actuaries	Mathematicians
Forecast disease outbreaks	1	2	3
Deterministic modeling	1	2	3
Modeling of life insurance reserves	1	2	3
Making financial projections	1	2	3
Work with census data	1	2	3
Mathematical models for Malaria	1	2	3

8 Which of the following employers might typical employ the different professionals. Indicate your opinion by marking the appropriate block(s) with an X. **You may choose more than one professional per employer**.

Employers	Statisticians	Actuaries	Mathematicians
Capitec or ABSA	1	2	3
Edgars or Foschini	1	2	3
GEMS or Discovery Health	1	2	3
Facebook	1	2	3
Takealot	1	2	3
Sasol	1	2	3
Pfizer	1	2	3
MTN	1	2	3
News 24	1	2	3
OUTsurance	1	2	3
BMW	1	2	3

9

Share your experience for each question	Yes	No
Have you ever advised a learner to follow **statistics** as a career?	1	2
Have you ever advised a learner to follow **actuarial science** as a career?	1	2
Have you ever advised a learner to follow **mathematics** as a career?	1	2

10The table shows a list of topics in the Mathematics curriculum. Rank these topics (according to your perception) into the order of importance. Write a **1** next to the most important topic, a **2** next to the second most important and continue in this fashion. **You may only write one no 1 and one no 2up to 8.**

Mathematics topics	Rank (1-8)
Functions	
Number patterns, Sequences and Series	
Finance, growth and decay	
Algebra	
Differential calculus	
Geometry (Euclidian and Analytical)	
Trigonometry	
Statistics and Probability	

11 What do you think the mark allocation for statistics in the curriculum should ideally be?

Grades	**Option 1**	**X**	**Option 2**	**X**	**Option 3**	**X**
Grade 10	Less than 10 marks		15 ± 5		More than 20	
Grade 11	Less than 15		20 ± 5		More than 25	
Grade 12	Less than 15		20 ± 5		More than 25	

LEARNER PERCEPTION

12 What do you believe is your learners' perception about statistics?	I do agree at all	I Disagree	I Agree	I Strongly agree
They understand it	**1**	**2**	**3**	**4**
They are afraid of it	**1**	**2**	**3**	**4**
They see the relevance	**1**	**2**	**3**	**4**
They are not excited about it	**1**	**2**	**3**	**4**
They are enthusiastic to learn about it	**1**	**2**	**3**	**4**

Thank you ☺

Department of Statistics
Statistics awareness project

The aim of the questionnaire is to:

- establish awareness about statistics as a profession; and
- explore perceptions about the role of statistics in real life.

Take Note:
This questionnaire is opinion based and therefore there are no incorrect answers. At each question, indicate the most appropriate answer(s) by making an X in the appropriate shaded block(s) or fill in an answer where applicable in the shaded area.

0	Respondent number

BIOGRAPHICAL INFORMATION

1	**<31**	**31-35**	**36-40**	**41-45**	**46-50**	**51-55**	**56-60**	**61-65**	**>65**
Your age in years is									

2	**NQ1**	**NQ2**	**NQ3**	**NQ4**	**NQ5**
The quintile of your school is					

3

Complete the table to reflect your qualifications	**Year obtained**	**Major subjects**
Diploma		
B-degree		
Postgraduate diploma		
Hons		
Masters		

4	**Gr 10**	**Gr 11**	**Gr 12**
How many years have **YOU** been teaching **MATHEMATICS** in the different grades			
How many years have **YOU** been teaching **MATHEMATICS LITERACY** in the different grades			
How many learners do **YOU** have currently in each grade in **MATHEMATICS**			
How many learners do **YOU** have currently in each grade in **MATHEMATICS LITERACY**			

PERCEPTIONS ABOUT STATISTICS

5 When were you initially introduced to statistics?

As a subject of learning	
As a subject of teaching	

6 In your opinion, which **ONE** of these phrases best describes your understanding of what statistics is?

A science of data collection and capturing	
Methods and techniques used to collect, summarize and interpret data	
Tabular and graphical illustrations used to summarise data	

7 Rate the **teaching** of statistics as a topic in terms of difficulty

Easy	
Moderate	
Difficult	

8 Indicate your level of agreement with the following statements.

	I do not agree at all	I Disagree	I Agree	I Strongly agree
a) The inclusion of Statistical concepts in the Mathematics syllabus is relevant.				
b) Statistics deserves to be included in the school curriculum.				
c) The statistical concepts in the curriculum is sufficient.				

9 Are you aware of the following degree qualifications? Mark all.

	Yes	**No**
BCom Statistics		
BSc Mathematical Statistics		
BSc Actuarial and Financial Mathematics		
BSc Mathematics		

10 Do you understand the difference between a pure statistical degree, a pure mathematics degree and an actuarial degree?

Yes	
No	
I am Uncertain	

11 Based on your opinion, give a rating of the following careers in terms of difficulty of obtaining employment on a scale of 1 to 5
***(1 = Very difficult to get employed, 5 = Easily employed*)**:

Actuary	
Statistician	
Mathematician	

12 Which of the following activities might form part of the typical daily work routine of the different professionals. Indicate your opinion by marking the appropriate block(s) with an X. **You may choose more than one professional per activity**.

Activities	**Statisticians**	**Actuaries**	**Mathematicians**
Working within the big data space			
Identify sales trends			
Determine the price of an insurance product			
Forecast an election winner			
Model rhino poachers' movements			
Recommend fashion items online			
Detect fraudulent transactions			
Develop products for insurance companies			
Analyze images to predict maize crop size			
Monitor corporate governance			
Monitoring building quality of motor vehicles			
Design experiments to assess the effect of drugs			
Forecast disease outbreaks			
Building deterministic models			
Modeling of life insurance reserves			
Making financial projections			
Work with census data			
Models for malaria predictions			

13 Which of the following employers might typical employ the different professionals. Indicate your opinion by marking the appropriate block(s) with an X. **You may choose more than one professional per employer**.

Employers	**Statisticians**	**Actuaries**	**Mathematicians**
Capitec or ABSA			
Edgars or Foschini			
GEMS or Discovery Health			
Facebook			
Takealot			
Sasol			
Pfizer			
MTN			
News 24			
OUTsurance			
BMW			

14 Share your experience for each question

	Yes	**No**
Have you ever advised a learner to follow **statistics** as a career?		
Have you ever advised a learner to follow **actuarial science** as a career?		
Have you ever advised a learner to follow **mathematics** as a career?		

15 The table shows a list of topics in the Mathematics curriculum. Rank these topics (according to your perception) into the order of importance. Write a **1** next to the most important topic, a **2** next to the second most important and continue in this fashion. The ranks must be unique (used only once) and thus no duplicates are allowed.

	Rank (1-4)
Geometry	
Algebra	
Statistics	
Finance, growth and Decay	

16 What do you think the mark allocation for statistics in the curriculum should ideally be?

Grades	**Option 1**	**X**	**Option 2**	**X**	**Option 3**	**X**
Grade 10	Less than 10 marks		15 ± 5		More than 20	
Grade 11	Less than 15		20 ± 5		More than 25	
Grade 12	Less than 15		20 ± 5		More than 25	

LEARNER PERCEPTION

17 What do you believe is your learners' perception about statistics?

	I do not agree at all	I Disagree	I Agree	I Strongly agree
They understand it				
They find it challenging				
They see the relevance				
They are not excited about it				
They are enthusiastic to learn about it				

Thank you ☺

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

FACULTY OF NATURAL AND AGRICULTURAL SCIENCES

DEPARTMENT OF STATISTICS

INFORMATION LEAFLET AND INFORMED CONSENT

THE PERCEPTIONS AND AWARENESS OF THE STATISTICS PROFESSION AND ROLE OF STATISTICS AMONGST MATHEMATICS TEACHERS

Primary investigator: Ms L Nodada, (BComm)
Study leader: Mr A Swanepoel, MSc, Department of Statistics, University of Pretoria
Co-study leader: Dr CJ Louw, PhD, UP, Ms S Makgai, (MSc), UP

Dear Potential research participant,

You are invited to participate in a research study that forms part of my formal Honours studies. This information leaflet will inform you about all aspects of the study.

The study aims at determining your perceptions about Statistics as part of the Mathematics curriculum and Statistics as a career. You will be required to complete a questionnaire at today's event. We will then email you a second questionnaire to obtain more information. The second questionnaire will be done online.

You will be required to:

- sign this informed consent form;
- complete the questionnaire handed to you at this event; and
- respond to a questionnaire that will be emailed to you. It should not take more than 20 minutes to complete the questionnaires.

Important information

- You **will not** be paid to participate in the study. However, you will receive a number that entitle you entrance in a lucky draw.
- Your participation in this study is entirely voluntary. You have the right to withdraw at any stage without any penalty or future disadvantage whatsoever.
- Your identity will not be known to us and all data will be reported without any link to you or your school.
- Data will be handled confidentially.
- The research supervisors are adequately qualified to oversee this research project. Additional information can be obtained from the primary supervisor, Mr A Swanepoel at andre.swanepoel@up.ac.za.

Your co-operation and participation in the study is highly appreciated. Please sign the informed consent below if you agree to participate in the study.

CONSENT

I hereby confirm that I have been adequately informed by the researcher about the nature, conduct, benefits and risks of the study. I have also received, read and understood the above written information. I am aware that the results of the study will be anonymously processed into a research report. I understand that my participation is voluntary and that I may, at any stage, without prejudice, withdraw my consent and participation in the study. I had sufficient opportunity to ask questions and of my own free will declare myself prepared to participate in the study.

Research participant's name: _____ (Please print)

Research participant's signature: _____

Date: _____

Researcher's name: Ms L Nodada

Researcher's signature: _____

Date: 15 March 2017

An overview of tests for normality

Anri Oosthuysen 14008166

STK795 Research Report

Submitted in partial fulfillment of the degree BCom(Hons) Statistics

Supervisor: Dr IJH Visagie, Co-supervisor: Ms P Nagar

Department of Statistics, University of Pretoria

30 October 2017

Abstract

Often, when analysing data, a starting point is to determine if the data are normally distributed. In this study, we will compare various tests for normality based on different characteristics of the normal distribution. In each case the idea is to test the hypothesis of normality by comparing the empirical properties of the data to the corresponding theoretical properties of the normal distribution. Under normality the property upon which the test statistic is based on should hold approximately for the data, especially for large samples. The hypothesis of normality is rejected for large empirical deviations from the property in question. The first property of the data considered is the empirical distribution function. The tests considered that are based on this property are the Kolmogorov-Smirnov test, the Cramér-von Mises test and the Anderson-Darling test. Thereafter we discuss two tests based on the empirical moment generating function. The first of these are based on a weighted distance between the moment generating function of a normal distribution and its empirical counterpart, while the second is based on a differential equation characterising the normal distribution. We compare the power of the tests against various alternative distributions. The powers of the tests are estimated using Monte Carlo simulation.

Declaration

I, *Anri Berna Oosthuysen,* declare that this essay, submitted in partial fulfillment of the degree *BCom(Hons) Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Anri Berna Oosthuysen

_ _

IJH Visagie

_ _

P Nagar

_ _

Date

Acknowledgements

I acknowledge the financial support from the Center for AI Research, Meraka Institute, CSIR and STATOMET.

Contents

List of Figures

List of Tables

1 Introduction

The assumption of normality is essential for many statistical procedures. This distribution is used frequently both in practice and in theoretical work and the assumption of normality forms the basis of various inferential techniques; see [20]. Statistical procedures that rely on the assumption of normality, are called parametric methods. Non-parametric methods are used when the distribution of the data is unknown, meaning we do not assume that the data follow a specific distribution; see [17]. Consider, for example, the case where we would like to test for independence between two variables. In this case, non-parametric tests have often been found to be less effective than parametric tests in detecting a weak dependence between the variables. If, however, the data are normally distributed, parametric tests can be used, which will lead to the increase in the accuracy of the findings.

A random variable X is said to be normally distributed with $X \sim N(\mu, \sigma^2)$ if its probability density function is;

$$f_x(x) \quad = \quad \frac{1}{\sqrt{2\pi}\sigma} exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \quad -\infty < x < \infty,$$

see [15]. Some variations of the probability density function is illustrated in Figure 1.

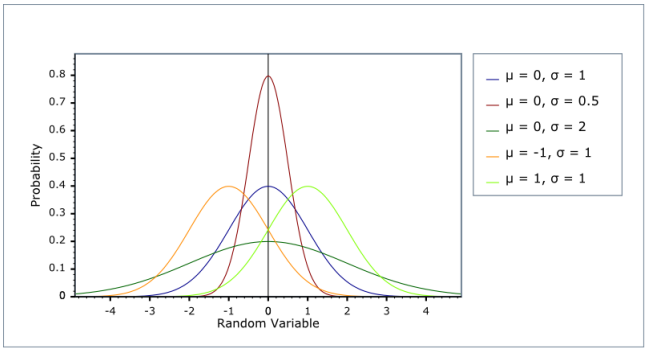


Figure 1: Normal density for various parameter combinations.

A normal or Gaussian density is bell-shaped and symmetric around its mean μ and has a variance σ^2; see [15]. The skewness and the kurtosis of the normal distribution are zero and three respectively.

Several authors have written review papers describing and comparing the different normality tests, see for example [6], [16], [19], [10], [20], [14], [5] and [12]. The different tests for normality are based on various properties of the normal distribution. In each case the idea is to compare the empirical properties of the data to the corresponding theoretical properties of the normal distribution. In each test we calculate a test statistic. This test statistics quantifies the discrepancy between the specified empirical property of the data and the corresponding theoretical property of the normal distribution. If this test statistic associated with a specific test indicates a large discrepancy, then we will reject the hypothesis

of normality. In this study, we are interested in testing whether or not observed data are realised from a normal distribution with some unknown $\mu \in \mathbb{R}$ and $\sigma^2 > 0$. In this research we will investigate various goodness-of-fit techniques based on two characteristics of the normal distribution; its distribution and moment generating functions.

Goodness-of-fit techniques are used in order to test the hypothesis that observed data are realised from a specific distribution or class of distributions. Let $X_1, X_2, ..., X_n$ be independent and identically distributed (i.i.d.) realisations from an unknown distribution F. If we want to test whether or not this sample is realised from a certain hypothesized distribution F_0, the following hypothesis is to be tested:

$$H_0 : F = F_{0,}$$
$$H_A : F \neq F_{0.}$$

The above hypothesis is known as a simple goodness-of-fit problem. If we want to test whether or not the sample is realised from a specific class of distributions, we would test a composite goodness-of-fit hypothesis. Under a simple hypothesis H_0 is fully specified, while a composite hypothesis only partially specifies the null hypothesis. In this research we will focus on the class of normal distributions. Let $\mathcal{N}$ be the class of normal distributions with some expected value $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$, the hypothesis that we are interested in testing is then:

$$\begin{aligned} H_0 &: F \in \mathcal{N}, \\ H_A &: F \notin \mathcal{N}. \end{aligned} \tag{1}$$

There are a variety of tests for normality available, in this research we will consider five of these tests. These tests include the Kolmogorov-Smirnov test, the Cramér-von Mises test and the Anderson-Darling test which are based on the empirical distribution function (EDF); see [20]. We look at two tests based on the moment generating function test, see [9]. The first of these tests compares the theoretical moment generating function to the empirical moment generating function (EMGF). The second test compares the first derivative of the empirical moment generating function to a function of the empirical moment generating function.

Each of the test statistics considered are based on the studentised values $Y_j = \frac{(X_j - \bar{X}_n)}{s}$, where $\bar{X}_n = \frac{1}{n}\sum_{j=1}^{n} X_j$ and $s^2 = \frac{1}{n}\sum_{j=1}^{n} \left(X_j - \bar{X}\right)^2$. We denote the ordered studentised values by $Y_{(1)} \leq ... \leq Y_{(n)}$. When using studentised values the hypothesis specified in (1) can be reduced to the hypothesis that the data are realised from a standard normal distribution.

The remainder of this research is organised as follows: In Section 2 we discuss various goodness-of-fit tests for normality as well as, the characteristics upon which these tests are based. We consider the different tests based on the empirical distribution function, these include the Kolmogorov-Smirnov test, the Cramér-von Mises test and the Anderson-Darling test. Next we will consider the tests based on the

moment generating function test. Section 3 shows Monte Carlo results obtained by generating samples from several non-normal distributions and testing the hypothesis of normality for each goodness-of-fit test discussed. The powers of the tests are estimated based on the rejection rate of the hypothesis test. In this research we use Monte Carlo simulation to estimate the critical values of the test statistics under the null hypothesis. We discuss the Monte Carlo simulation results, as well as compare the power of the tests against various alternatives and then some conclusions relating to the most powerful tests are drawn. In Section 4 we discuss a practical example for testing the normality of the observed data.

2 Literature review

We consider three different tests based on the empirical distribution function; the Kolmogorov-Smirnov test, the Cramér-von Mises test and the Anderson-Darling test. Next we discuss two tests based on the empirical moment generating function.

2.1 Tests based on the empirical distribution function

Let $x_1, x_2, ..., x_n$ denote an observed sample of size n from a random variable X. The goodness-of-fit tests considered below are based on a measure of distance between the empirical distribution function and the distribution function. The idea is to compare the empirical distribution function test, with the distribution function of the normal distribution to see if there is a close correspondence between them; see [20] and [14].

The empirical distribution function is a step function used as a proxy for the underlying distribution function. The empirical distribution function evaluated in some point $x \in \mathbb{R}$, $F_n(x)$, is the fraction of observations that are less than or equal to x. The empirical distribution function is defined as;

$$F_n(x) \quad = \quad \frac{1}{n} \sum_{i=1}^{n} I\left(X_i \leq x\right),$$

where $I(\cdot)$ denotes the indicator function.

Denote the ordered sample by $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)}$. The empirical distribution function can be rewritten in the following form;

$$F_n(x) = \begin{cases} 0 & for\ x < X_{(1)}, \\ \frac{i}{n} & for\ X_{(i)} \leq x \leq X_{(i+1)},\ i = 1, 2, \ldots, n-1, \\ 1 & for\ x \geq X_{(n)}, \end{cases}$$

see [16]. As a result, the empirical distribution function forms a step function with n jumps of size $\frac{1}{n}$ at

each of the observed sample values.

Let F denote the distribution function of X. Figure 2 shows an example of an empirical distribution function obtained from a sample of size 20 of studentised values, drawn from a normal distribution. The standard normal distribution function is superimposed in the figure. The tests for normality based on the empirical distribution function typically measure some distance between the empirical distribution function and the distribution function of the standard normal distribution.

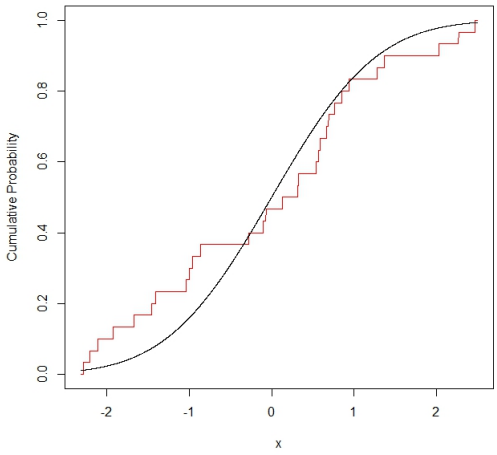


Figure 2: EDF of a sample from a standard normal distribution with superimposed distribution function.

The test based on the empirical distribution function can be divided into two classes, one being the supremum class which is based on the largest difference between $F_n(x)$ and $F(x)$, and the other being the quadratic class which is based on the square difference $(F_n(x) - F(x))^2$; see [4]. The Kolmogorov-Smirnov test falls under the supremum class and the Cramér-von Mises test and the Anderson-Darling test falls under the quadratic class of the EDF tests.

2.1.1 The Kolmogorov-Smirnov test

In 1933 Andrey Nikolaevich Kolmogorov proposed the first test for standard normality based on observed samples. In 1939 Valdimir Ivanovich Smirnov used a test statistic similar to that proposed by Kolmogorov, but based on the maximal distance between the EDF's of two samples; see [11]. The one sample test is for testing whether or not a set of observations is from a specified continuous distribution. In [1], Lilliefors extended this test to testing the more general hypothesis of normality without specifying the values of μ and σ^2. The resulting test is also known as the Lilliefors test, which was developed independently by Lilliefors (1967) in [11] and by Van Soest (1967) in [18]. The Lilliefors test is based on studentised data. The original Kolmogorov-Smirnov (KS) test assumes specified parameters of the hypothesized distribution which are known in advance. The Lilliefors test, on the other hand, is based on studentised

sample values. As a result, this test can be used to test the general hypothesis specified in (1) .

The test statistic of the Kolmogorov-Smirnov test is defined as:

$$KS_n \quad = \quad \sup\left\{\left|F_n\left(x\right) - F\left(x\right)\right|\right\}, \quad for\ x \in \mathbb{R}.$$

For computational purposes, the Kolmogorov-Smirnov test can be decomposed into the three parts as follows:

$$KS_n \quad = \quad \max\left(KS_n^+, KS_n^-\right),$$

where KS_n^+ is the greatest vertical difference when $F\left(x\right)$ is greater than $F_n\left(x\right)$;

$$KS_n^+ = \sup\left\{F(x) - F_n(x)\right\},$$

and KS_n^- measures the greatest vertical difference when $F_n(x)$ exceeds $F(x)$;

$$KS_n^- \quad = \quad \sup\left\{F_n(x) - F(x)\right\}.$$

The Kolmogorov-Smirnov test rejects for large values of KS_n.

2.1.2 The Cramér-von Mises test

The Cramér-von Mises (CvM) test statistic falls within the quadratic class of statistics, since it is based on the squared vertical difference between $F_n(x)$ and the hypothesized distribution $F\left(x\right)$; see [14]. The Cramér-von Mises test is an omnibus test, meaning that this test has substantial power against all non-normal alternatives for large samples. In 1928 Cramér [7] introduced the following test statistic;

$$\omega^2 \quad = \quad \int_{-\infty}^{\infty} \left(F_n(x) - F(x)\right)^2 dK(x),$$

where $K(x)$ is some kernel function. H_0, given in (1), is to be rejected if the test statistic defined in ω^2 is too large. Von Mises [7] then made a few suggestions and developed a new test statistic;

$$W_n^2 \quad = \quad n \int_{-\infty}^{\infty} \left[\left(F_n(x) - F(x)\right)^2\right] \psi(x) dF(x), \tag{2}$$

where $\psi(x)$ is a weight function. Setting $\psi(x) = 1$ defines the Cramér-von Mises test. The test statistic in (2) can be written as;

$$CVM_n = \frac{1}{12n} + \sum_{i=1}^{n} \left(Z_i - \frac{2i-1}{2n} \right)^2,$$

where $Z_i = \phi \frac{(X_i - \bar{X})}{S}$, with

$$\phi(x) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{1}{2} x^2 \right).$$

Large values of the Cramér-von Mises will lead to rejection of the null hypothesis. Stephens suggested that the Cramér-von Mises and the Anderson-Darling tests are some of the best pairs of empirical distribution function tests for the testing of normality in terms of powers; see [4] and [19].

2.1.3 The Anderson-Darling test

In 1952 and 1954 Theodore W. Anderson and D.A Darling developed the Anderson-Darling (AD) test for specific values of the parameters μ and σ^2; see [10]. In the 1970's Stephens adapted the test statistic proposed by Anderson and Darling for parameters that may be unknown; see [10].

The Anderson-Darling test proposed in [2], is a modification of the Cramér-von Mises test, in which the weight function $\psi(x)$ is chosen such that more weight is given to the tails of the distribution than is the case in the Cramér-von Mises test. The Anderson-Darling test has been shown to exhibit relatively high powers; see [3]. For each fixed sample size the critical values of the Anderson-Darling test needs to be estimated using Monte Carlo simulations.

The Anderson-Darling test statistic is similar to the Cramér-von Mises test statistic, the modification lies in the form of the weight function. The test statistic for the Anderson-Darling test is:

$$W_n^2 = n \int_{-\infty}^{\infty} \left[(F_n(x) - F^*(x))^2 \right] \left([F(x)(1 - (F(x))]^{-1} \right) dF(x). \tag{3}$$

Note that the Anderson-Darling test is obtained by replacing $\psi(x) = 1$ with $\psi(x) = \left([F(x)(1 - (F(x))]^{-1} \right)$ in (2).

The test statistic in (3) can be written as;

$$A^2 = -\sum_{i=1}^{n} \left[\frac{(2i-1)\left(logP_i + log\left(1 - P_{n+1-i}\right)\right)}{n} \right] - n,$$

where

$$P_i = \Phi(Y_i) = \int_{-\infty}^{Y_i} \frac{e^{-\frac{1}{2}t^2}}{\sqrt{2\pi}} dt,$$

see [4].

2.2 Tests based on the empirical moment generating function

The moment generating function of a random variable X characterises its probability distribution. The moment generating function is:

$$M(t) = E\left(e^{tX}\right).$$

We consider two different tests based on the empirical moment generating function. The powers of these two tests based on this function are compared to the tests discussed previously.

Both test statistics considered below are similar in form to the Cramér-von Mises test. The first is based on a weighted L^2-distance between the moment generating function of the standard normal distribution and the empirical moment generating function of the studentised values of the sample. The second test is based on a differential equation characterising the standard normal distribution.

2.2.1 A test based on the empirical moment generating function

In [5], Epps proposed a test statistic based on the weighted squared difference between the moment generating function and its empirical counterpart. The proposed test statistic is demonstrated as follow:

$$T_{n,\beta}^{(1)} = n\int_{-\infty}^{\infty}\left[M_n(t) - M_X(t)\right]^2 e^{-\beta t^2}dt, \tag{4}$$

where $M_X(t)$ is the moment generating function of the standard normal distribution and $M_n(t)$ is the empirical moment generating function of $Y_1,\ldots,Y_n$. The weight function $e^{-\beta t^2}$, with $\beta > 0$, is required in order to ensure that the above integral is finite. The empirical moment generating function is given by:

$$M_n(t) = \frac{1}{n}\sum_{j=1}^{n} e^{tY_{n,j}}, \quad t \in \mathbb{R}.$$

In (4), $M_X(t)$ is the moment generating function of a standard normal distribution. Below we derive the moment generating function for the normal distribution with mean μ and variance σ^2,

$$\begin{aligned} M_X(t) &= E\left[e^{tX}\right] \\ &= \int_{-\infty}^{\infty} e^{tx}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}dx \end{aligned}$$

$$
\begin{aligned}
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}\left(-2tx\sigma^2+(x-\mu)^2\right)} dx \\
&= e^{\mu t+\frac{1}{2}t^2\sigma^2} \left[\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\left(\mu+t\sigma^2\right)}{\sigma}\right)^2} dx \right] \\
&= exp\left(\mu t + \frac{1}{2}\sigma^2 t^2\right), \quad t \in \mathbb{R}. \qquad (5)
\end{aligned}
$$

Using (5), we see that the moment generating function for the standard normal distribution is given by:

$$
M_X(t) = exp(\frac{1}{2}t^2), \quad t \in \mathbb{R}. \qquad (6)
$$

If the data are realised from a normal distribution then the studentised residuals $Y_1, Y_2..., Y_n$ should be approximately standard normally distributed, especially for large samples. Therefore, for large samples, $M_n(t)$ should be approximately equal to $M_X(t)$; see [9].

The test statistic in (4) can be rewritten as:

$$
T_{n,\beta}^{(1)} = \sqrt{\pi}\left(\frac{1}{n\sqrt{\beta}} \sum_{j,k=1}^{n} exp\left(\frac{(Y_j+Y_k)^2}{4\beta} \right) - \frac{2}{\sqrt{\beta-\frac{1}{2}}} \sum_{j=1}^{n} exp\left(\frac{Y_j^2}{4\beta-2} \right) + \frac{n}{\sqrt{\beta-1}} \right); \qquad (7)
$$

see [9]. For a derivation of equation (7), see Appendix 1. This test rejects normality for large values of $T_{n,\beta}^{(1)}$.

2.2.2 A test based on a differential equation of the empirical moment generating function

The moment generating function $M_X(t)$ of a standard normal distribution is given in (6). The derivative of $M_X(t)$ is given by:

$$
M_X'(t) = te^{\frac{1}{2}t^2} = tM_X(t), \quad t \in \mathbb{R}.
$$

Under normality, the above relation should approximately hold for the empirical moment generating function and its derivative. The derivative of the empirical moment generating function is

$$
M_n'(t) = \frac{d}{dt}\frac{1}{n}\sum_{j=1}^{n} e^{tX_j} = \frac{1}{n}\sum_{j=1}^{n} e^{tX_j} X_j.
$$

The test statistic is based on the squared difference between the derivative of the empirical moment generating function, $M_n'(t)$, and the empirical moment generating function multiplied by t, $tM_n(t)$.

The test statistic is

$$\begin{aligned} T_{n,\beta}^{(2)} &= \int_{-\infty}^{\infty} \left(M_n^{'}(t) - tM_n(t) \right)^2 e^{-\beta t^2} dt \\ &= n \int_{-\infty}^{\infty} \left(\frac{1}{n} \sum_{j=1}^{n} e^{tX_j} X_{j.} - t \frac{1}{n} \sum_{j=1}^{n} e^{tX_j} \right)^2 e^{-\beta t^2} dt, \end{aligned} \tag{8}$$

where $e^{-\beta t^2}$ is a weight function with $\beta > 0$. This weight function is required in order to ensure that the test statistic is finite. The test statistic in (8) can be rewritten as;

$$T_{n,\beta}^{(2)} = \sqrt{\frac{\pi}{\beta}} \frac{1}{n} \sum_{i,j=1}^{n} exp\left(\frac{Z_{n,i,j}^2}{4\beta} \right) \left\{ Y_i Y_j + \frac{1 - 2Y_i Z_{i,j}}{2\beta} + \frac{Z_{n,i,j}^2}{4\beta^2} \right\}, \tag{9}$$

where $Z_{n,i,j} = Y_{n,i} + Y_{n,j}$. For a derivation of the equation (9), see Appendix 2. Normality is rejected for large values of $T_{n,\beta}^{(2)}$.

3 Simulation study

In this research we use Monte Carlo simulation to estimate critical values for the various tests discussed in Section 2 using 100 000 Monte Carlo replications. We estimate the powers of the tests against various alternative distributions. These powers are estimated by the proportion of 10 000 samples, for which the null hypothesis of normality is rejected. The statistical software package R is used in order to obtain the numerical results reported below; see [13]. The sample sizes used are, $n = 20, 50, 100$, and a nominal level of significance of $\alpha = 5\%$ is used throughout. We include two tests from each of the classes of the tests based on the empirical moment generating function, in each case the tuning parameter are chosen to be $\beta = 5$ and $\beta = 10$ respectively.

In R we use the *nortest* package; see [8], in order to calculate the power estimate of the Kolmogorov-Smirnov test, the Cramér-von Mises test and the Anderson-Darling test. The powers of the empirical moment generating function tests are estimated using the code provided in Appendix 3.

The 95^{th} percentile of the realised values of the test statistic based on simulation from a $N(0,1)$ distribution is used in order to estimate the critical values of the tests in question. We use two versions of $T_{n,\beta}^{(1)}$ and $T_{n,\beta}^{(2)}$ that is obtained by setting $\beta = 5$ and $\beta = 10$. The powers of the seven different tests are calculated for the given distributions. We consider distributions from each of the following classes of distributions, the symmetric short-tailed class, the symmetric long-tailed class and the asymmetric class. The three symmetric short-tailed distributions considered are the uniform distribution, $Uni(0,1)$, the triangular distribution, and the truncated normal distribution, $Trunc(-2,2)$. The three symmetric long-tailed distribution are the Cauchy, $t(3)$ and $t(5)$ distributions. For the asymmetric distribution we

considered four different distributions including the log-normal, Weibull(5), Weibull(10) and the skew-normal distribution. The density function of the skew-normal distribution is:

$$f\left(x;\mu,\sigma^2,\alpha\right) \quad = \quad \frac{1}{\sigma\pi} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \int_{-\infty}^{\alpha\left(\frac{x-\mu}{\sigma}\right)} exp\left(-\frac{t^2}{2}\right) dt, \quad x \in \mathbb{R}.$$

A standard skew-normal distribution with skewness parameter 5, $SN(0,1,5)$ is used below.

Table 1: Realised power of the tests for normality in percentage for a sample size of n=20

Distribution	KS_{20}	CvM_{20}	AD_{20}	${T^{(1)}}_{20,5}$	${T^{(1)}}_{20,10}$	${T^{(2)}}_{20,5}$	${T^{(2)}}_{20,10}$
$N(0,1)$	5	5	5	5	5	5	5
$Uni(0,1)$	10	15	18	0	0	1	0
$Triangular$	4	4	4	1	1	1	1
$Trunc(-2,2)$	4	4	4	1	1	1	1
$Cauchy$	85	88	88	82	83	80	81
$t(3)$	27	32	34	36	37	36	36
$t(5)$	14	17	18	22	23	22	22
$Log-normal$	79	88	91	83	82	85	84
$Sn(n,0,1,5,0)$	18	22	25	22	21	23	23
$Weibull(5)$	6	6	6	5	5	5	5
$Weibull(10)$	11	12	14	15	15	15	15

Table 1 reports the estimated powers of the various tests for normality, for a sample size of $n = 20$. First, we consider the symmetric short-tailed class. The Anderson-Darling test achieves a higher power for testing whether or not data are normally distributed, and the moment generating function test performs the poorest. The empirical distribution function tests outperform the empirical moment generating function tests. When considering the symmetric long-tailed class, the empirical distribution function tests have better powers for the Cauchy distribution. For the $t\,(3)$ and the $t\,(5)$ distributions, the empirical moment generating function tests have higher powers. The Anderson-Darling test performs best against members of the symmetric long-tailed class. For the asymmetric class, the empirical distribution function tests performs slightly better for the log-normal, the skewed-normal and the Weibull(5) distributions. For the Weibull(10) distribution the empirical moment generating function tests have higher powers.

In the symmetric long-tailed class and asymmetric class, we have quite comparable powers between the empirical distribution function tests and the empirical moment generating function tests. For the symmetric short-tailed class the empirical moment generating function tests exhibit low powers. The Anderson-Darling test performs best overall. The observed powers of all four tests based on the empirical moment generating function are quite similar.

Table 2 and 3 report the estimated powers of the various tests for normality for a sample size of $n = 50$ and $n = 100$ respectively. The results shown in Tables 2 and 3 are similar to those reported in Table 1. We note that as the sample size n increases, the powers of the different tests also increase.

Table 2: Realised power of the tests for normality in percentage for a sample size of n=50

Distribution	KS_{50}	CvM_{50}	AD_{50}	${T^{(1)}}_{50,5}$	${T^{(1)}}_{50,10}$	${T^{(2)}}_{50,5}$	${T^{(2)}}_{50,10}$
$N(0,1)$	5	5	5	5	5	5	5
$Uni(0,1)$	26	44	58	0	0	0	0
$Triangular$	4	5	5	0	0	0	0
$Trunc(-2,2)$	5	5	6	0	0	0	0
$Cauchy$	99	100	100	98	99	98	98
$t(3)$	49	57	61	62	63	59	60
$t(5)$	21	28	31	38	39	36	36
$Log-normal$	100	100	100	100	100	100	100
$Sn(n,0,1,5,0)$	41	53	58	49	47	56	54
$Weibull(5)$	8	8	8	6	6	7	7
$Weibull(10)$	20	25	28	32	31	35	35

Table 3: Realised power of the tests for normality in percentage for a sample size of n=100

Distribution	KS_{100}	CvM_{100}	AD_{100}	${T^{(1)}}_{100,5}$	${T^{(1)}}_{100,10}$	${T^{(2)}}_{100,5}$	${T^{(2)}}_{100,10}$
$N(0,1)$	5	5	5	5	5	5	5
$Uni(0,1)$	59	84	95	0	0	0	0
$Triangular$	5	6	8	0	0	0	0
$Trunc(-2,2)$	6	8	9	0	0	0	0
$Cauchy$	100	100	100	100	100	100	100
$t(3)$	75	84	86	84	85	79	81
$t(5)$	33	43	48	56	57	51	52
$Log-normal$	100	100	100	100	100	100	100
$Sn(n,0,1,5,0)$	73	85	90	84	82	89	88
$Weibull(5)$	11	12	14	9	8	12	11
$Weibull(10)$	38	47	53	58	57	64	63

The graph representing the EDF tests, in Figure 3 and 4, has different colors, each representing a normality test. The black line represents the Anderson-Darling test, the blue line the Cramér-von Mises test, and the red line the Kolmogorov-Smirnov test. The graph representing the EMGF test colors are as follows, the blue line represents the $T^{(1)}_{n,10}$ test, the black line the $T^{(2)}_{n,10}$ test, the red line the $T^{(1)}_{n,5}$ test and the green line the $T^{(2)}_{n,5}$ test. Figure 3 below illustrates the powers that the various tests achieve against the skewed normal distribution as a function of a sample size. We see that among the empirical distribution function tests, the Anderson-Darling test has the highest powers, followed by the Cramér-von Mises test and the Kolmogorov-Smirnov test. Figure 3 shows that for the empirical moment generating function the $T^{(1)}_{n,10}$ test exhibits the highest powers, followed by the $T^{(2)}_{n,10}$, the $T^{(1)}_{n,5}$ and then the $T^{(2)}_{n,5}$ test. Figure 4 shows the powers for the $t\,(3)$ distribution. We see that for the empirical distribution function the Anderson-Darling test has the highest powers, followed by the Cramér-von Mises test and then the Kolmogorov-Smirnov test. For the empirical moment generating function, Figure 4 shows that the $T^{(2)}_{n,5}$ test exhibits the highest powers followed by the $T^{(1)}_{n,5}$, the $T^{(2)}_{n,10}$ and then the $T^{(1)}_{n,10}$ test.

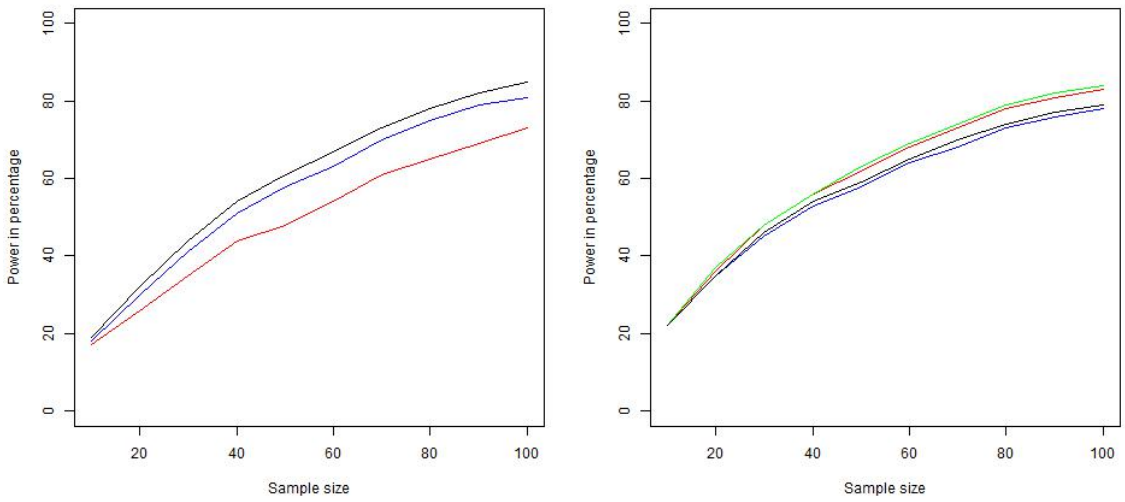


Figure 3: Powers of various tests against the skewed normal distribution, for the EDF test (left) and the EMGF test (right).

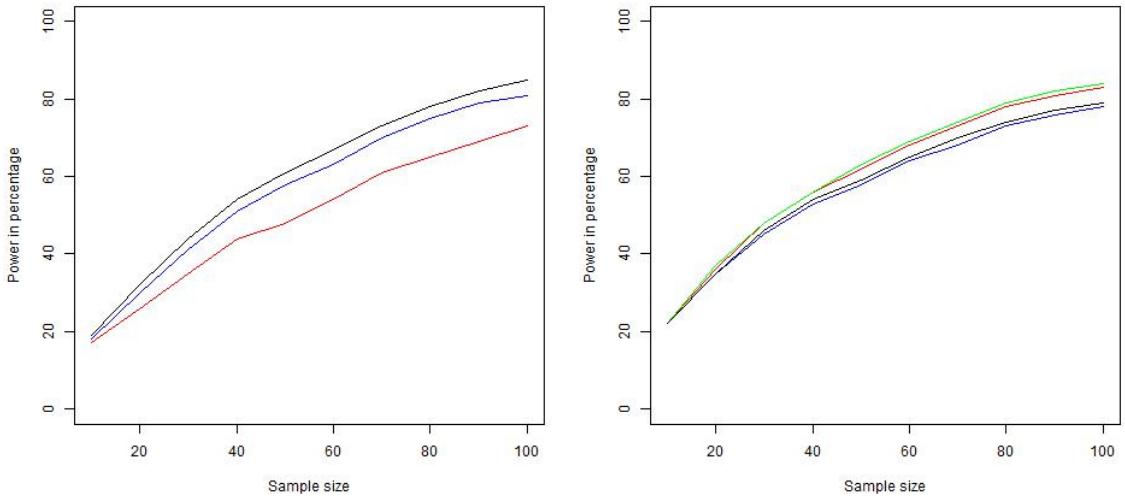


Figure 4: Powers of various tests against the $t(3)$ distribution, for the EDF test (left) and the EMGF test (right).

4 Testing the normality of observed data

A practical application of these seven tests listed above was implemented in SAS[1]. We use a data set of 20 observations of the average fruit weight (grams) of apples per tree for 20 trees in an agricultural experiment[2]. Using *Proc iml* we calculate the test statistics of the different empirical moment generating function tests. Using *Proc Univariate* in SAS we calculated the test statistic for the empirical distribution

[1]Copyright (c) 2002-2012 by SAS Institute Inc., Cary, NC, USA. All Rights Reserved
[2]http://college.cengage.com/mathematics/brase/understandable_statistics/7e/students/datasets/svls/frames/frame.html

function tests. Table 4 reports these test statistics along with the critical values obtained in R.

	Test statistic	Critical values $n = 20$
Kolmogorov-Smirnov	0.1795	0.1911
Cramér-von Mises	0.1481	0.1220
Anderson-Darling	0.7992	0.7173
$T_{n,5}^{(1)}$	0.0080	0.0112
$T_{n,10}^{(1)}$	0.0004	0.0007
$T_{n,5}^{(2)}$	0.1174	0.1576
$T_{n,10}^{(2)}$	0.0145	0.0226

Table 4: Goodness-of-fit tests for normal distribution

In each case we reject the normality if the test statistic is greater than the critical value. We can conclude that the Anderson-Darling test and the Cramér-von Mises test reject the hypothesis of normality. The remaining five normality tests have test statistics smaller than the critical values, indicating that we do not reject for normality.

Figure 5 illustrates the empirical distribution of the data together with the estimated normal density.

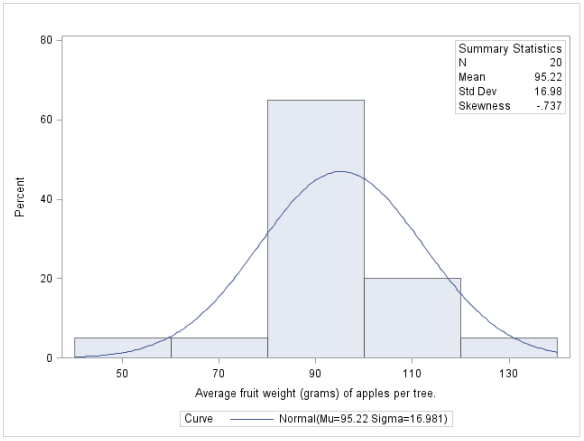


Figure 5: Average fruit weight (grams) of apples per tree for 20 trees in an agricultural experiment.

5 Conclusion

In this research, we compare several different normality tests in terms of the power achieved against various alternative distributions. From the results obtained we can conclude that the Anderson-Darling test has higher powers compared to the remaining normality tests. Comparing the empirical distribution function and the empirical moment generating function tests we conclude that the empirical distribution function tests have higher powers than the empirical moment generating function tests, except in the case of the symmetric long-tailed class of distributions. Here the empirical moment generating function

tests outperform the other tests. In general, we can see that the Kolmogorov-Smirnov test achieved the lowest power among the other empirical distribution function tests for normality.

References

[1] H. Abdi and P. Molin. Lilliefors/van Soest test of normality. *2007), Encyclopedia of Measurement and Statistics. Available at: http://www. utdallas. edu/ herve/Abdi-Lillie2007-pretty. pdf*, 2007.

[2] T. W. Anderson and D.A. Darling. A test of goodness of fit. *Journal of the American statistical association*, 49(268):765–769, 1954.

[3] M. Arshad, M.T. Rasool, and M.I. Ahmad. Anderson Darling and modified Anderson Darling tests for generalized pareto distribution. *Pakistan Journal of Applied Sciences*, 3(2):85–88, 2003.

[4] R.B. D'Agostino and M.A. Stephens. *Goodness-Of-Fit Techniques*, volume 68. Marcel Dekker, Inc, 1986.

[5] T.W. Epps, K.J. Singleton, and L.B. Pulley. A test of separate families of distributions based on the empirical moment generating function. *Biometrika*, 69(2):391–399, 1982.

[6] R. C. Geary. Testing for normality. *Biometrika*, 34(3/4):209–242, 1947.

[7] K. Ghoudi and D. Mcdonald. Cramer-von Mises regression. *Canadian Journal of Statistics*, 28(4):689–714, 2000.

[8] J. Gross and U. Ligges. *nortest: Tests for Normality*, 2015. R package version 1.0-4.

[9] N. Henze and S. Koch. On a test of normality based on the empirical moment generating function. *Statistical Papers*, pages 1–13, 2016.

[10] F. Laio. Cramer-von Mises and Anderson-Darling goodness of fit tests for extreme value distributions with unknown parameters. *Water Resources Research*, 40(9), 2004.

[11] H.W. Lilliefors. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American statistical Association*, 62(318):399–402, 1967.

[12] S.G. Meintanis. A review of testing procedures based on the empirical characteristic function. *South African Statistical Journal*, 50(1):1–14, 2016.

[13] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[14] N. Razali and B.W. Yap. Power comparisons of Shapiro-wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests. *Journal of Statistical Modeling and Analytics*, 2, June 2011.

[15] M.I Ribeiro. Gaussian probability density functions: Properties and error characterization. *Institute for Systems and Robotics, Lisboa, Portugal*, 2004.

[16] H. Riedwyl. Goodness of fit. *Journal of the American Statistical Association*, 62(318):390–398, 1967.

[17] D.J. Sheskin. *Handbook of parametric and nonparametric statistical procedures.* CRC Press, 2003.

[18] J. Soest. Some experimental results concerning tests of normality. *Statistica Neerlandica*, 21(1):91–97, 1967.

[19] M. A. Stephens. EDF statistics for goodness-of-fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730–737, 1974.

[20] B. W. Yap and C. H. Sim. Comparisons of various types of normality tests. *Journal of Statistical Computation and Simulation*, 81(12):2141–2155, 2011.

Appendix 1: Derivation of equation (7).

The equation (7) of the empirical moment generating function is derived below. Note that one of the terms in the derivation can be simplified as follow:

$$\begin{aligned}
exp\left(t\left(Y_j+Y_k\right)\right)exp\left(-\beta t^2\right) &= exp\left[-\left(\beta t^2-t\left(Y_j+Y_k\right)\right)\right] \\
&= exp\left[-\left(\left(\sqrt{\beta}t-\frac{Y_j+Y_k}{2\sqrt{\beta}}\right)^2-\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right)\right] \\
&= exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right)exp\left(-\left(\sqrt{\beta}t-\frac{Y_j+Y_k}{2\sqrt{\beta}}\right)^2\right) \\
&= exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right)exp\left(-\frac{1}{2}\left(\frac{t-\frac{Y_j+Y_k}{2\sqrt{\beta}}}{\frac{1}{2}\frac{1}{\sqrt{\beta}}}\right)^2\right) \\
&= exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right)exp\left(-\frac{1}{2}\left(\frac{t-\frac{Y_j+Y_k}{2\sqrt{\beta}}}{\frac{1}{\sqrt{2\beta}}}\right)^2\right)\frac{1}{\sqrt{2\pi}\frac{1}{\sqrt{2\beta}}}\sqrt{2\pi}\frac{1}{\sqrt{2\beta}} \\
&= exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right)\sqrt{\frac{\pi}{\beta}}\frac{1}{\sqrt{2\pi}\frac{1}{\sqrt{2\beta}}}exp\left(-\frac{1}{2}\left(\frac{t-\frac{Y_j+Y_k}{2\sqrt{\beta}}}{\frac{1}{\sqrt{2\beta}}}\right)^2\right) \\
&= \sqrt{\frac{\pi}{\beta}}exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right)\phi_T\left(t\right), \\
&\quad where\ T\sim N\left(\frac{\left(Y_j+Y_k\right)}{2\beta},\frac{1}{2\beta}\right). \qquad (10)
\end{aligned}$$

Equation 7 can be rewritten as:

$$\begin{aligned}
T_{n\beta}^{(1)} &= n\int_{-\infty}^{\infty}\left(M_n\left(t\right)-M_x\left(t\right)^2\right)exp\left(-\beta t^2\right)dt \\
&= n\int_{-\infty}^{\infty}\left(\left(M_n\left(t\right)^2-2M_n\left(t\right)M_x\left(t\right)+M_x\left(t\right)^2\right)exp\left(-\beta t^2\right)\right)dt \\
&= n\int_{-\infty}^{\infty}M_n\left(t\right)^2exp\left(-\beta t^2\right)dt-2n\int_{-\infty}^{\infty}M_n\left(t\right)M_x\left(t\right)exp\left(-\beta t^2\right)dt+n\int_{-\infty}^{\infty}M_x\left(t\right)^2exp\left(-\beta t^2\right)dt \\
&= nI_1-2nI_2+nI_3. \qquad (11)
\end{aligned}$$

Below we consider I_1, I_2 and I_3 respectively.

$$\begin{aligned}
I_1 &= \int_{-\infty}^{\infty} \left[M_n(t)^2\right] exp\left(-\beta t^2\right) dt \\
&= \frac{1}{n^2} \int_{-\infty}^{\infty} \left[\sum_{j=1}^{n} exp\left(tY_j\right)\right]^2 exp\left(-\beta t^2\right) dt \\
&= \frac{1}{n^2} \sum_{j=1}^{n}\sum_{k=1}^{n} \int_{-\infty}^{\infty} \sqrt{\frac{\pi}{\beta}} exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right) \phi_T(t)\, dt \\
&= \frac{1}{n^2}\sqrt{\frac{\pi}{\beta}} \sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right) \int_{-\infty}^{\infty} \phi_T(t)\, dt \\
&= \frac{1}{n^2}\sqrt{\frac{\pi}{\beta}} \sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right), \qquad (12) \\
&\quad \textit{see equation } 10.
\end{aligned}$$

$$\begin{aligned}
I_2 &= \int_{-\infty}^{\infty} M_n(t)\, M_x(t)\, exp\left(-\beta t^2\right) dt \\
&= \int_{-\infty}^{\infty} \frac{1}{n} \sum_{j=1}^{n} exp\left(tY_j\right) exp\left(-\frac{1}{2}t^2\beta t^2\right) dt \\
&= \frac{1}{n} \sum_{j=1}^{n} \int_{-\infty}^{\infty} exp\left(tY_j + \frac{1}{2}t^2 - \beta t^2\right) dt \\
&= \frac{1}{n} \sum_{j=1}^{n} \int_{-\infty}^{\infty} exp\left(-\left(\beta t^2 - \frac{1}{2}t^2 - tY_j\right)\right) dt \\
&= \frac{1}{n} \sum_{j=1}^{n} \int_{-\infty}^{\infty} exp\left(-\left[\left(\beta - \frac{1}{2}\right)t^2 - tY_j\right]\right) dt \\
&= \frac{1}{n} \sum_{j=1}^{n} \int_{-\infty}^{\infty} exp\left(-\left[\left(\beta - \frac{1}{2}\right)t^2 - Y_j t + \frac{Y_j^2}{4\left(\beta-\frac{1}{2}\right)}\right] + \frac{Y_j^2}{4\left(\beta-\frac{1}{2}\right)}\right) dt \\
&= \frac{1}{n} \sum_{j=1}^{n} exp\frac{Y_j^2}{4\left(\beta-\frac{1}{2}\right)} \int_{-\infty}^{\infty} exp\left(-\left[\left(\beta - \frac{1}{2}\right)t^2 - Y_j t + \frac{Y_j^2}{4\left(\beta-\frac{1}{2}\right)}\right] + \frac{Y_j^2}{4\left(\beta-\frac{1}{2}\right)}\right) dt \\
&= \frac{1}{n} \sum_{j=1}^{n} exp\frac{Y_j^2}{4\left(\beta-\frac{1}{2}\right)} \int_{-\infty}^{\infty} exp\left(-\left[\sqrt{\beta-\frac{1}{2}}\,t - \frac{Y_j}{2\sqrt{\beta-\frac{1}{2}}}\right]^2\right) dt
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{n}\sum_{j=1}^{n} exp\frac{Y_j^2}{4\left(\beta-\frac{1}{2}\right)} \int_{-\infty}^{\infty} exp\left(-\left[\frac{t-\frac{Y_j}{2\left(\beta-\frac{1}{2}\right)}}{\frac{1}{\sqrt{\beta-\frac{1}{2}}}}\right]^2\right)dt \\
&= \frac{1}{n}\sum_{j=1}^{n} exp\left(\frac{Y_j^2}{4\beta-2}\right) \int_{-\infty}^{\infty} exp\left(\frac{-\frac{1}{2}\left(t-\frac{Y_j}{2\beta-1}\right)^2}{\frac{1}{2\beta-\frac{1}{2}}}\right)\frac{1}{\sqrt{2\pi\left(\frac{1}{2}\beta-1\right)}}dt\sqrt{\frac{\pi}{\beta-\frac{1}{2}}} \\
&= \frac{1}{n}\sqrt{\frac{\pi}{\beta-\frac{1}{2}}}\sum_{j=1}^{n} exp\left(\frac{Y_j^2}{4\beta-2}\right). \qquad (13)
\end{aligned}
$$

$$
\begin{aligned}
I_3 &= \int_{-\infty}^{\infty} \left(M_x(t)\right)^2 exp\left(-\beta t^2\right)dt \\
&= \int_{-\infty}^{\infty} exp\left(\frac{1}{2}t^2\right)^2 exp\left(-\beta t^2\right)dt \\
&= \int_{-\infty}^{\infty} exp\left(t^2-\beta t^2\right)dt \\
&= \int_{-\infty}^{\infty} exp\left(-\left(\beta-1\right)t^2\right)dt \\
&= \int_{-\infty}^{\infty} exp\left(\frac{-t^2}{\frac{n}{\beta-1}}\right)dt \\
&= \int_{-\infty}^{\infty} exp\left(-\frac{1}{2}\left(\frac{t^2}{\frac{1}{2(\beta-1)}}\right)\right)\frac{1}{\sqrt{2\pi\frac{1}{2(\beta-1)}}}dt \\
&= \sqrt{\frac{\pi}{\beta-1}}. \qquad (14)
\end{aligned}
$$

Substituting (12), (13) and (14) into (11), we obtain

$$
\begin{aligned}
T_{n,\beta}^{(1)} &= nI_1 - 2nI_2 + nI_3 \\
&= n\frac{1}{n^2}\sqrt{\frac{\pi}{\beta}}\sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right) - 2n\frac{1}{n}\sqrt{\frac{\pi}{\beta-\frac{1}{2}}}\sum_{j=1}^{n} exp\left(\frac{Y_j^2}{4\beta-2}\right) + n\sqrt{\frac{\pi}{\beta-1}} \\
&= \sqrt{\pi}\left(\frac{1}{n\sqrt{\beta}}\sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right) - \frac{2}{\sqrt{\beta-\frac{1}{2}}}\sum_{j=1}^{n} exp\left(\frac{Y_j^2}{4\beta-2}\right) + \frac{n}{\sqrt{\beta-1}}\right)
\end{aligned}
$$

$$
T_{n,\beta}^{(1)} = \sqrt{\pi}\left(\frac{1}{n\sqrt{\beta}}\sum_{j,k=1}^{n} exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right) - \frac{2}{\sqrt{\beta-\frac{1}{2}}}\sum_{j=1}^{n} exp\left(\frac{Y_j^2}{4\beta-2}\right) + \frac{n}{\sqrt{\beta-1}}\right). \qquad (15)
$$

Appendix 2: Derivation of equation (9).

The equation (9) of the empirical moment generating function is derived below.

$$T_{n,\beta}^{(2)} = n\int_{-\infty}^{\infty}\left(M_n^{'}(t)-tM_n(t)\right)^2 exp\left(-\beta t^2\right)dt.$$

$$\begin{aligned} M_n(t) &= \frac{1}{n}\sum_{j=1}^{n} exp\left(tY_j\right). \\ M_n^{'}(t) &= \frac{1}{n}\sum_{j=1}^{n} Y_j exp\left(tYj\right). \end{aligned}$$

Equation 9 can be rewritten as:

$$\begin{aligned} T_{n,\beta}^{(2)} &= n\int_{-\infty}^{\infty}\left(M_n^{'}(t)-tM_n(t)\right)^2 exp\left(-\beta t^2\right)dt \\ &= n\int_{-\infty}^{\infty}\left(\left[M_n^{'}(t)\right]^2 - 2tM_n^{'}(t)M_n(t)+t^2\left[M_n(t)\right]^2\right)exp\left(-\beta t^2\right)dt \\ &= n\int_{-\infty}^{\infty}\left[M_n^{'}(t)\right]^2 exp\left(-\beta t^2\right)dt - 2n\int_{-\infty}^{\infty} tM_n^{'}(t)M_n(t)exp\left(-\beta t^2\right)dt + n\int_{-\infty}^{\infty} t^2\left[M_n(t)\right]^2 exp\left(-\beta t^2\right)dt \\ &= nI_1 - 2nI_2 + nI_3. \end{aligned} \tag{16}$$

Below we consider I_1, I_2 and I_3 respectively.

$$\begin{aligned} I_1 &= \int_{-\infty}^{\infty}\left[M_n^{'}(t)\right]^2 exp\left(-\beta t^2\right)dt \\ &= \frac{1}{n^2}\int_{-\infty}^{\infty}\left[\sum_{j=1}^{n} Y_j exp\left(tY_j\right)\right]^2 exp\left(-\beta t^2\right)dt \\ &= \frac{1}{n^2}\int_{-\infty}^{\infty}\sum_{j=1}^{n}\sum_{k=1}^{n} Y_jY_k exp\left(t\left(Y_j+Y_k\right)\right)exp\left(-\beta t^2\right)dt \\ &= \frac{1}{n^2}\sum_{j=1}^{n}\sum_{k=1}^{n} Y_jY_k\int_{-\infty}^{\infty} exp\left(t\left(Y_j+Y_k\right)\right)exp\left(-\beta t^2\right)dt \\ &= \frac{1}{n^2}\sum_{j=1}^{n}\sum_{k=1}^{n} Y_jY_k\int_{-\infty}^{\infty}\sqrt{\frac{\pi}{\beta}}exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right)\phi_T(t)\,dt \\ &= \frac{1}{n^2}\sqrt{\frac{\pi}{\beta}}\sum_{j=1}^{n}\sum_{k=1}^{n} Y_jY_k exp\left(\frac{\left(Y_j+Y_k\right)^2}{4\beta}\right), \\ &\quad \textit{see equation } 10. \end{aligned} \tag{17}$$

$$
\begin{aligned}
I_2 &= \int_{-\infty}^{\infty} tM_n^{'}(t)\, M_n(t)\, exp\left(-\beta t^2\right) dt \\
&= \frac{1}{n^2} \int_{-\infty}^{\infty} t \left(\sum_{j=1}^{n} Y_j exp\left(tY_j\right)\right) \left(\sum_{k=1}^{n} exp\left(tY_k\right)\right) exp\left(-\beta t^2\right) dt \\
&= \frac{1}{n^2} \int_{-\infty}^{\infty} t \sum_{j=1}^{n}\sum_{k=1}^{n} Y_j exp\left(t\left(Y_j + Y_k\right)\right) exp\left(-\beta t^2\right) dt \\
&= \frac{1}{n^2} \sum_{j=1}^{n}\sum_{k=1}^{n} Y_j \int_{-\infty}^{\infty} t\, exp\left(t\left(Y_j + Y_k\right)\right) exp\left(-\beta t^2\right) dt \\
&= \frac{1}{n^2} \sum_{j=1}^{n}\sum_{k=1}^{n} Y_j \int_{-\infty}^{\infty} t\sqrt{\frac{\pi}{\beta}} exp\left(\frac{\left(Y_j + Y_k\right)^2}{4\beta}\right) \phi_T(t)\, dt \\
&= \frac{1}{n^2} \sum_{j=1}^{n}\sum_{k=1}^{n} Y_j \sqrt{\frac{\pi}{\beta}} exp\left(\frac{\left(Y_j + Y_k\right)^2}{4\beta}\right) \int_{-\infty}^{\infty} t\phi_T(t)\, dt \\
&= \frac{1}{n^2} \sqrt{\frac{\pi}{\beta}} \sum_{j=1}^{n}\sum_{k=1}^{n} Y_j exp\left(\frac{\left(Y_j + Y_k\right)^2}{4\beta}\right) E[T] \\
&= \frac{1}{n^2} \sqrt{\frac{\pi}{\beta}} \sum_{j=1}^{n}\sum_{k=1}^{n} Y_j exp\left(\frac{\left(Y_j + Y_k\right)^2}{4\beta}\right) \frac{Y_j + Y_k}{2\beta} \\
&= \frac{1}{n^2} \sqrt{\frac{\pi}{\beta}} \sum_{j=1}^{n}\sum_{k=1}^{n} Y_j \frac{Y_j + Y_k}{2\beta} exp\left(\frac{\left(Y_j + Y_k\right)^2}{4\beta}\right), \qquad (18)
\end{aligned}
$$

see equation 10.

$$
\begin{aligned}
I_3 &= \int_{-\infty}^{\infty} t^2 \left[M_n(t)\right]^2 exp\left(-\beta t^2\right) dt \\
&= \frac{1}{n^2} \int_{-\infty}^{\infty} t^2 \left[\sum_{j=1}^{n} exp\left(tY_j\right)\right]^2 exp\left(-\beta t^2\right) dt \\
&= \frac{1}{n^2} \int_{-\infty}^{\infty} t^2 \sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(t\left(Y_j + Y_k\right)\right) exp\left(-\beta t^2\right) dt \\
&= \frac{1}{n^2} \sum_{j=1}^{n}\sum_{k=1}^{n} \int_{-\infty}^{\infty} t^2 exp\left(t\left(Y_j + Y_k\right)\right) exp\left(-\beta t^2\right) dt \\
&= \frac{1}{n^2} \sum_{j=1}^{n}\sum_{k=1}^{n} \int_{-\infty}^{\infty} t^2 \sqrt{\frac{\pi}{\beta}} exp\left(\frac{\left(Y_j + Y_k\right)^2}{4\beta}\right) \phi_T(t)\, dt
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{n^2}\sqrt{\frac{\pi}{\beta}}\sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right)\int_{-\infty}^{\infty} t^2\phi_T(t)\,dt \\
&= \frac{1}{n^2}\sqrt{\frac{\pi}{\beta}}\sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right)E\left[T^2\right] \\
&= \frac{1}{n^2}\sqrt{\frac{\pi}{\beta}}\sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right)\left(Var\left[T\right]+\left(E\left[T\right]\right)^2\right) \\
&= \frac{1}{n^2}\sqrt{\frac{\pi}{\beta}}\sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right)\left(\frac{(Y_j+Y_k)^2}{4\beta^2}+\frac{1}{2\beta}\right), \qquad (19)\\
&\quad \textit{see equation } 10.
\end{aligned}
$$

Substituting (17), (18) and (19) into (16), we obtain

$$
\begin{aligned}
T_{n,\beta}^{(2)} &= nI_1 - 2nI_2 + nI_3 \\
&= \frac{1}{n}\sqrt{\frac{\pi}{\beta}}\sum_{j=1}^{n}\sum_{k=1}^{n} Y_jY_k exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right) - \frac{2}{n}\sqrt{\frac{\pi}{\beta}}\sum_{j=1}^{n}\sum_{k=1}^{n} Y_j\left(\frac{Y_j+Y_k}{2\beta}\right)exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right) \\
&\quad +\frac{1}{n}\sqrt{\frac{\pi}{\beta}}\sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right)\left(\frac{(Y_j+Y_k)^2}{4\beta^2}+\frac{1}{2\beta}\right) \\
&= \sqrt{\frac{\pi}{\beta}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\left\{Y_jY_k exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right) - Y_j\left(\frac{Y_j+Y_k}{\beta}\right)exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right)\right. \\
&\quad \left. +exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right)\left(\frac{(Y_j+Y_k)^2}{4\beta^2}+\frac{1}{2\beta}\right)\right\} \\
&= \sqrt{\frac{\pi}{\beta}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right)\left\{Y_jY_k - \frac{Y_j\left(Y_j+Y_k\right)}{\beta} + \frac{(Y_j+Y_k)^2}{4\beta^2} + \frac{1}{2\beta}\right\} \\
&= \sqrt{\frac{\pi}{\beta}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right)\left\{Y_jY_k - \frac{2Y_j\left(Y_j+Y_k\right)-1}{2\beta} + \frac{(Y_j+Y_k)^2}{4\beta^2}\right\} \\
&= \sqrt{\frac{\pi}{\beta}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{(Y_j+Y_k)^2}{4\beta}\right)\left\{Y_jY_k + \frac{1-2Y_j\left(Y_j+Y_k\right)}{2\beta} + \frac{(Y_j+Y_k)^2}{4\beta^2}\right\} \\
&= \sqrt{\frac{\pi}{\beta}}\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} exp\left(\frac{(Y_i+Y_j)^2}{4\beta}\right)\left\{Y_iY_j + \frac{1-2Y_i\left(Y_i+Y_j\right)}{2\beta} + \frac{(Y_i+Y_j)^2}{4\beta^2}\right\} \\
&= \sqrt{\frac{\pi}{\beta}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{n} exp\left(\frac{Z_{i,j}^2}{4\beta}\right)\left\{Y_iY_j + \frac{1-2Y_iZ_{i,j}}{2\beta} + \frac{Z_{i,j}^2}{4\beta^2}\right\}.
\end{aligned}
$$

$$
T_{n,\beta}^{(2)} = \sqrt{\frac{\pi}{\beta}}\frac{1}{n}\sum_{i,j=1}^{n} exp\left(\frac{Z_{i,j}^2}{4\beta}\right)\left\{Y_iY_j + \frac{1-2Y_iZ_{i,j}}{2\beta} + \frac{Z_{i,j}^2}{4\beta^2}\right\}.
$$

Appendix 3: R-code used in simulation study

The R-code used to calculate powers.

```
#############################################################################################
MCcv    =       100000  #setting the number of MCs for critical values
MCpo    =       100000  #setting the number of MCs for power
n       =       50      #setting the sample size
#############################################################################################

#loading the relevant library
library(nortest)
library(triangle)
library(truncnorm)
library(sn)
library(stargazer)

#defining the first mgf test statistic
MGFT1_5.test <- function(X,beta1){
        n     = length(X)
        Xbar  = mean(X)
        Sn    = sqrt(sum((X-Xbar)^2)/n)
        Y     = (X-Xbar)/Sn
  Tmat1 = matrix(NA,n,n)

for (j in 1:n){
        for (k in 1:n){
                zjk         = Y[j]+Y[k]
                Tmat1[j,k] = exp(zjk^2/(4*beta1))
                }
        }
Tmat2 = matrix(NA,n,1)
for (j in 1:n){
        Tmat2[j] = exp((Y[j]^2)/(4*beta1-2))
        }

T1 = sum(Tmat1)/(n*sqrt(beta1))
T2 = 2/sqrt(beta1-0.5)*sum(Tmat2)
```

```
T3 = n/sqrt(beta1-1)
T.n.beta1_5 = sqrt(pi)*(T1-T2+T3)
return(T.n.beta1_5)
}

MGFT1_10.test <- function(X,beta2){
        n     = length(X)
        Xbar  = mean(X)
        Sn    = sqrt(sum((X-Xbar)^2)/n)
        Y     = (X-Xbar)/Sn
  Tmat1 = matrix(NA,n,n)

for (j in 1:n){
        for (k in 1:n){
                zjk          = Y[j]+Y[k]
                Tmat1[j,k] = exp(zjk^2/(4*beta2))
                }
        }
Tmat2 = matrix(NA,n,1)
for (j in 1:n){
        Tmat2[j] = exp((Y[j]^2)/(4*beta2-2))
        }

T1 = sum(Tmat1)/(n*sqrt(beta2))
T2 = 2/sqrt(beta2-0.5)*sum(Tmat2)
T3 = n/sqrt(beta2-1)
T.n.beta1_10 = sqrt(pi)*(T1-T2+T3)
return(T.n.beta1_10)
}

#defining the second mgf test statistic
MGFT2_5.test <- function(X,beta1){
        n    = length(X)    Xbar = mean(X)
        Sn   = sqrt(sum((X-Xbar)^2)/n)
```

```
        Y     = (X–Xbar)/Sn
        Tmat  = matrix(NA,n,n)
for (j in 1:n){
for (k in 1:n){
Zjk   = Y[j]+Y[k]
T1    = exp(Zjk^2/(4*beta1))
T2    = Y[j]*Y[k] + Zjk^2/(4*beta1^2)*(1-2*beta1)+ 1/(2*beta1)
Tmat[j,k] = T1*T2
}
}
T.n.beta2_5 = sqrt(pi/beta1)/n*sum(Tmat)
return(T.n.beta2_5)
}

MGFT2_10.test <- function(X,beta2){
        n     = length(X)
        Xbar  = mean(X)
        Sn    = sqrt(sum((X–Xbar)^2)/n)
        Y     = (X–Xbar)/Sn
        Tmat  = matrix(NA,n,n)
for (j in 1:n){
for (k in 1:n){
Zjk   = Y[j]+Y[k]
T1    = exp(Zjk^2/(4*beta2))
T2    = Y[j]*Y[k] + Zjk^2/(4*beta2^2)*(1-2*beta2) + 1/(2*beta2)
Tmat[j,k] = T1*T2
}
}
T.n.beta2_10 = sqrt(pi/beta2)/n*sum(Tmat)
return(T.n.beta2_10)
}

SimFromDist <- function(n,DistNum){
        if (DistNum==0){
```

```
        X = rnorm(n)
    }
    if (DistNum==1){
        X = runif(n)
    }
    if (DistNum==2){
        X = rtriangle(n)
    }
    if (DistNum==3){
        X =  rtruncnorm(n,-2,2)
    }
    if (DistNum==4){
        X =   rcauchy(n)
    }
    if (DistNum==5){
        X =   rt(n,3)
    }
    if (DistNum==6){
        X =   rt(n,5)
    }
    if (DistNum==7){
        X =     rlnorm(n)
    }
    if (DistNum==8){
        X =     rsn(n,xi=0, omega=1, alpha=5, tau=0)[1:n]
    }
    if (DistNum==9){
        X =     rweibull(n,5,1)
    }
    if (DistNum==10){
        X =     rweibull(n,10,1)
    }
    return(X)
}
```

```
#defining MC setup
set.seed(1234)          #setting random seed
MC=MCcv                 #setting the number of MC replications
alpha=0.05              #setting the nominal significance level
beta1=5                 #setting the tuning parameter for the mgf tests
beta2=10

#initialising test statitsics
ks.t = numeric(MC)
cm.t = numeric(MC)
ad.t = numeric(MC)
m1_5.t = numeric(MC)
m2_5.t = numeric(MC)
m1_10.t = numeric(MC)
m2_10.t = numeric(MC)

ptm<-proc.time() ##########

#calculating test statistics
pb=winProgressBar(title="Critical values,n=50",label="0% done")
for (j in 1:MC)
        {
        data<-rnorm(n)
        data<-(data-mean(data))/sqrt(var(data))
        ks<-ks.test(data,"pnorm")
        ks.t[j]<-ks$statistic
        cm<-cvm.test(data)
        cm.t[j]<-cm$statistic
        ad<-ad.test(data)
        ad.t[j]<-ad$statistic
        m1_5.t[j]<-MGFT1_5.test(data,beta1)
        m2_5.t[j]<-MGFT2_5.test(data,beta1)
        m1_10.t[j]<-MGFT1_10.test(data,beta2)
```

```
        m2_10.t[j]<-MGFT2_10.test(data,beta2)
        info=sprintf("%d%% done",floor((j/MC*100)))
        setWinProgressBar(pb,j/MC,label=info)
        }
close(pb)

tmrcv<-(proc.time()-ptm)[3] ##########

#sorting test statistics
ks.t = sort(ks.t)
cm.t = sort(cm.t)
ad.t = sort(ad.t)
m1_5.t = sort(m1_5.t)
m2_5.t = sort(m2_5.t)
m1_10.t = sort(m1_10.t)
m2_10.t = sort(m2_10.t)

#calculating critical values
indx  = floor(MC*(1-alpha))
ks.cv = ks.t[indx]
cm.cv = cm.t[indx]
ad.cv = ad.t[indx]
m1_5.cv = m1_5.t[indx]
m2_5.cv = m2_5.t[indx]
m1_10.cv = m1_10.t[indx]
m2_10.cv = m2_10.t[indx]

ks.power = numeric(11)
cm.power = numeric(11)
ad.power = numeric(11)
m1_5.power = numeric(11)
m2_5.power = numeric(11)
m1_10.power = numeric(11)
m2_10.power = numeric(11)
```

```
MC=MCpo

ptm<-proc.time() ##########

pb=winProgressBar(title="Powers,n=50",label="0% done")

for (k in 1:11){

        DistNum = k-1

#initialising test statistics
ks.t = numeric(MC)
cm.t = numeric(MC)
ad.t = numeric(MC)
m1_5.t = numeric(MC)
m2_5.t = numeric(MC)
m1_10.t = numeric(MC)
m2_10.t = numeric(MC)

#calculating test statistics
for (j in 1:MC)
        {
        data<-SimFromDist(n,DistNum)
        data<-(data-mean(data))/sqrt(var(data))
        ks<-ks.test(data,"pnorm")
        ks.t[j]<-(ks$statistic>ks.cv)
        cm<-cvm.test(data)
        cm.t[j]<-(cm$statistic>cm.cv)
        ad<-ad.test(data)
        ad.t[j]<-(ad$statistic>ad.cv)
        m1_5<-MGFT1_5.test(data,beta1)
        m1_5.t[j]<-(m1_5>m1_5.cv)
        m2_5<-MGFT2_5.test(data,beta1)
```

```
        m2_5.t[j]<-(m2_5>m2_5.cv)
        m1_10<-MGFT1_10.test(data,beta2)
        m1_10.t[j]<-(m1_10>m1_10.cv)
        m2_10<-MGFT2_10.test(data,beta2)
        m2_10.t[j]<-(m2_10>m2_10.cv)
        }
#calculating powers
ks.power[k] = round(mean(ks.t)*100,0)
cm.power[k] = round(mean(cm.t)*100,0)
ad.power[k] = round(mean(ad.t)*100,0)
m1_5.power[k] = round(mean(m1_5.t)*100,0)
m2_5.power[k] = round(mean(m2_5.t)*100,0)
m1_10.power[k] = round(mean(m1_10.t)*100,0)
m2_10.power[k] = round(mean(m2_10.t)*100,0)

info=sprintf("%d%% done",floor((k/11*100)))
setWinProgressBar(pb,k/11,label=info)
        }
close(pb)

tmrpo<-(proc.time()-ptm[3] ##########

#displaying powers
ks.power
cm.power
ad.power
m1_5.power
m1_10.power
m2_5.power
m2_10.power

#table
Results = matrix(NA,11,7)
Results[,1] = ks.power
```

```
Results[,2] = cm.power
Results[,3] = ad.power
Results[,4] = m1_5.power
Results[,5] = m1_10.power
Results[,6] = m2_5.power
Results[,7] = m2_10.power
stargazer(Results)
save.image("n50Results.RData")
```

The R-code used to calculate graphs for skewed normal distribution.

```
DistNum = 8
##############################################################################
MCcv   = 100000              #setting the number of MCs for critical values
MCpo   = 10000               #setting the number of MCs for powers
n_grid = seq(10,100,10)      #setting the sample size
set.seed(1234)
##############################################################################

#loading the relevant library
library(nortest)
library(triangle)
library(truncnorm)
library(sn)
library(stargazer)

#defining the first mgf test statistic
MGFT1.test <- function(X, beta){
        n     = length(X)
        Xbar  = mean(X)
        Sn    = sqrt(sum((X-Xbar)^2)/n)
        Y     = (X-Xbar)/Sn
Tmat1 = matrix(NA,n,n)
        for (j in 1:n){
                for (k in 1:n){
                        zjk          = Y[j]+Y[k]
```

```
                    Tmat1[j,k] = exp(zjk^2/(4*beta))
                }
        }
 Tmat2 = matrix(NA,n,1)
        for (j in 1:n){
                Tmat2[j] = exp((Y[j]^2)/(4*beta-2))
                }
T1 = sum(Tmat1)/(n*sqrt(beta))
T2 = 2/sqrt(beta-0.5)*sum(Tmat2)
T3 = n/sqrt(beta-1)
T.n.beta = sqrt(pi)*(T1-T2+T3)
return(T.n.beta)
}

#defining the second mgf test statistic
MGFT2.test <- function(X,beta){
        n    = length(X)
        Xbar = mean(X)
        Sn   = sqrt(sum((X-Xbar)^2)/n)
        Y    = (X-Xbar)/Sn
        Tmat = matrix(NA,n,n)
for (j in 1:n){
for (k in 1:n){
Zjk  = Y[j]+Y[k]
T1   = exp(Zjk^2/(4*beta))
T2   = Y[j]*Y[k] + (1-2*Y[j]*Zjk)/(2*beta)+ Zjk^2/(4*beta^2)
Tmat[j,k] = T1*T2
}
}
T.n.beta = sqrt(pi/beta)/n*sum(Tmat)
return(T.n.beta)
}

SimFromDist <- function(n,DistNum){
```

```
    if (DistNum==0){
    X = rnorm(n)
    }
    if (DistNum==1){
    X = runif(n)
    }
    if (DistNum==2){
    X = rtriangle(n)
    }
    if (DistNum==3){
    X = rtruncnorm(n,-2,2)
    }
    if (DistNum==4){
    X = rcauchy(n)
    }
    if (DistNum==5){
    X = rt(n,3)
    }
    if (DistNum==6){
    X = rt(n,5)
    }
    if (DistNum==7){
    X = rlnorm(n)
    }
    if (DistNum==8){
    X = rsn(n, xi=0, omega=1, alpha=5, tau=0)[1:n]
    }
    if (DistNum==9){
    X = rweibull(n,5,1)
    }
    if (DistNum==10){
    X = rweibull(n,10,1)
    }
return(X)
```

```
}

#defining MC setup
set.seed(1234)          #setting random seed
MC    = MCcv            #setting the number of MC replications
alpha = 0.05            #setting the nonminal significance level
beta1 = 5               #setting the tuning parameter for the mgf tests
beta2 = 10

ptm <- proc.time()    ##########

pb = winProgressBar(title="Power graph", label="0% done")

ks.power    = numeric(length(n_grid))
cm.power    = numeric(length(n_grid))
ad.power    = numeric(length(n_grid))
m1_5.power  = numeric(length(n_grid))
m1_10.power = numeric(length(n_grid))
m2_5.power  = numeric(length(n_grid))
m2_10.power = numeric(length(n_grid))

for (k in 1:length(n_grid)){
        n = n_grid[k]

#initialising test statitsics
ks.t    = numeric(MC)
cm.t    = numeric(MC)
ad.t    = numeric(MC)
m1_5.t  = numeric(MC)
m1_10.t = numeric(MC)
m2_5.t  = numeric(MC)
m2_10.t = numeric(MC)

#calculating test statistics
```

```
for (j in 1:MC)            {
        data        <- rnorm(n)
        data        <- (data-mean(data))/sqrt(var(data))
        ks          <- ks.test(data,"pnorm")
        ks.t[j]     <- ks$statistic
        cm          <- cvm.test(data)
        cm.t[j]     <- cm$statistic
        ad          <- ad.test(data)
        ad.t[j]     <- ad$statistic
        m1_5.t[j]   <- MGFT1.test(data,beta1)
        m1_10.t[j]  <- MGFT1.test(data,beta2)
        m2_5.t[j]   <- MGFT2.test(data,beta1)
        m2_10.t[j]  <- MGFT2.test(data,beta2)
}

#sorting test statistics
ks.t    = sort(ks.t)
cm.t    = sort(cm.t)
ad.t    = sort(ad.t)
m1_5.t  = sort(m1_5.t)
m1_10.t = sort(m1_10.t)
m2_5.t  = sort(m2_5.t)
m2_10.t = sort(m2_10.t)

#calculating critical values
indx     = floor(MC*(1-alpha))
ks.cv    = ks.t[indx]
cm.cv    = cm.t[indx]
ad.cv    = ad.t[indx]
m1_5.cv  = m1_5.t[indx]
m1_10.cv = m1_10.t[indx]
m2_5.cv  = m2_5.t[indx]
m2_10.cv = m2_10.t[indx]
MC = MCpo
```

```
#initialising test statistics
ks.t     = numeric(MC)
cm.t     = numeric(MC)
ad.t     = numeric(MC)
m1_5.t   = numeric(MC)
m2_5.t   = numeric(MC)
m1_10.t  = numeric(MC)
m2_10.t  = numeric(MC)

#calculating test statistics
        for (j in 1:MC){
                data        <- SimFromDist(n,DistNum)
                data        <- (data-mean(data))/sqrt(var(data))
                ks          <- ks.test(data,"pnorm")
                ks.t[j]     <- (ks$statistic>ks.cv)
                cm          <- cvm.test(data)
                cm.t[j]     <- (cm$statistic>cm.cv)
                ad          <- ad.test(data)
                ad.t[j]     <- (ad$statistic>ad.cv)
                m1_5        <- MGFT1.test(data,beta1)
                m1_5.t[j]   <- (m1_5>m1_5.cv)
                m1_10       <- MGFT1.test(data,beta2)
                m1_10.t[j]  <- (m1_10>m1_10.cv)
                m2_5        <- MGFT2.test(data,beta1)
                m2_5.t[j]   <- (m2_5>m2_5.cv)
                m2_10       <- MGFT2.test(data,beta2)
                m2_10.t[j]  <- (m2_10>m2_10.cv)
        }

#calculating powers
ks.power[k]     = round(mean(ks.t)*100,0)
cm.power[k]     = round(mean(cm.t)*100,0)
ad.power[k]     = round(mean(ad.t)*100,0)
```

```
m1_5.power[k]  = round(mean(m1_5.t)*100,0)
m1_10.power[k] = round(mean(m1_10.t)*100,0)
m2_5.power[k]  = round(mean(m2_5.t)*100,0)
m2_10.power[k] = round(mean(m2_10.t)*100,0)

info <- sprintf("%d%% done", floor((k/length(n_grid)*100)))
setWinProgressBar(pb, k/length(n_grid), label=info) }
close(pb)

tmr <- (proc.time()-ptm)[3]     ##########

jpeg('EDFplotSN.jpg')
plot(n_grid,ks.power,type="l",ylim=c(0,100),col="red",xlab="Sample
size",ylab="Power in percentage",main="Powers of EDF tests")
lines(n_grid,cm.power,type="l",col="blue")
lines(n_grid,ad.power,type="l")
dev.off()

jpeg('EMGFplotSN.jpg')
plot(n_grid,m1_5.power,type="l",ylim=c(0,100),col="red",xlab="Sample
size",ylab="Power in percentage",main="Powers of EMGF tests")
lines(n_grid,m1_10.power,type="l",col="blue")
lines(n_grid,m2_5.power,type="l",col="green")
lines(n_grid,m2_10.power,type="l")
dev.off()
```

The R-code used to calculate graphs for t(3) distribution.

```
 DistNum = 5
##############################################################################
MCcv   = 100000            #setting the number of MCs for critical values
MCpo   = 10000             #setting the number of MCs for powers
n_grid = seq(10,100,10)    #setting the sample size
set.seed(1234)
##############################################################################
```

```
#loading the relevant library
library(nortest)
library(triangle)
library(truncnorm)
library(sn)
library(stargazer)

#defining the first mgf test statistic
MGFT1.test <- function(X,beta){
        n     = length(X)
        Xbar  = mean(X)
        Sn    = sqrt(sum((X-Xbar)^2)/n)
        Y     = (X-Xbar)/Sn
Tmat1 = matrix(NA,n,n)
        for (j in 1:n){
                for (k in 1:n){
                        zjk         = Y[j]+Y[k]
                        Tmat1[j,k] = exp(zjk^2/(4*beta))
                }
        }
 Tmat2 = matrix(NA,n,1)
        for (j in 1:n){
                Tmat2[j] = exp((Y[j]^2)/(4*beta-2))
                }
T1 = sum(Tmat1)/(n*sqrt(beta))
T2 = 2/sqrt(beta-0.5)*sum(Tmat2)
T3 = n/sqrt(beta-1)
T.n.beta = sqrt(pi)*(T1-T2+T3)
return(T.n.beta)
}

#defining the second mgf test statistic
MGFT2.test <- function(X,beta){
        n    = length(X)
```

```
        Xbar = mean(X)
        Sn   = sqrt(sum((X-Xbar)^2)/n)
        Y    = (X-Xbar)/Sn
        Tmat = matrix(NA,n,n)
for (j in 1:n){
for (k in 1:n){
Zjk  = Y[j]+Y[k]
T1   = exp(Zjk^2/(4*beta))
T2   = Y[j]*Y[k] + (1-2*Y[j]*Zjk)/(2*beta)+ Zjk^2/(4*beta^2)
Tmat[j,k] = T1*T2
}
}
T.n.beta = sqrt(pi/beta)/n*sum(Tmat)
return(T.n.beta)
}

SimFromDist <- function(n,DistNum){
        if (DistNum==0){
        X = rnorm(n)
        }
        if (DistNum==1){
        X = runif(n)
        }
        if (DistNum==2){
        X = rtriangle(n)
        }
        if (DistNum==3){
        X = rtruncnorm(n,-2,2)
        }
        if (DistNum==4){
        X = rcauchy(n)
        }
        if (DistNum==5){
        X = rt(n,3)
```

```
        }
        if (DistNum==6){
        X = rt(n,5)
        }
        if (DistNum==7){
        X = rlnorm(n)
        }
        if (DistNum==8){
        X = rsn(n, xi=0, omega=1, alpha=5, tau=0)[1:n]
        }
        if (DistNum==9){
        X = rweibull(n,5,1)
        }
        if (DistNum==10){
        X = rweibull(n,10,1)
        }
return(X)
}

#defining MC setup
set.seed(1234)          #setting random seed
MC    = MCcv            #setting the number of MC replications
alpha = 0.05            #setting the nonminal significance level
beta1 = 5               #setting the tuning parameter for the mgf tests
beta2 = 10

ptm <- proc.time()    ##########

pb = winProgressBar(title="Power graph", label="0% done")

ks.power    = numeric(length(n_grid))
cm.power    = numeric(length(n_grid))
ad.power    = numeric(length(n_grid))
m1_5.power  = numeric(length(n_grid))
```

```
m1_10.power = numeric(length(n_grid))
m2_5.power  = numeric(length(n_grid))
m2_10.power = numeric(length(n_grid))

for (k in 1:length(n_grid)){
        n = n_grid[k]

#initialising test statitsics
ks.t    = numeric(MC)
cm.t    = numeric(MC)
ad.t    = numeric(MC)
m1_5.t  = numeric(MC)
m1_10.t = numeric(MC)
m2_5.t  = numeric(MC)
m2_10.t = numeric(MC)

#calculating test statistics
for (j in 1:MC)           {
        data       <- rnorm(n)
        data       <- (data-mean(data))/sqrt(var(data))
        ks         <- ks.test(data,"pnorm")
        ks.t[j]    <- ks$statistic
        cm         <- cvm.test(data)
        cm.t[j]    <- cm$statistic
        ad         <- ad.test(data)
        ad.t[j]    <- ad$statistic
        m1_5.t[j]  <- MGFT1.test(data,beta1)
        m1_10.t[j] <- MGFT1.test(data,beta2)
        m2_5.t[j]  <- MGFT2.test(data,beta1)
        m2_10.t[j] <- MGFT2.test(data,beta2)
}

#sorting test statistics
ks.t    = sort(ks.t)
```

```
cm.t     = sort(cm.t)
ad.t     = sort(ad.t)
m1_5.t   = sort(m1_5.t)
m1_10.t  = sort(m1_10.t)
m2_5.t   = sort(m2_5.t)
m2_10.t  = sort(m2_10.t)

#calculating critical values
indx     = floor(MC*(1-alpha))
ks.cv    = ks.t[indx]
cm.cv    = cm.t[indx]
ad.cv    = ad.t[indx]
m1_5.cv  = m1_5.t[indx]
m1_10.cv = m1_10.t[indx]
m2_5.cv  = m2_5.t[indx]
m2_10.cv = m2_10.t[indx]
MC = MCpo

#initialising test statistics
ks.t     = numeric(MC)
cm.t     = numeric(MC)
ad.t     = numeric(MC)
m1_5.t   = numeric(MC)
m2_5.t   = numeric(MC)
m1_10.t  = numeric(MC)
m2_10.t  = numeric(MC)

#calculating test statistics
        for (j in 1:MC){
                data        <- SimFromDist(n,DistNum)
                data        <- (data-mean(data))/sqrt(var(data))
                ks          <- ks.test(data,"pnorm")
                ks.t[j]     <- (ks$statistic>ks.cv)
                cm          <- cvm.test(data)
```

```
                cm.t[j]     <- (cm$statistic>cm.cv)
                ad          <- ad.test(data)
                ad.t[j]     <- (ad$statistic>ad.cv)
                m1_5        <- MGFT1.test(data,beta1)
                m1_5.t[j]   <- (m1_5>m1_5.cv)
                m1_10       <- MGFT1.test(data,beta2)
                m1_10.t[j]  <- (m1_10>m1_10.cv)
                m2_5        <- MGFT2.test(data,beta1)
                m2_5.t[j]   <- (m2_5>m2_5.cv)
                m2_10       <- MGFT2.test(data,beta2)
                m2_10.t[j]  <- (m2_10>m2_10.cv)
        }

#calculating powers
ks.power[k]     = round(mean(ks.t)*100,0)
cm.power[k]     = round(mean(cm.t)*100,0)
ad.power[k]     = round(mean(ad.t)*100,0)
m1_5.power[k]   = round(mean(m1_5.t)*100,0)
m1_10.power[k]  = round(mean(m1_10.t)*100,0)
m2_5.power[k]   = round(mean(m2_5.t)*100,0)
m2_10.power[k]  = round(mean(m2_10.t)*100,0)

info <- sprintf("%d%% done", floor((k/length(n_grid)*100)))
setWinProgressBar(pb, k/length(n_grid), label=info) }
close(pb)

tmr <- (proc.time()-ptm)[3]     ##########

jpeg('EDFplotT3.jpg')
plot(n_grid,ks.power,type="l",ylim=c(0,100),col="red",xlab="Sample
size",ylab="Power in percentage",main="Powers of EDF tests")
lines(n_grid,cm.power,type="l",col="blue")
lines(n_grid,ad.power,type="l")
dev.off()
```

```
jpeg('EMGFplotT3.jpg')
plot(n_grid,m1_5.power,type="l",ylim=c(0,100),col="red",xlab="Sample
size",ylab="Power in percentage",main="Powers of EMGF tests")
lines(n_grid,m1_10.power,type="l",col="blue")
lines(n_grid,m2_5.power,type="l",col="green")
lines(n_grid,m2_10.power,type="l")
dev.off()
```

Appendix 4: SAS-code used in practical application

```
data Appels;
input Amount @@;
label Amount='Average fruit weight (grams) of apples per tree.';
datalines;
85.3 86.9 96.8 108.5 113.8 87.7 94.5 99.9 92.9 67.3
90.6 129.8 48.9 117.5 100.8 94.5 94.4 98.9 96 99.4
;
title 'Average fruit weight (grams) of apples
          per tree for 20 trees in an agricultural experiment.';

*MGF1 test;

proc iml;
print "MGF tests";
use Appels;
read all into x;
        n    = nrow(x);
        pi   = constant("pi");
        beta = 10;
        xbar = x[:];
        Sn   = sqrt(ssq(x - xbar)/n);
        Y    = (x-xbar)/Sn;
        Tmat1 = J(n,n,.);
```

```
do j=1 to n;
        do k=1 to n;
                Zjk=Y[k]+Y[j];
                Tmat1[j,k]=exp(Zjk**2/(4#beta));
                end;
        end;
Tmat2 = J(n,1,.);
        do j=1 to n;
                Tmat2[j] = exp((Y[j]**2)/(4*beta-2));
        end;
T1 = sum(Tmat1)/(n*sqrt(beta));
T2 = 2/sqrt(beta-0.5)*sum(Tmat2);
T3 = n/sqrt(beta-1);
Tn_beta = sqrt(pi)*(T1-T2+T3);
print Tn_beta;
quit;

*MGF2 test;

proc iml;
print "MGF tests";
use Appels;
read all into x;
n    = nrow(x);
pi   = constant("pi");
beta = 10;
xbar = x[:];
Sn   = sqrt(ssq(x - xbar)/n);
Y    = (x-xbar)/Sn;
newtmat = J(n,1,0);
        do ii=1 to n;
                        Tmatij2 = J(n,1,0);
        do jj=1 to n;
        Zij2=Y[ii]+Y[jj];
```

```
        T1_1=exp((Zij2##2)/(4#beta));
        T2_2=(Y[ii]#Y[jj])+((Zij2##2)/(4#(beta##2)))+ ((1 - 2#Y[ii]#Zij2)/(2#beta));
        Tmatij2[jj]=T1_1#T2_2;
        end;
newtmat[ii]=Tmatij2[+];
end;
Tn_beta2=sqrt((pi)/beta)#(1/n)#sum(newtmat);
print Tn_beta2;
quit;
proc univariate data=Appels;
var Amount;
histogram / normal
        vaxis    = axis1
        name     = 'MyHist';
        inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
        / pos = ne  header = 'Summary Statistics';
axis1 label=(a=90 r=0);
run;
```

Gaussian processes applied to class-imbalanced datasets

David John Rosevear 13084667

STK795 Research Report

Submitted in partial fulfilment of the degree BCom(Hons) Statistics

Supervisor: Dr A de Waal

Department of Statistics, University of Pretoria

30 October 2017

Abstract

Modelling class-imbalanced data is problematic. On such data, classifiers tend to misclassify minority class observations. Considering the potential practical use of a classifier that is especially robust to class imbalance, the performance of a Gaussian process classifier is evaluated to determine the degree to which it addresses the problem. GP classification is compared to support-vector machine, random forest and logistic regression classification on three synthetic datasets. The results show that under class imbalance, Gaussian process classification does indeed perform well relative to the other techniques.

At the time of writing, a short version of this paper with the same title is in revision for the Annual Proceedings of the South African Statistical Association Conference 2017.

Declaration

I, *David John Rosevear,* declare that this essay, submitted in partial fulfilment of the degree *BCom(Hons) Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

David John Rosevear

Dr A de Waal

26 October 2017

Date

Acknowledgements

Thank you to my parents, Tracey and David Rosevear, for enabling and always encouraging me to advance myself.

To my supervisor, Dr de Waal, thank you for guiding me towards and challenging me with this exceptionally relevant field of research.

Finally, I gratefully acknowledge the financial assistance provided by the Centre for Artificial Intelligence Research (CAIR), Meraka Institute, CSIR.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

In binary classification, class imbalance presents itself in datasets where there is a disproportionate number of observations belonging to one class relative to the other class. A model trained on such data tends to misinterpret the behaviour of minority class observations, causing them to be misclassified [3].

An example of this is in credit scoring, where a certain loan applicant's creditworthiness is assessed on whether applicants with similar attributes defaulted in the past. A lender would historically have favoured those who are less likely to default, and so only a tiny proportion of the corresponding dataset would comprise defaulters. Of course, erroneously classifying defaulters as non-defaulters would prove disastrous for any lending institution. Thus, investigating methods to create robust predictive models in the presence of class imbalance is entirely applicable.

Comparisons are made between the classification performance of Gaussian processes (GPs), logistic regression (LR), support-vector machines (SVMs), and random forests (RFs) on three synthetic datasets in which class imbalance is present (and upon which the imbalance is compounded by removing observations of the minority class).

The theoretical understanding and application of GPs is the focus of this research report.

1.2 Related work

In order to investigate the class imbalance problem, de Waal et al. [3] evaluate the performance of two-class non-parametric kernel density estimation classifiers using either class imbalance or the Bernoulli distribution as a prior. Their findings show that using class imbalance as a prior decreases classification performance, whereas the use of the Bernoulli prior increases classification performance.

Seidu [13] compares the performance of LR to a Gaussian classifier in creating a model which predicts whether a company will go bankrupt based on its financial ratios. GP covariance functions are also compared to one another in classification. It is concluded that GP classification is superior to LR (partly because it is non-parametric) and that the squared-exponential is the most suitable covariance function for classification in this application.

Then, Brown and Mues [2] compare the classifying ability of RFs, SVMs and gradient boosting to neural networks, decision trees and LR. This is done in the context of credit scoring. They conclude that gradient boosting and RFs are the best classification techniques in this class imbalance setting.

Finally, alternative data-orientated sampling techniques are investigated by Drummond and Holte [4]. They use cost curves to compare the extent to which under-sampling the majority class and over-sampling the minority class solves the class imbalance problem. They conclude that the under-sampling is most

effective, whereas the over-sampling yields almost no improvement.

2 Theory of Gaussian processes

2.1 Introduction

GPs are used to perform classification and regression [10], such as in speech or handwriting recognition. In this machine learning context, they form a supervised learning technique. GPs are also used more generally for spatial analysis (then referred to as *kriging*) in geostatistics [12]. Such applications include modelling geological and meteorological patterns [12]. Although GPs may be applied to both regression and classification, the focus of this paper is on classification.

2.2 Definition

A GP is a set of random variables $\{X_t : t \epsilon S\}$ such that any finite subset of which $\{X_{t_1}, \ldots, X_{t_n}\}$ are jointly multivariate Gaussian distributed [12].

The elements of the random vector $\{X_{t_1}, \ldots, X_{t_n}\}$ introduced above are referred to as the Gaussian process' *finite dimensional distributions*. They are a collection of multivariate Gaussian random variables, which will be elaborated upon in this section.

The Gaussian distribution best describes the unknown frequency of residual errors [7], and so its use is entirely appropriate in probabilistic modelling. This follows from the central limit theorem, which states that the sums of independent and identically distributed random variables follow an approximately Gaussian distribution [10].

The probability density function (PDF) of the univariate Gaussian distribution is given by

$$\mathcal{N}(x|\mu, \sigma^2) \quad \triangleq \quad \frac{1}{\sqrt{2\pi\sigma^2}} \; e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \tag{1}$$

where $\mu = \mathbb{E}[X]$ is the mean and $\sigma^2 = var[X]$ is the variance [10].

Figure 1 indicates a univariate Gaussian distribution with $\mu = 0$ and $\sigma^2 = 1$.

Extending the univariate Gaussian to D dimensions, the PDF of the multivariate Gaussian distribution is given by

$$\mathcal{N}(x|\mu, \Sigma) \quad \triangleq \quad \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \; exp\,[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)] \tag{2}$$

where $\mu = \mathbb{E}[x]\epsilon\mathbb{R}^D$ is the mean vector and $\Sigma = cov[x]$ is the D x D covariance matrix [10].

A bivariate Gaussian distribution with $\mu_x = 0$, $\mu_y = 0$, $\sigma_x^2 = 5$ and $\sigma_y^2 = 20$ is illustrated in Figure 2.

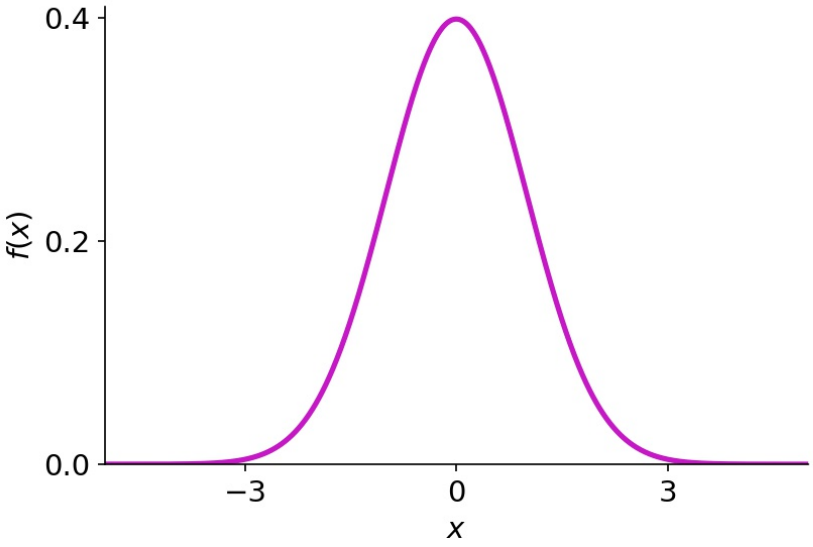


Figure 1: Univariate Gaussian distribution

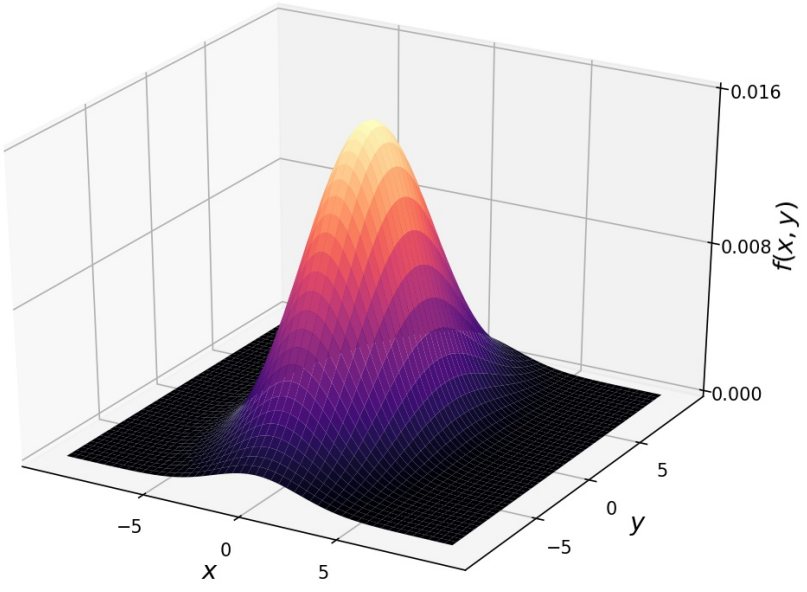


Figure 2: Bivariate Gaussian distribution

2.3 Bayesian preliminaries

Bayesian inference forms the basis of predictive modelling in GP regression and classification, and so the basic concepts underlying the Bayesian approach are discussed in this subsection. Lee [8] is used as a guide.

For two events A and B, Bayes' theorem states that

$$P(A|B) \;=\; \frac{P(B|A)\;P(A)}{P(B)}.$$

It follows from this that

$$P(A|B) \;\propto\; P(B|A)\cdot P(A), \tag{3}$$

and so the conditional probability of A given B is proportional to the conditional probability of B given A multiplied by the probability of A.

Now, in the context of statistical learning, let β be some parameter or attribute of a predictive model and let Y be observed data. Then, from (3), we can write

$$p(\beta|Y) \;\propto\; f(Y|\beta)\cdot\pi(\beta). \tag{4}$$

In this expression, (4):

- $\pi(\beta)$ is the prior distribution. This represents all that is known about β before any data has been observed.
- $f(Y|\beta)$ is the likelihood function. This describes how likely the data points Y are given that β is true.
- $p(\beta|Y)$ is the posterior distribution. This represents all that is known about β once the model has been trained on the data Y.

Therefore,

$$\textit{posterior} \;\propto\; \textit{likelihood}\cdot\textit{prior}.$$

As more data is observed, the posterior distribution is updated and a truer distribution of β emerges.

Finally, there is the posterior predictive distribution. This distribution is formed by integrating the unknown variable (β as above) out of the posterior distribution. The result is a distribution which can be directly used to predict new data points based on the already observed data.

2.4 Regression

Before introducing GPs for classification, GPs for regression is presented here only for the sake of explaining the theory leading up to the classification case. The classification problem is an extension of the regression problem, just as LR is an extension of OLS regression.

2.4.1 Overview

This section is based on Murphy [10] Section 15.1 and 15.2, and Rasmussen and Williams [12] Section 2.2.

Suppose we have some continuous space, in which there are an infinite number of points. Each of these points is linked to a Gaussian random variable. Then, any finite subset of this infinite set of points is called a GP. Furthermore, the degree to which each point depends on every other point is defined by the covariance function[1]. In other words, the covariance function specifies how similar two points, say x and x', are in this space.

Now, to reiterate, let x and x' be two arbitrary points in some space. Every point in this space—and there are an infinite number of points—is linked to a Gaussian random variable. Any finite subset of these random variables is jointly multivariate Gaussian distributed. And so, the assumption made throughout is that the distribution of the GP is the joint distribution of $p(f(x_1), \ldots, f(x_N))$ where $x_1 \ldots x_N$ is an arbitrary finite selection of infinitely many points. Each one of these individual points is linked to its own Gaussian random variable. Thus $p(f(x_1), \ldots, f(x_N))$ is multivariate Gaussian distributed.

GP regression is especially versatile in that it is able to learn patterns of data which do not necessarily conform to a particular shape. This stems from the fact that GP regression is non-parametric. Each $f(x)$ above is in fact a parameter, but the model is able to implement as many parameters as it likes to best describe the observed data—still considering model generality, of course. Furthermore, the choice of available parameters is infinite. This is confirmed when we consider that the parameters $f(x_1), \ldots, f(x_N)$ are a finite selection of an infinite number of points in the space, as stated above.

2.4.2 GP regression contrasted with Bayesian linear regression

Now, let us contrast the non-parametric feature of GP regression with the parametric feature of Bayesian linear regression. Using the latter technique, the number of parameters used would have to suit the nature of the data to be modelled. So, the model $Y = \theta_0 + \theta_1 x$ is only appropriate for describing approximately linear data. GP regression, however, is free of this restriction as the number of possible parameters to be used is infinite.

[1] *Covariance function* and *kernel* are equivalent terms, although *covariance function* is generally used in the context of GPs.

Also, in Bayesian linear regression, a primary objective is to determine a distribution that may describe each parameter θ. GP regression differs from this totally in that a distribution is instead created to describe the various *functions* which suit the observed data.

2.4.3 Model formulation

Now, consider the process

$$f(x) \sim \mathcal{GP}(m(x), \kappa(x, x')) \tag{5}$$

in which $m(x)$ is the mean function and $\kappa(x, x')$ is a positive semidefinite covariance function (which defines how similar x and x' are). This is the prior on the regression function. Then,

$$m(x) = \mathbb{E}[f(x)] \tag{6}$$

and

$$\kappa(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))^T]. \tag{7}$$

For the sake of simplicity, it is assumed that $m(x) = 0$.

Ten samples from a GP prior distribution using the Matérn covariance function (see Section 2.5) are illustrated in Figure 3. Being a prior, no data has yet been observed and so each sampled process is unique.

Figures 3 and 4 are generated by an adaptation of Scikit-learn's [11] program *Illustration of prior and posterior Gaussian process for different kernels*[2].

2.4.4 Posterior conditional distribution for the multivariate normal distribution

Assume that $p(x_1, x_2)$ is a joint Gaussian distribution with its parameters given by

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix},$$

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix},$$

$$\Lambda = \Sigma^{-1} = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix}.$$

[2] *http://scikit-learn.org/stable/auto_examples/gaussian_process/plot_gpr_prior_posterior.html*

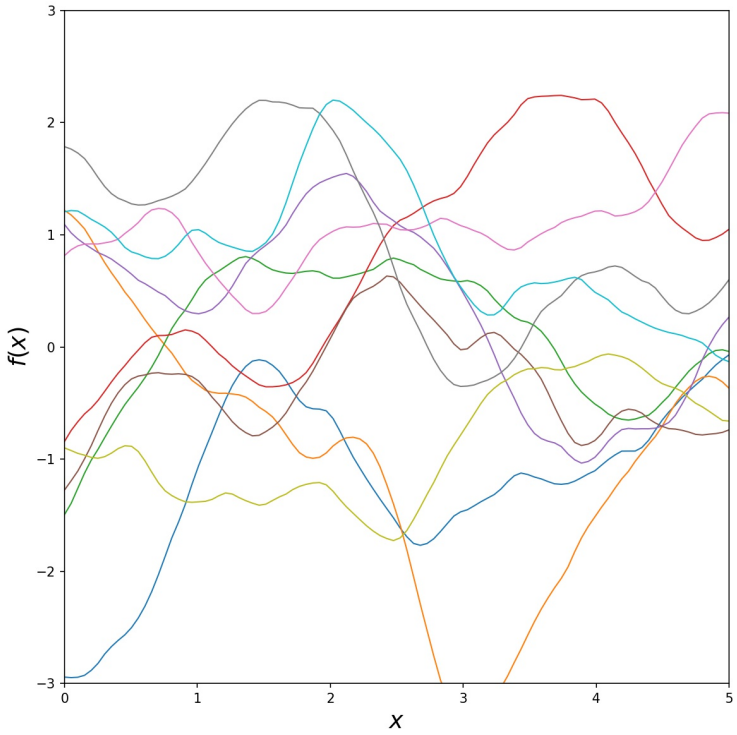


Figure 3: Ten samples from a GP prior distribution using the Matérn covariance function

From this, the marginal probabilities can be calculated as

$$p(x_1) = \mathcal{N}(x_1|\mu_1, \Sigma_{11})$$

$$p(x_2) = \mathcal{N}(x_2|\mu_2, \Sigma_{22}).$$

Therefore, the posterior conditional distribution can be formulated by

$$\begin{aligned} p(x_1|x_2) &= \mathcal{N}(x_1|\mu_{1|2}, \Sigma_{1|2}) \\ \mu_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \\ &= \mu_1 - \Lambda_{12}\Lambda_{11}^{-1}(x_2 - \mu_2) \\ &= \Sigma_{1|2}(\Lambda_{11}\mu_1 - \Lambda_{12}(x_2 - \mu_2)) \\ \Sigma_{1|2} &= \Sigma_{11} - \Sigma_{22}^{-1}\Sigma_{12}\Sigma_{21} = \Lambda_{11}^{-1}. \end{aligned} \tag{8}$$

2.4.5 Learning using noise-free data

Noise-free data is appropriate to consider as it can be encountered in computer simulations [12].

Consider a training set $\mathcal{A} = \{(x_i, f_i), i = 1 : N\}$ where $f_i = f(x_i)$ (thus observations in $\mathcal{A}$ are noise-free). X denotes a possible training input and X_* denotes the corresponding testing input. f is a training output and f_* is a testing output.

As per the prior, the distribution which links the training and testing outputs is given as

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{9}$$

where $K(X, X_*)$, for example, expresses the covariances of all training and testing points X and X_* within an N x N_* matrix, as there are N training points and N_* testing points.

Once data has been observed, the prior distribution may be *conditioned* on the data to form the posterior distribution. This is done by restricting the joint distribution (9) such that it includes only functions that coincide with the observed data. The posterior distribution is given by

$$p\left(f_*|X_*, X, f\right) = N\left(f_*|\mu_*, \Sigma_*\right) \tag{10}$$

where

$$\mu_* = \mu(X_*) + [K(X_*, X)][K(X, X)]^{-1}(f - \mu(X)) \tag{11}$$

and

$$\Sigma_* = K(X_*, X_*) - [K(X_*, X)][K(X, X)]^{-1}K(X, X_*). \tag{12}$$

This follows from the derived definitions for the marginal and conditional probabilities of a multivariate normal distribution in Section 2.4.4.

Figure 4 illustrates ten functions sampled from the posterior distribution of a GP using the Matérn covariance function. The red dots represent data on which the prior has been conditioned to form the posterior. Comparing this to Figure 3, observe that all of the functions in Figure 4 pass through the data. Thus it is evident that the posterior distribution is a distribution that has been conditioned on the data, so that any function sampled from the posterior would necessarily always be compatible with every data point.

Note that, because the training data is noise-free, a GP regression model trained on such data will be able to predict with absolute certainty the outcome $f(x)$ given that x is a data point that has already been observed [10].

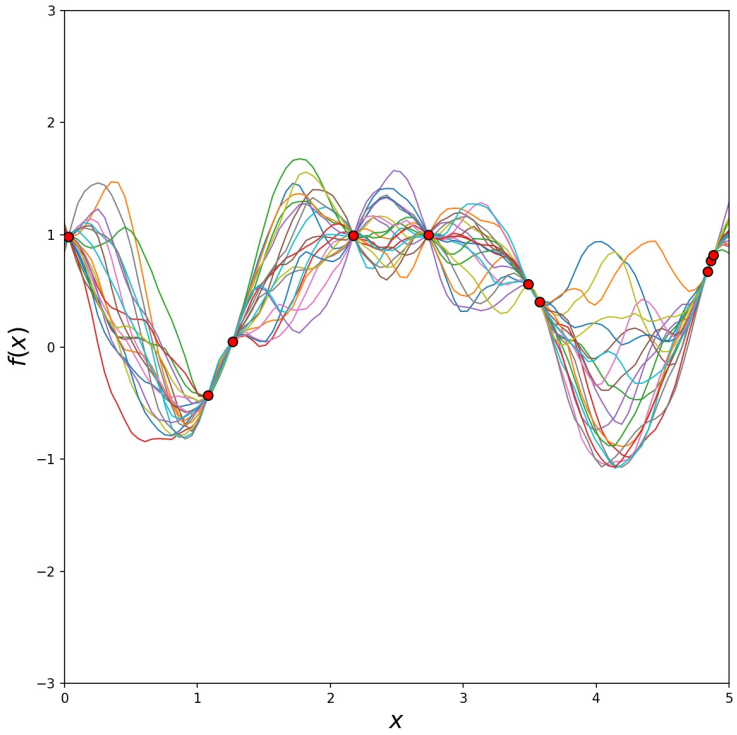


Figure 4: Ten samples from a GP posterior distribution using the Matérn covariance function

2.4.6 Learning using noisy data

Consider now the function $y = f(x) + \varepsilon$, which generates noisy data. Randomness (noise) occurs through the term ϵ.

With the added noise, the covariance of the data becomes

$$cov(y) = K(X, X) + \sigma_n^2 I$$

where σ_n^2 is assumed to be the variance of the Gaussian noise ε. $\sigma_n^2 I$ is a diagonal matrix because it is assumed that each noise term ε is independent.

Thus, the distribution which now links the training and testing outputs is given as

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right). \tag{13}$$

Then, given a zero mean, the posterior distribution becomes

$$p\left(f_* | X_*, X, y\right) = N\left(f_* | \mu_*, \Sigma_*\right) \tag{14}$$

where

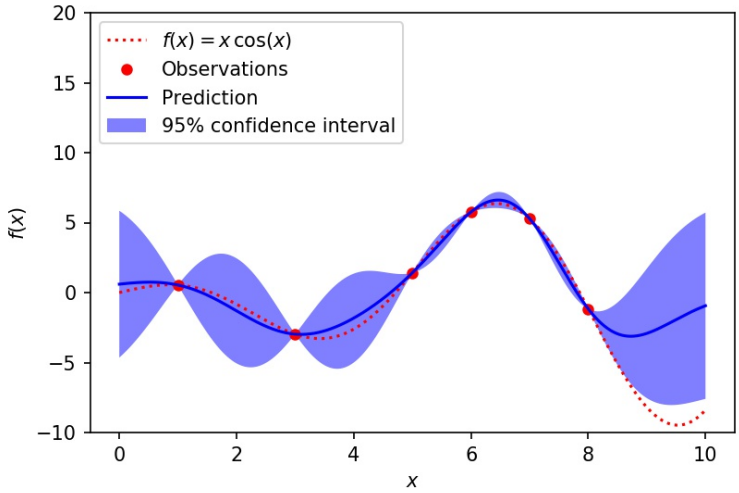


Figure 5: GP regression on noise-free data using the RBF covariance function

$$\mu_* = [K(X_*, X)][K(X, X) + \sigma_n^2 I]^{-1} y \tag{15}$$

and

$$\Sigma_* = K(X_*, X_*) - [K(X_*, X)][K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*). \tag{16}$$

Finally, for a single test point, the posterior mean can be written as

$$\bar{f}_* = k_*^T (K + \sigma^2 I)^{-1} y,$$

where k_* is the vector of covariance values between the single test point and the training points.

Figures 5 and 6 demonstrate the difference between modelling noisy data versus noise-free data. Both figures are generated by an adaptation of Scikit-learn's [11] program *Gaussian Processes regression: basic introductory example*[3].

The underlying functions which generate the data are $f(x) = xcos(x)$ and $f(x) = xcos(x) + \varepsilon$ respectively. The GP regression model is then created, using the radial-basis function (RBF[4]) as the covariance function. The noise level of the function shown in Figure 6 is indicated by vertical lines. The shaded area represents a 95% confidence interval for predictions. Notice how, in the noise-free case, predictions at each observation are made with absolute certainty. This is not so in the noisy case.

Sections 2.4.5 and 2.4.6 provide the formulation of GP regression model parameters. The following section addresses the estimation of these parameters.

[3] *http://scikit-learn.org/stable/auto_examples/gaussian_process/plot_gpr_noisy_targets.html*
[4] The Radial-basis function covariance function is often referred to as the Squared Exponential (SE) covariance function.

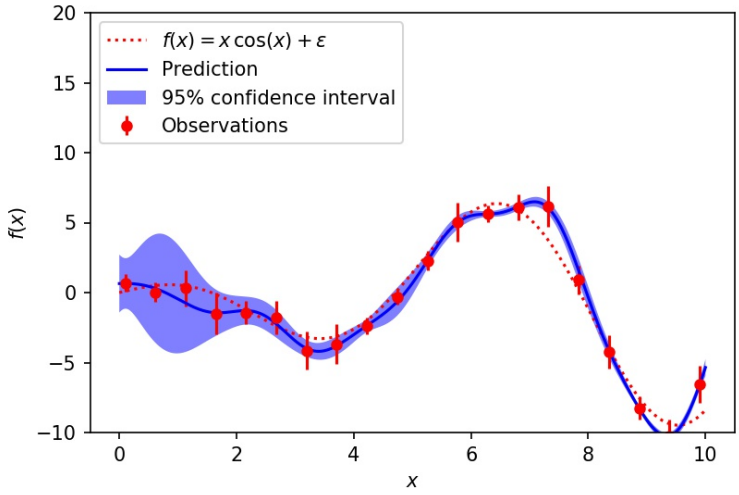


Figure 6: GP regression on noisy data using the RBF covariance function

2.4.7 Covariance function parameter estimation

Simply put, a final regression function is estimated (given some data) by *tuning* the parameters of the covariance function.

As explained in Murphy [10] Section 15.2.4, the parameters of the covariance function being used are estimated such that the marginal likelihood is maximised. This is done by using Bayes' theorem to marginalise out the Gaussian vector f

$$p(y|X) = \int p(y|f, X)p(f|X)df.$$

Then, since

$$p(f|X) = N(f|0, K)$$

and

$$p(y|f) = \prod_i N(y_i|f_i\sigma_y^2),$$

the log marginal likelihood is

$$\begin{aligned} \log p(y|X) &= \log N(y|0, K_y) \\ &= -\frac{1}{2}yK_y^{-1}y - \frac{1}{2}log|K_y| - \frac{N}{2}log(2\pi), \end{aligned} \tag{17}$$

where $-\frac{1}{2}yK_y^{-1}y$ represents the extent to which the model fits the data, $-\frac{1}{2}log|K_y|$ is the complexity

of the model and $-\frac{N}{2}log(2\pi)$ is a constant.

Model fit refers to the difference between predicted and observed data points; a better fitting model will have smaller residual values. Model complexity refers to the model's loss of generality. A model that is too complex would describe the observed data very well, but its implied loss of generality means that its performance on unseen data is compromised. This principle is broadly referred to as *Occam's razor.* In a more general sense, it postulates that the simplest (or most *parsimonious*) explanation for an event should be favoured. In the context of statistics, given a sufficient model fit, the most parsimonious model should always be used.

Therefore, in parameter estimation, the optimisation problem is essentially balancing model fit and model complexity. The aim here is to find the optimal parameters to achieve this balance. In general, as model fit is increased, model complexity is decreased and vice versa.

A compromise is achieved by maximising the marginal likelihood

$$\begin{aligned}\frac{\partial}{\partial\theta_q}\log p(y|x) &= \frac{1}{2}y^T K_y^{-1}\frac{\partial K_y}{\partial\theta_q}K_y^{-1}y - \frac{1}{2}tr\left(K_y^{-1}\frac{\partial K_y}{\partial\theta_q}\right)\\ &= \frac{1}{2}tr\left((\alpha\alpha^T - K_y^{-1})\frac{\partial K_y}{\partial\theta_q}\right),\end{aligned}$$

where θ represents the respective covariance function's parameters and $\alpha = K_y^{-1}y$. Note that the form of $\frac{\partial K_y}{\partial\theta_q}$ depends on the form of the covariance function.

In order to compute the actual parameter estimates, the Cholesky decomposition[5] (which is said by Murphy [10] to be relatively robust) may be used.[6] This was investigated by Rasmussen and Williams [12]; their work is used as a guide here and their pseudocode is shown in Algorithm 1.

To start, the computation that is being performed is given by

$$K_y = LL^T$$

and the predictive mean is

$$\overline{f_*} = k_*^T K_y^{-1}y.$$

Inputted into the algorithm are the training inputs (X), the training outputs (y), the test inputs x_*, the covariance function κ and the noise level σ_n^2. Note that the $\sigma_y^2 I$ in Line 1 is for noisy observations.

[5] See *Cholesky factorisation. Encyclopedia of Mathematics. URL:* `http://www.encyclopediaofmath.org/index.php?title=Cholesky_factorization&oldid=37467`

[6] A full Bayesian solution may also be used to compute the posterior of the parameters. See Murphy [10] Section 5.2.4.2, page 523.

Algorithm 1 Cholesky decomposition for GP Regression

Inputs: training inputs (X), training outputs (y), test inputs (x_*), covariance function (κ), noise level (σ_n^2)

1 $L = cholesky(K + \sigma_y^2 I)$
2 $\alpha = L^T \setminus (L \setminus y)$
3 $\mathbb{E}(f_*) = k_*^T \alpha$
4 $v = L \setminus k_*$
5 $var(f_*) = \kappa(x_*, x_*) - v^T v$
6 $\log p(y|X) = -\frac{1}{2} y^T \alpha - \sum_i logL_{ii} - \frac{N}{2} log(2\pi)$

Outputs: log marginal likelihood $\log p(y|X)$, predictive mean $\overline{f_*}$, predicted variance $var(f_*)$

2.5 Covariance functions

This subsection uses Chapter 4 of Rasmussen and Williams [12] as a theoretical guide, and an adapted version of the Python program provided in Scikit-learn's [11] *Illustration of prior and posterior Gaussian process for different kernels*[7]. Then, Scikit-learn's corresponding GPs documentation[8] is closely followed for the description of each covariance function's parameters.

The covariance function defines how similar any given data point is to its neighbours, and by this it makes assumptions about the characteristics of the data to be modelled [12]. Therefore, the choice of a suitable covariance function is fundamental in creating a meaningful model.

A covariance function may be described as stationary or isotropic (or both). An isotropic covariance function is necessarily stationary; a stationary function is not necessarily isotropic. A stationary covariance function (which is unaffected by translations in the training data) is merely a function of the difference between x and x'

$$\kappa(x, x') = f(x - x'), \tag{18}$$

whereas an isotropic covariance function (which is unaffected by all rigid motions in the training data) is a function of the Euclidean distance between x and x'

$$\kappa(x, x') = f(|x - x'|). \tag{19}$$

A dot-product covariance function (which is unaffected by rotations of the training data about the origin) has neither of these properties. Rather, it is a function of $x \cdot x'$. One such example would be

$$\kappa(x, x') = \sigma_0^2 + x \cdot x',$$

where σ_0^2 is the known variance of the GP. A dot-product covariance function is necessarily non-

[7] `http://scikit-learn.org/stable/auto_examples/gaussian_process/plot_gpr_prior_posterior.html`
[8] `http://scikit-learn.org/stable/modules/gaussian_process.html#gaussian-process-kernel-api`

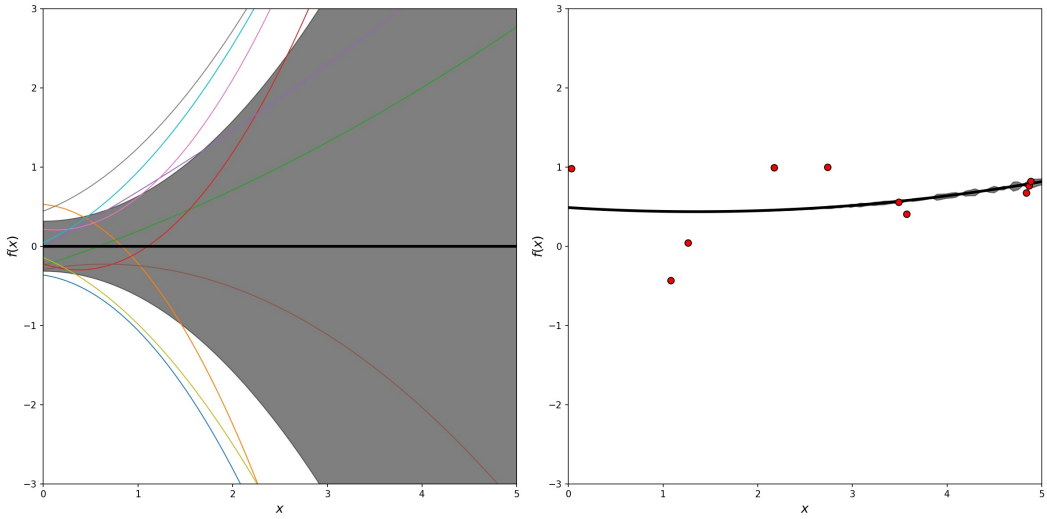


Figure 7: GP using the dot-product covariance function

stationary.

In Figures 7 to 10, the black line indicates the predictive mean and the shaded area indicates one standard deviation from this mean (interpretable as predictive confidence). Ten samples from the prior distribution are shown on the left; ten samples from the posterior distribution are shown on the right.

2.5.1 Dot-product covariance function

$$\kappa(x, x') = \sigma_0^2 + x \cdot x'$$

This is a non-stationary covariance function with one parameter, σ_0. It is said to be homogenous if $\sigma_0 = 0$; inhomogeneous otherwise. Also, the parameter may be raised to an appropriate exponent.

In Figure 7, $\sigma_0 = 1$ in the prior and σ_0 is squared.

2.5.2 Radial-basis function (RBF) covariance function

$$\kappa(x, x') = \sigma_f^2 exp\left(1 - \frac{(x - x')^2}{2l^2}\right)$$

This stationary covariance function has the parameters l and σ_f^2. The former is the length scale parameter, which determines the degree to which the function varies over the horizontal plane. The latter determines the variation on the vertical plane. A greater l translates to a smoother, less erratic function.

The RBF covariance function is isotropic if l is a scalar, and *anisotropic* (meaning *not* isotropic) if l is a vector (whose number of dimensions must match the number of data points inputted).

The RBF covariance function with $l = 1$ and $\sigma_f^2 = 1$ in the prior is illustrated in Figure 8.

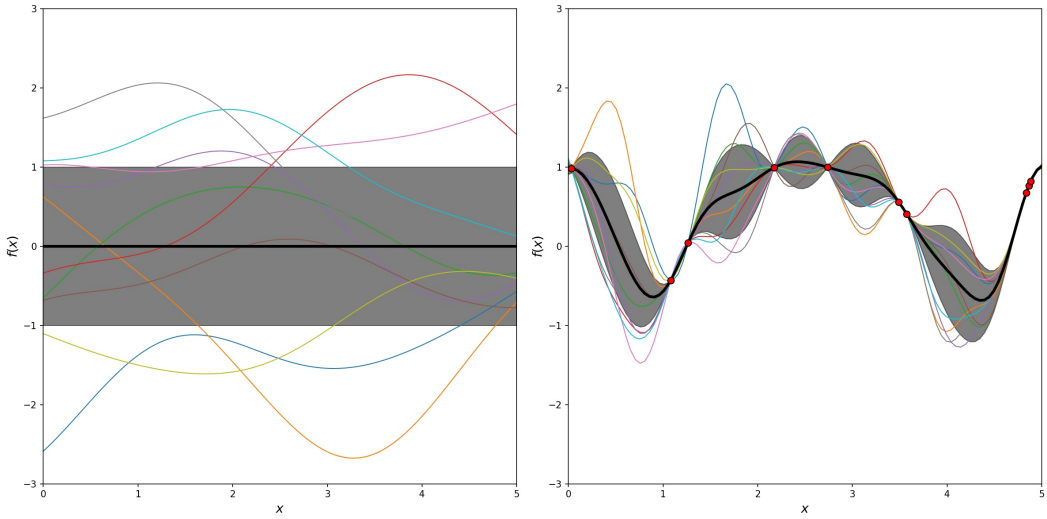


Figure 8: GP using the radial-basis function covariance function

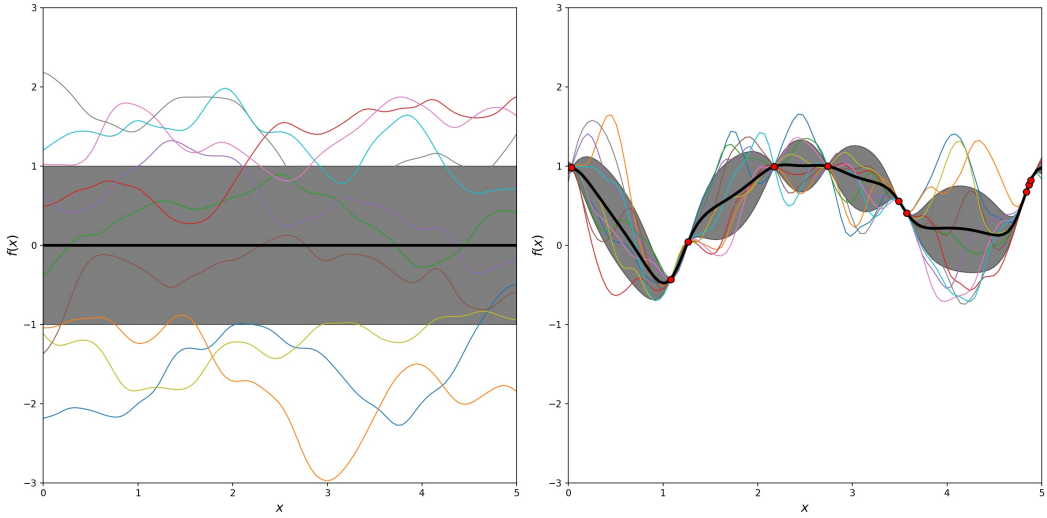


Figure 9: GP using the rational quadratic covariance function

2.5.3 Rational quadratic covariance function

$$\kappa(x, x') = \left(1 + \frac{(x - x')^2}{2\alpha l^2}\right)^{-\alpha}$$

Following from the RBF, the length-scale parameter l is still present and has the same effect. The RBF is now extended though, to include a *scale mixture* parameter α. This is included as the rational quadratic expresses an infinite number of RBF covariance functions all having different length-scale parameters, summed together. As above, the rational quadratic function is stationary and may be either isotropic or anisotropic.

The rational quadratic converges to the RBF as α tends towards infinity.

In Figure 9, the rational quadratic is used with $l = 1$ and $\alpha = 0.1$ in the prior.

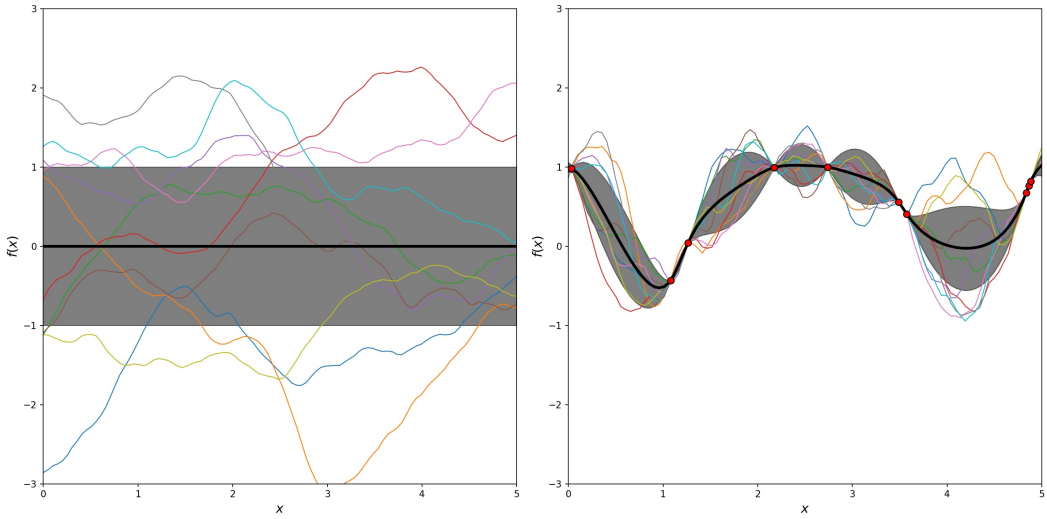


Figure 10: GP using the Matérn covariance function

2.5.4 Matérn covariance function

$$\kappa(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}(x - x')}{l} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}(x - x')}{l} \right)$$

The Matérn covariance function is an even more elaborate extension of the RBF. l works as before, but now the new parameter ν directly controls the smoothness of the function. As ν approaches infinity, so the Matérn covariance function becomes identical to the RBF. A modified Bessel function is expressed by K_{ν}.

Like the RBF, the Matérn is stationary and may be either isotropic or anisotropic.

A Matérn covariance function is shown in Figure 10 with $l = 1$ and $\alpha = 1.5$ used for the prior.

2.5.5 Exponential-sine-squared covariance function

$$\kappa(x, x') = exp\left(-\frac{2sin^2(\pi|x - x'|/p)}{l^2} \right)$$

The exponential-sine-squared covariance function, derived by Mackay [9], is a periodic function. It may be used to model periodic data such as atmospheric carbon dioxide concentration or cyclical stock prices.

l is again the length-scale parameter, which determines the function's smoothness. The new *periodicity* parameter p, controls the length of each period. This covariance function may be either isotropic or anisotropic.

In Figure 11, an exponential-sine-squared covariance function is used with $l = 1$ and $p = 3$ in the prior.

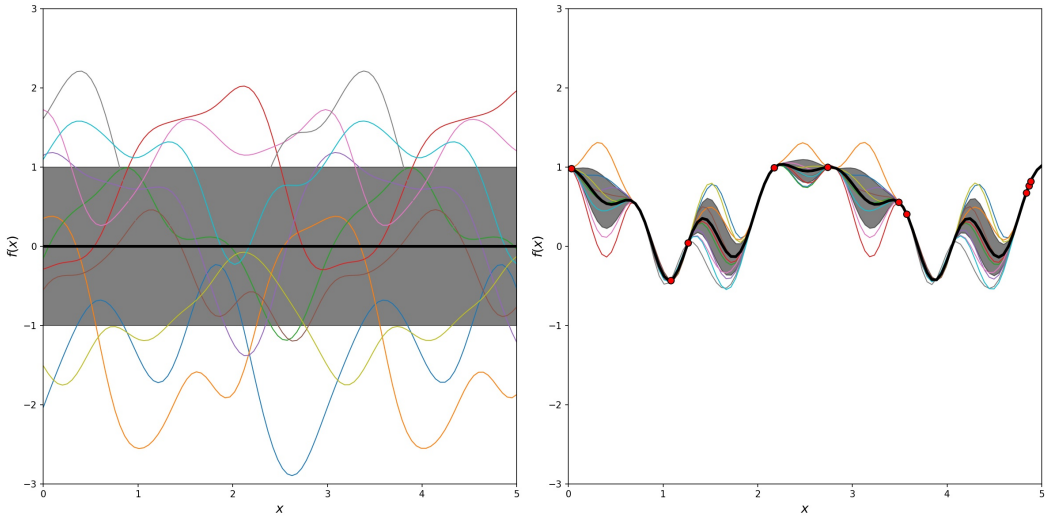


Figure 11: GP using the exponential-sine-squared covariance function

2.6 Classification

2.6.1 Introduction

This section is based on Murphy [10] Section 15.3.1, and Rasmussen and Williams Section 3.3 and 3.4.

In classification, the objective is to classify data into one of a number of classes. Training data is observed to form a predictive model, and then testing data is used to evaluate its predictive ability. Only two-class GP classification is investigated in this section.

The aim here is to, for every input point, compute the probability that the given input point belongs to a certain class. If the probability that an input point belongs to a certain class is greater than or equal to a certain value (called the decision *threshold* or *boundary*, which is usually set at 0.5), it is classified as belonging to the class. This is called probabilistic classification.

We wish to model the probability that a given input point belongs to a certain class, as opposed to just modelling a real-valued output for every input.

An example of two-class classification could be in determining whether a given fruit is either a litchi or a mango. This assessment could be based on attributes such as the weight and colour of each fruit: a lighter fruit whose skin is red or brown would more likely be classified as a litchi; a heavier green or yellow fruit would probably be classified as a mango.

The following mathematical discourse is based on Murphy [10] Section 15.3.1 and Rasmussen and Williams [12] Sections 3.3 and 3.4.

2.6.2 Model formulation

The obvious way to develop a GP classifier would be to formulate a posterior distribution using the Gaussian prior (5) and likelihood function (17); however, the Gaussian prior is incompatible with the

Bernoulli likelihood function. Thus, the logistic function is incorporated, and the classification model is formulated in the following way:

Firstly, our aim is to model $p(y_i|x_i)$. That is, to create a model which returns the probability that an input data point x_i belongs to the class y_i. Since there are only two classes, y_i is either -1 or 1. Now, we take a sample function $f(x)$ from a GP with mean 0 and covariance function κ, so $f(x) \sim \mathcal{GP}(0, \kappa)$.

Here, f is a latent function as we do not directly use any of its values. This can also be referred to as a *nuisance* function, since the output of the function is irrelevant to us; only the output of π is of interest for each test input x_*. The use of the nuisance function f is only for the sake of developing the model and will eventually be integrated out when the posterior predictive distribution is computed.

Lastly, we define a logistic function $\delta(z) = sigm(z)$ through which we observe the output of f. This is so that, as it is done in LR, we can model the *probability* of the response variable being a 1 or a 0 as opposed to modelling the response variable itself. This output is taken from the function we define as π, on which we wish to derive a prior distribution. The resulting model is given by

$$\pi(x) \triangleq p(y = +1|x) = \delta(f(x)). \tag{20}$$

For each test input x_* we wish to obtain a distribution of the the corresponding f, which we call f_*. So, the likelihood is given by

$$p(f_*|X, y, x_*) = \int p(f_*|X, x_*, f)p(f|X, y)df. \tag{21}$$

Then for predictions, we use

$$\bar{\pi}_* \triangleq p(y_* = +1|X, y, x_*) = \int \delta(f_*)p(f_*|X, y, x_*)df_*, \tag{22}$$

where $\delta(f_*)$ is the prior and $p(f_*|X, y, x_*)$ is the likelihood.

2.6.3 Posterior computation

Since the likelihood function is sigmoidal, it is non-normal and therefore intractable. This means that the integral cannot be solved. In order to overcome this problem, we use Gaussian approximation to determine a normalised version of the likelihood function so that it may become tractable. We then use the Laplace approximation on this function to analytically approximate the integral.

Following Murphy [10], the log of the unnormalised posterior is given by

$$
\begin{aligned}
l(f) &= \log p(y|f) + \log p(f|X) \\
&= \log p(y|f) - \frac{1}{2} f^T K^{-1} f - \frac{1}{2} \log |K| - \frac{N}{2} \log 2\pi. \qquad (23)
\end{aligned}
$$

In order to minimise the above function $l(f)$, the gradient used is

$$g = -\nabla log p(y|f) + K^{-1} f,$$

and the Hessian matrix used is

$$H = W + K^{-1}.$$

Ultimately, by using Gaussian approximation, the posterior distribution is computed as

$$p(f|X, y) \approx \mathcal{N}(f, (K^{-1} + W)^{-1}). \qquad (24)$$

This expression is now normalised, and so it may be analytically approximated using Laplace approximation.

2.6.4 Posterior predictive computation

The posterior predictive distribution is formed by all possible unobserved data points, given the already observed data.

For the purpose of this computation, we define x_* to be a test point which will be inputted into the latent function f introduced in Section 2.6.2. We define f_* as a latent function used in the context of posterior predictive computation.

The posterior predictive distribution is finally given by

$$p(f_*|x_*, X, y) = \mathcal{N}(\mathbb{E}[f_*], var[f_*]). \qquad (25)$$

(For more detail on how this is derived, see Murphy [10] pages 526 and 527)

Translating (25) into a predictive distribution that is able to handle binary responses, we require the use of the following function:

Algorithm 2 Cholesky decomposition for binary GP classification using Gaussian approximation

1 [The MAP estimate is computed in lines 2 to 9 by using IRLS.]
2 $f = 0$
3 repeat
4 $W = -\nabla\nabla logp(y|f)$
5 $B = I_N + W^{\frac{1}{2}} K W^{\frac{1}{2}}$
6 $L = cholesky(B)$
7 $b = Wf + \nabla logp(y|f)$
8 $a = b - W^{\frac{1}{2}} L^T \setminus (L \setminus (W^{\frac{1}{2}} Kb))$
9 $f = Ka$
10 until *converged*
11 $logp(y|X) = logp(y|f) - \frac{1}{2} a^T f - \sum_i logL_{ii}$
12 [Prediction is performed in lines 13 to 16.]
13 $\mathbb{E}[f_*] = k_*^T \nabla logp(y|f)$
14 $v = L \setminus (W^{\frac{1}{2}} k_*)$
15 $var[f_*] = k_{**} - v^T v$
16 $p(y_* = 1) = \int sigm(z)\mathcal{N}(z|\mathbb{E}[f_*], var[f_*])dz$

$$\begin{aligned} \pi_* &= p(y_* = 1|x_*, X, y) \\ &\approx \int \delta(f_*) p(f_*|x_*, X, y) df_* \end{aligned} \tag{26}$$

Monte Carlo or probit approximation may be used to estimate (26).

2.6.5 Marginal likelihood computation

As in the regression case, the parameters of the covariance function being used are estimated such that the marginal likelihood is maximised given the data. The Laplace approximation yields

$$\log p(y|X) \approx l(\hat{f}) - \frac{1}{2} \log |H| + constant, \tag{27}$$

and so the marginal likelihood is finally given as

$$\log p(y|X) \approx \log p(y|\hat{f}) - \frac{1}{2} \hat{f}^T K^{-1} \hat{f} - \frac{1}{2} \log |K| - \frac{1}{2} \log |K^{-1} + W|. \tag{28}$$

2.6.6 Final parameter estimation

The equations in Sections 2.6.2, 2.6.3, 2.6.4 and 2.6.5 all culminate in Algorithm 2, which estimates the parameters of a given covariance function to fit the training data. Once again, the Cholesky decomposition is used. The algorithm was created by Rasmussen and Williams [12] and adapted by Murphy [10].

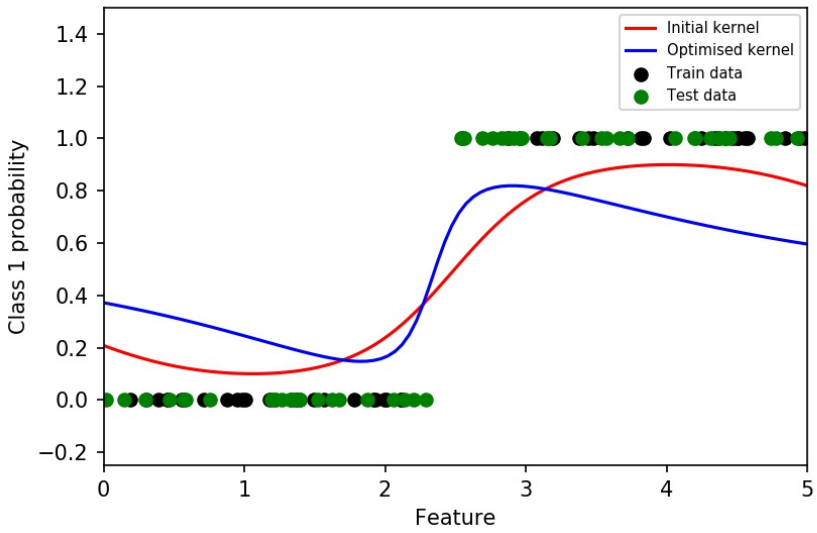


Figure 12: Initial RBF covariance function (blue) and optimised RBF covariance function (red)

2.6.7 Illustrations

The following illustrations are generated by adaptations of example programmes used in Scikit-learn's documentation on GPs [9] [11]. The accompanying descriptions in the documentation are also used as a guide in writing this section.

Figures 12 and 13 show how the parameters of the covariance function are *tuned* to suit the given data. The blue function is the RBF covariance function before any data has been considered; the red function represents the same same covariance function now *optimised* for the given data.

Observe the sigmoid shape of the curves, which results from the use of the logistic function.

There is a slight quirk in this example: the log-loss (0.173 for the initial and 0.313 for the optimised covariance function) indicates that the initial covariance function performs better on the test data than the *tuned* covariance function. This is because of the more dramatic change in probability that the optimised covariance function displays. This steep part of the curve implies that more data points that are on the boundary of each class will be assigned probabilities closer to 0.5, implying greater uncertainty for their associated predictions. This phenomenon can be attributed to the classifier's use of the Laplace approximation.

Figures 12 and 13 are generated by an adaptation of Scikit-learn's [11] program *Probabilistic predictions with Gaussian process classification (GPC)*[10].

The corresponding Figure 13 shows the relationship between the RBF's length scale parameter (l) and the log-marginal-likelihood. As explained, the log-marginal-likelihood must be maximised by choosing

[9] `http://scikit-learn.org/stable/modules/gaussian_process.html#basic-kernels`
[10] `http://scikit-learn.org/stable/auto_examples/gaussian_process/plot_gpc.html`

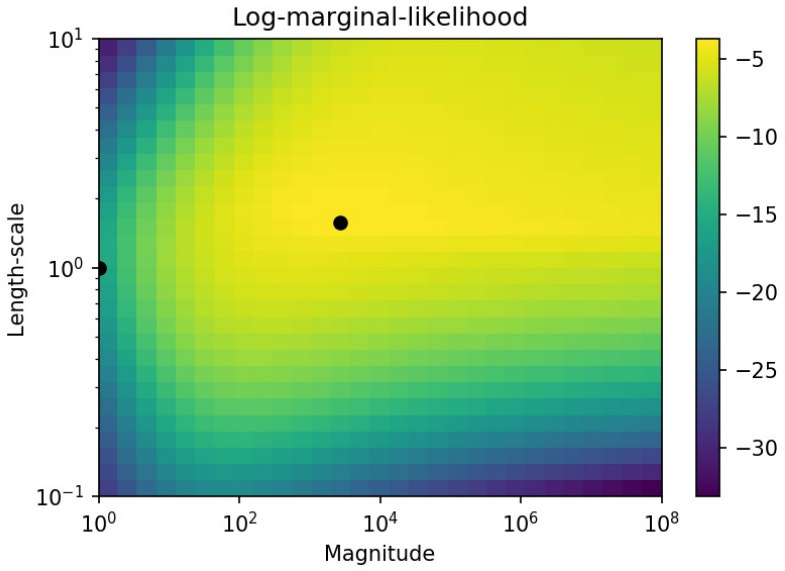


Figure 13: Relationship between log-marginal-likelihood and the length-scale parameter of the RBF

the optimal covariance function parameters. This process is illustrated in Figure 13. The black dot at 10^0 is the initial length-scale parameter, and the other black dot is the length-scale parameter value such that the log-marginal-likelihood is maximised. This parameter value dictates the shape of the blue covariance function in Figure 12.

Figure 14 is generated by an adaptation of Scikit-learn's [11] program *A two-dimensional classification example showing iso-probability lines for the predicted probabilities*[11]. It demonstrates how probabilities are used to make the classification decision for each input point. Here, the dot-product kernel is used and the probabilities are transformed into iso-probability lines. We are predicting whether a given data point (x_1, x_2) used in the function $G(x)$ will yield ≤ 0 or > 0. The value computed (indicated by the colour map white to black) is the probability that, given the input data point, the outcome of $G(x)$ will be smaller than or equal to zero. The data points which are on or below the iso-probability line of 0.5 are classified as belonging to this class (blue), and those that are above the 0.5 iso-probability are *not* classified as belonging to the class (red). Of course, there are an infinite number of iso-probability lines.

Figure 15 is generated by an adaptation of Scikit-learn's [11] program *Illustration of Gaussian process classification (GPC) on the XOR dataset*[12]. It shows classification done on XOR data using the RBF covariance function (left) and the dot-product covariance function (right). The RBF covariance function is stationary and isotropic, and the dot-product covariance function is non-stationary. The log-marginal-likelihood values indicate that the dot-product kernel yields the best performance in this case. However, as indicated in the program's description, stationary covariance functions do perform better in general. The program description attributes this exception to the fact that the class boundaries in the XOR data

[11] `http://scikit-learn.org/stable/auto_examples/gaussian_process/plot_gpc_isoprobability.html`
[12] `http://scikit-learn.org/stable/auto_examples/gaussian_process/plot_gpc_xor.html`

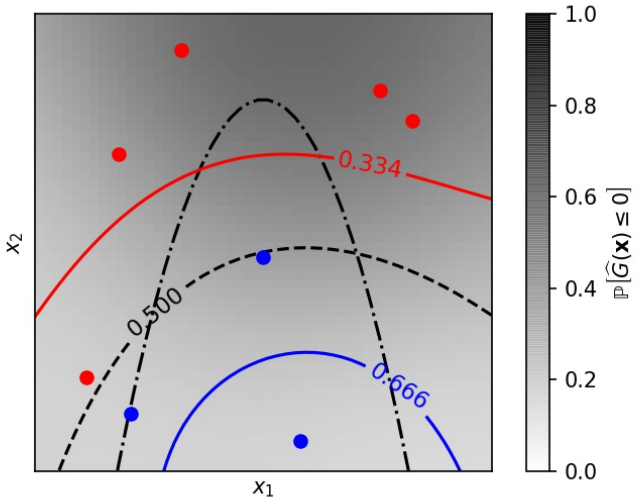


Figure 14: Iso-probability lines showing how data points are classified

are linear.

3 Application

3.1 Experimental design

Our research question is, *Does GP classification perform better than SVM, RF and LR classification when class imbalance is present?*

The performance of each technique is tested using 500 observations on three different datasets with the class proportion set to 0.5, 0.4, 0.3, 0.2, 0.1 and then 0.05.

3.2 Datasets

Data is simulated from Scikit-learn's [11] 'toy' data generators[13]. These are 'Make moons', 'Make circles' and 'Make classification'. 'Make moons' generates data points which resemble two interleaving semicircles and 'Make circles' generates data points which resemble two concentric circles. Both of these data formations are considered difficult to model, hence their inclusion in the experiments. Lastly, 'Make classification' is configured to generate linearly separable data. For each observation, two X points are generated together with their corresponding binary Y point. Refer to the 'Input data' panes of Figures 22 and 23 in Section A of the appendix for illustrations of the datasets.

[13] `http://scikit-learn.org/stable/datasets/index.html`

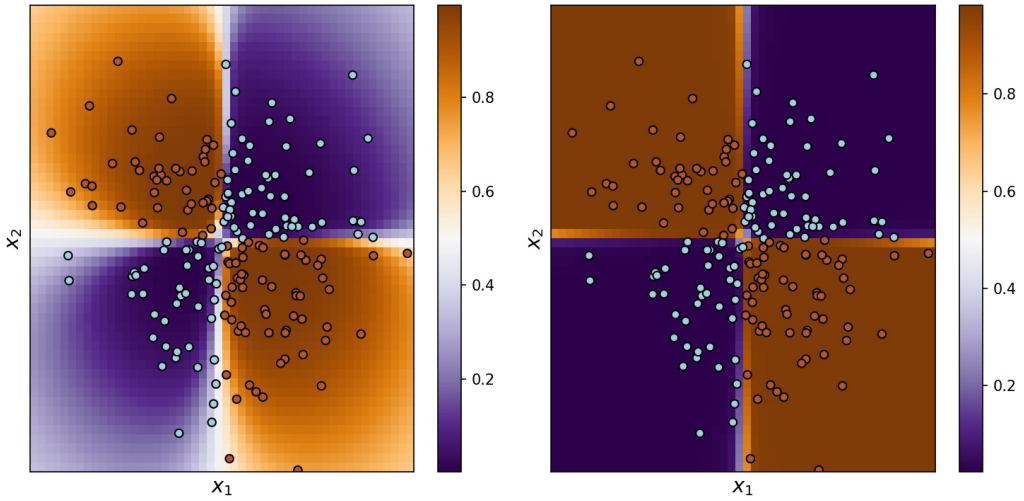


Figure 15: Comparison of RBF and dot-product covariance functions on XOR data in GP classification

	Predicted response: 0	Predicted response:1	
Actual response: 0	True negative (TN)	False positive (FP)	Total number of times actual response is 0
Actual response: 1	False negative (FN)	True positive (TP)	Total number of times actual response is 1
	Total number of times a 0 is predicted	Total number of times a 1 is predicted	

Table 1: Confusion matrix format

3.3 Evaluation metrics

3.3.1 Confusion matrix

The confusion matrix provides a simple breakdown of whether predictions turn out to be correct. We use elements from this matrix, shown in Table 1, to compute the metrics in 3.3.2 and 3.3.3.

The TN value is the number of times the response is correctly predicted as a 0; the FN value is the number of times the response is incorrectly predicted as a 0 (type II error).

Similarly, the TP value is the number of times the response is correctly predicted as a 1; the FP value is the number of times the response is incorrectly predicted as a 1 (type I error).

3.3.2 True negative rate

$$= \frac{\text{Number of times an observation is correctly predicted as class 0}}{\text{Total number of actual class 0 observations}} = \frac{TN}{TN + FP}$$

The true negative rate (TNR) indicates the proportion of class 0 observations correctly classified as being class 0 observations. This measure is completely relevant in evaluating a classifier that has processed imbalanced data, as it gives us a measure of how well the classifier is able to correctly classify minority class observations. In our experiments, class 0 is the minority class.

The TNR is crucial in practical applications where class imbalance is present. One such situation is credit scoring. The TNR in this context would be the proportion of defaulters who are correctly classified as defaulters. Classifying defaulters as non-defaulters would potentially be damaging to the company granting credit to entities that are not creditworthy. It would be far less damaging to misclassify non-defaulters as defaulters: the bank would merely lose out on a money-making client rather than bear the financial consequences of unrecoverable credit. Therefore, we focus on a classifier's ability to identify defaulters, with the potential cost of misclassifying some non-defaulters. This is shown by the TNR.

3.3.3 The Matthews correlation coefficient

The Matthews correlation coefficient (MCC) [1] measures the performance of binary classifiers. It simply describes the correlation between the predicted values and the actual values. This measure is particularly well-suited to imbalanced data as it is critical of all components of the confusion matrix. It is calculated as

$$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}.$$

The coefficient ranges from -1 to +1. A measure of -1 indicates complete discord between predicted values and actual values; 0 indicates that the correlation between predicted and actual values is not necessarily any better than a prediction made by a classifier that makes decisions randomly; and 1 indicates perfect agreement between predicted and actual values. It is an elegant measure that gives an indication of overall classifier performance yet is also sensitive to the classifier potentially misclassifying observations of the minority class, and so in this paper we use it as the primary measure upon which to compare models.

3.3.4 Classification accuracy is misleading

It should be noted that the classification accuracy metric does not feature in this paper as it can be highly misleading, especially in the case of class imbalance. Consider a hypothetical dataset with class 1 representing 90% of the observations and class 0 representing the other 10%. If a classifier were to classify all observations in this dataset as belonging to class 1, it would yield an accuracy of 90%. This is a high score, yet the classifier would completely have failed in its task of discriminating between the two classes.

Therefore, only the MCC and TNR are used.

3.4 Summary of other techniques

The other three classification techniques that are compared are vastly different in how they work: SVMs create classification regions by constructing hyperplanes such that the distance from each hyperplane to the nearest set of training points belonging to different classes is maximised [10]; LR applies a regression model to the log-odds observed through the logistic function [5]; and RFs take bootstrap samples of data from which they create decision trees that are eventually aggregated to calculate a majority vote for classification decisions [6].

3.5 Experiment

The experiments are carried out using the Python programming language and the Scikit-learn machine learning library [11]. The program used to conduct the experiments is an adaptation of Scikit-learn's program 'Classifier Comparison' [14].

Each model is trained on 60% of the dataset and testing is done on the other 40%. Stratification is used in splitting the data into training and testing sets, and so the proportion of each class present in the sets stays the same.

Imbalance between the two classes is created by dropping random observations of the one class while keeping the total number of observations at 500.

The RBF covariance function is used for both GP and SVM classification, with parameters $\sigma_f^2 = 1$ and $l = 1$ for the GP and $\gamma = 2$ for the SVM.

We model each of the three datasets 30 times, with the dataset simulation and training set selection randomised on every iteration. The results are then averaged. This is called *cross-validation*.

4 Results

4.1 Results on the 'Make moons' dataset

On this dataset, Figure 16 indicates that according to the MCC, GP classification is the best overall classifier initially and when class imbalance is introduced—except where the minority class proportion is 0.3 (where it is marginally outperformed by SVM classification) and 0.05 (where it is outperformed by RF classification). The corresponding data can be found in Table 2.

In terms of the TNR, Figure 17 and Table 3 show that the GP classifier correctly classifies a greater number of minority class observations than the other classifiers where the minority class proportion is 0.5, 0.4, 0.3 and 0.2. For 0.1 and 0.05, RF classification yields the highest TNR.

[14] `http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html`

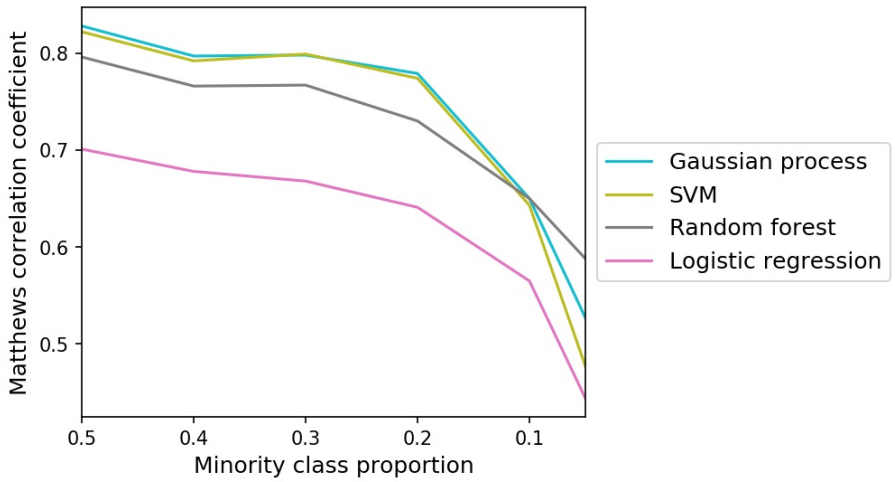


Figure 16: MCC with varying minority class proportion on the 'Make moons' dataset for SVM, GP, RF and LR classifiers

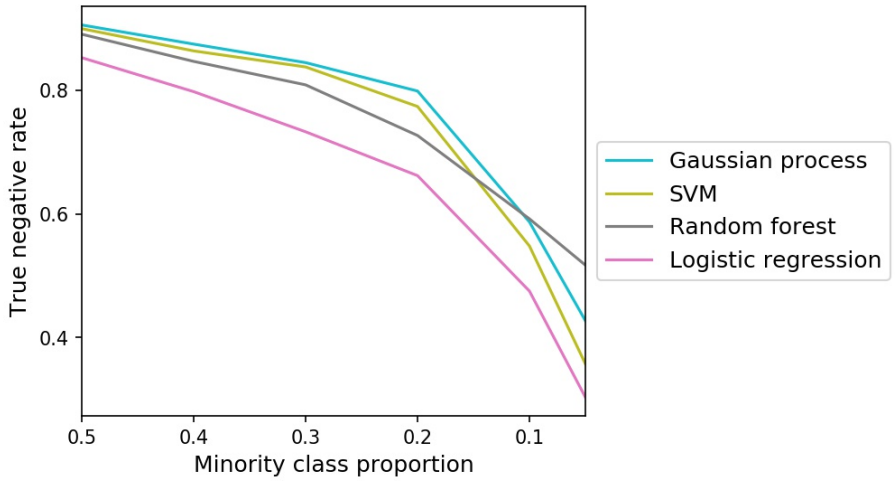


Figure 17: TNR with varying minority class proportion on the 'Make moons' dataset for SVM, GP, RF and LR classifiers

GP classification attains the highest Matthews correlation coefficient in three out of the five imbalanced tests. However, GP and SVM classification perform very similarly in this experiment, and so with their ability to accurately classify observations of both classes, either of them may be considered the best classifier in this case. RF classification is possibly the most suitable classifier for extremely imbalanced data.

4.2 Results on the 'Make circles' dataset

As shown in Figure 18 and Table 4, the MCC indicates that GP classification outperforms all techniques at every level of class imbalance.

Figure 19 and Table 5 indicate that the GP and SVM classifiers accurately classify a similar number of minority class observations compared to the other techniques where the minority class proportion is

Minority class proportion	SVM	GP	RF	LR
0.5	0.822	0.828	0.796	0.701
0.4	0.792	0.797	0.766	0.678
0.3	0.799	0.798	0.767	0.668
0.2	0.774	0.779	0.730	0.641
0.1	0.643	0.650	0.650	0.565
0.05	0.477	0.527	0.588	0.444

Table 2: MCC on the 'Make moons' dataset for SVM, GP, RF and LR classifiers with varying minority class proportion

Minority class proportion	SVM	GP	RF	LR
0.5	0.900	0.906	0.891	0.853
0.4	0.864	0.875	0.847	0.798
0.3	0.838	0.845	0.809	0.733
0.2	0.774	0.799	0.727	0.662
0.1	0.548	0.587	0.592	0.475
0.05	0.357	0.427	0.517	0.303

Table 3: TNR on the 'Make moons' dataset for SVM, GP, RF and LR classifiers with varying minority class proportion

0.5 and 0.4, but for all other levels of minority class proportion GP classification is the top performer in this regard.

Considering the classification performance reflected by the MCC, GP classification may be considered the best overall classifier on this dataset when class imbalance is present. When the data is balanced, SVM and GP classification perform similarly.

4.3 Results on the 'Linearly separable' dataset

On the linearly separable data, Figure 20 and Table 6 illustrate that GP classification is the best overall when the data is imbalanced. When the classes are equally represented, SVM classification attains the highest MCC. All of the classifiers perform similarly on this dataset though, except at the 0.05 minority class proportion, where the performance of the GP classifier surpasses the performance of the other techniques significantly.

Observing Figure 21 and Table 7, the GP classifier correctly classifies the greatest number of minority class observations at all levels of class imbalance except 0.5, where SVM classification is marginally better.

Minority class proportion	SVM	GP	RF	LR
0.5	0.777	0.791	0.744	-0.050
0.4	0.769	0.772	0.739	0.029
0.3	0.744	0.758	0.698	0.000
0.2	0.685	0.719	0.638	0.000
0.1	0.522	0.631	0.534	0.000
0.05	0.173	0.487	0.413	0.010

Table 4: MCC on the 'Make circles' dataset for SVM, GP , RF and LR classifiers with varying minority class proportion

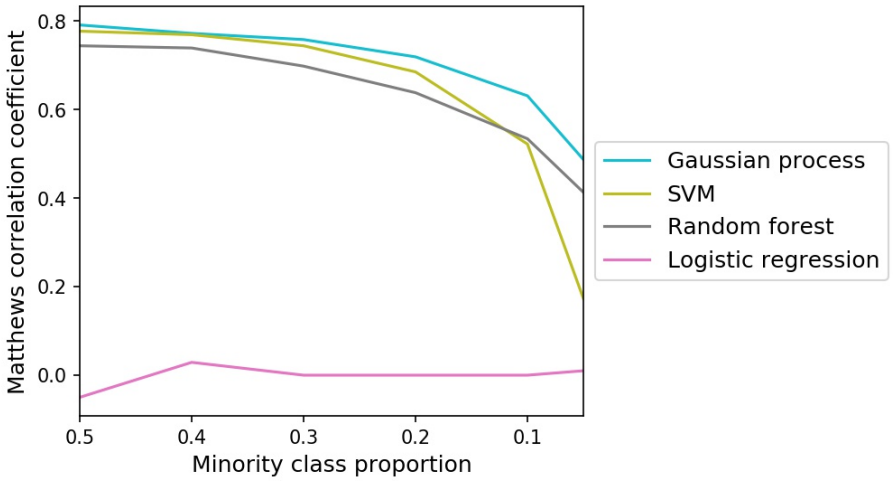


Figure 18: MCC with varying minority class proportion on the 'Make circles' dataset for SVM, GP, RF and LR classifiers

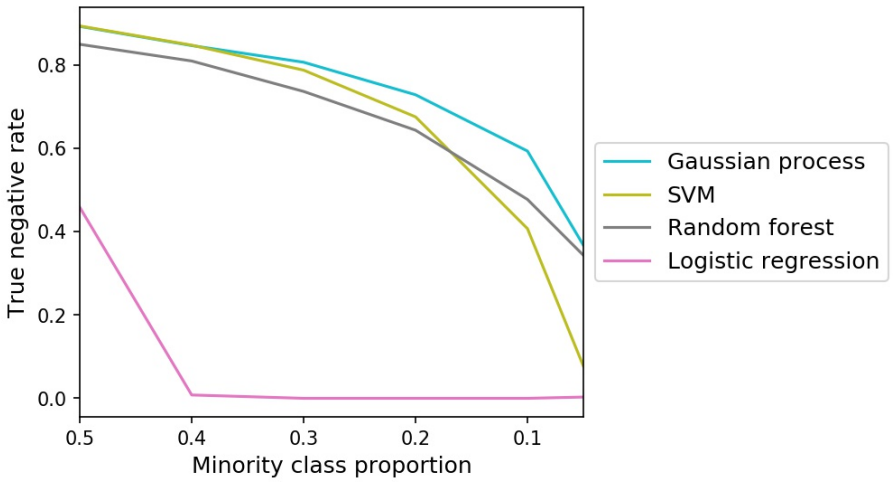


Figure 19: TNR with varying minority class proportion on the 'Make circles' dataset for SVM, GP, RF and LR classifiers

Minority class proportion	**SVM**	**GP**	**RF**	**LR**
0.5	0.893	0.892	0.849	0.458
0.4	0.847	0.846	0.809	0.008
0.3	0.787	0.806	0.736	0.000
0.2	0.675	0.728	0.643	0.000
0.1	0.407	0.593	0.477	0.000
0.05	0.077	0.367	0.343	0.003

Table 5: TNR on the 'Make circles' dataset for SVM, GP, RF and LR classifiers with varying minority class proportion

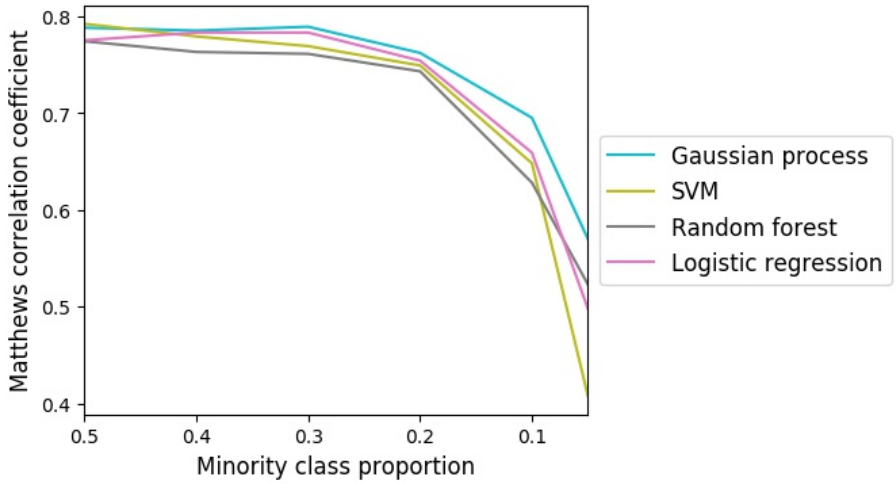


Figure 20: MCC with varying minority class proportion on the 'Linearly separable' dataset for SVM, GP, RF and LR classifiers

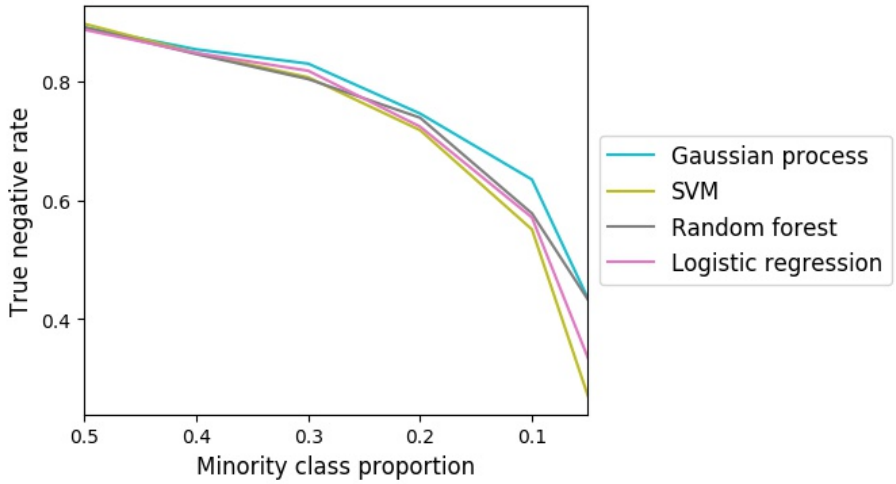


Figure 21: TNR with varying minority class proportion on the 'Linearly separable' dataset for SVM, GP, RF and LR classifiers

Considering the above, GP classification may be considered the best classifier for class-imbalanced linearly separable data.

4.4 Results summary

The results show that on three different synthetic datasets of 500 observations each, the MCC indicates that compared to SVM, RF and LR classifiers, the GP classifier handles class-imbalanced data particularly well.

GP classification is consistently the best-performing classifier on all three of the datasets when the minority class proportion is 0.4, 0.2 and 0.1[15].

[15]The RF classifier attains the same MCC as the GP classifier on the 'Make moons' data at 0.1 minority class proportion.

Minority class proportion	SVM	GP	RF	LR
0.5	0.792	0.788	0.774	0.775
0.4	0.779	0.785	0.763	0.783
0.3	0.769	0.789	0.761	0.783
0.2	0.749	0.762	0.743	0.754
0.1	0.648	0.695	0.628	0.659
0.05	0.408	0.570	0.523	0.498

Table 6: MCC on the 'Linearly separable' dataset for SVM, GP, RF and LR classifiers with varying minority class proportion

Minority class proportion	SVM	GP	RF	LR
0.5	0.897	0.891	0.891	0.887
0.4	0.848	0.854	0.846	0.848
0.3	0.807	0.830	0.804	0.818
0.2	0.718	0.746	0.739	0.724
0.1	0.551	0.635	0.578	0.571
0.05	0.271	0.435	0.433	0.335

Table 7: TNR on the 'Linearly separable' dataset for SVM, GP, RF and LR classifiers with varying minority class proportion

At the most extreme minority class proportion of 0.05, GP classification performs best on the 'Make circles' and 'Linearly separable' datasets.

Considering only the non-linear datasets ('Make moons' and 'Make circles'), GP classification yields the highest MCC when the classes are equally represented and for when the minority class proportion is 0.4, 0.2 and 0.1.

On the 'Make circles' dataset—whose data points are particularly difficult to classify—GP classification attains the highest MCC for every one of the tests.

See Section A in the appendix for illustrations of the behaviour of each classifier on the three datasets with and without the presence of class imbalance.

5 Conclusion

5.1 Concluding remarks

This paper investigates the extent to which a GP classifier is able to classify binary class-imbalanced data. Particular emphasis is placed on the classification accuracy of minority class observations, although the overall classification performance is considered.

The findings indicate that under class imbalance, GP classification does indeed perform well relative to the other techniques.

This has implications for a number of practical applications in which binary classification is used to make decisions and where class imbalance is unavoidable. These applications include credit scoring and fraud detection. The results strongly suggest that in this kind of setting, the use of a GP classifier may

imply fewer misclassifications of minority class observations and increased overall classification accuracy.

This translates to fewer defaulters being classified as non-defaulters and fewer fraudulent transactions being classified as genuine. It follows from this that the total cost related to misclassifications is reduced. This is done while defaulters are still correctly classified as defaulters and genuine transactions are still correctly classified as genuine transactions.

Explicit probability values can be obtained from a GP classification model, and so one may determine, say, the probability that a certain transaction is fraudulent or the probability that a certain entity is not creditworthy. This is a great advantage.

The predictive power and versatility of GP classification comes with one limitation, however: GPs do not scale well to datasets in which there are a large number of dimensions or observations, and so their computational cost is particularly high. With the advent of quantum computing, though, this may soon no longer be a limitation.

With its robustness to class imbalance, GP classification should be considered in practice for creating models from class-imbalanced datasets.

5.2 Limitations and future work

This paper is limited in that the datasets used in the experiments are artificial, and there are only two features associated with each data point. Although this allows for us to visualise how the classifiers behave, the conclusions reached in this paper would not necessarily hold true if there were a greater number of attributes for each data point. Therefore, in future work, classifier performance should be evaluated on real-world datasets whose observations exist in higher dimensional feature space.

Over-sampling minority class observations and under-sampling majority class observations could also be investigated to determine whether resulting models are better able to distinguish observations of each class.

Then, the performance of unary and binary classification could be compared, to establish whether either is better suited to class imbalance.

Finally, the computational issues of GPs could be investigated. Alternative (possibly faster) approximation algorithms could be tested in order to resolve this problem to some extent.

References

[1] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. Optimal classifier for imbalanced data using Matthews correlation coefficient metric. *PLOS ONE*, 12(6):1–17, 06 2017.

[2] Iain Brown and Christophe Mues. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3):3446 – 3453, 2012.

[3] A. de Waal, C. M. van der Walt, and E. Rademeyer. Low default credit scoring using two-class non-parametric kernel density estimation. In *Proceedings of the 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, November 2016.

[4] Chris Drummond and Robert C. Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Proceedings of the ICML '03 Workshop on Learning from Imbalanced Datasets*, January 2003.

[5] D.N. Gujarati. *Basic Econometrics*. Economic Series. McGraw Hill, 2003.

[6] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York, 2013.

[7] E.T. Jaynes and G.L. Bretthorst. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[8] P.M. Lee. *Bayesian Statistics: An Introduction*. Wiley, 2012.

[9] David JC MacKay. Introduction to Gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168:133–166, 1998.

[10] K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptative Computation and Machine Learning Series. University Press Group Limited, 2006.

[13] Mohammed Nazib Seidu. Predicting bankruptcy risk: A Gaussian process classification model. Master's thesis, Linköping University, Department of Computer and Information Science, 2015.

Appendix

A Visualisation of classifier behaviour

To compare the unique behaviour of each classifier when class imbalance is present compared to when it is not, Figures 22 and 23 are shown together. In Figure 22, the classes are equally represented and in Figure 23, the minority class proportion is 0.05. In each pane, the MCC is indicated in the bottom right-hand corner and the TNR is indicated in the top left-hand corner.

Class 1 is represented by blue dots and class 0 (which becomes the minority class) is indicated by red dots. The training and testing points are represented by transparent and opaque dots respectively which together comprise the full dataset as they are shown in each 'Input data' pane. Also illustrated are the decision boundaries for each case which indicate the predicted probability of a point belonging to one of the classes if it falls into that region. In the SVM case, explicit probability values are not calculated and so the decision boundaries actually represent distances from the hyperplanes which separate classification regions. The darker the shading in the SVM case, the further away a point is from the hyperplanes[16]. However, the intuition stays the same. Blue shading indicates the decision boundary for class 1, and red for class 0. The darker the colour, the higher the probability.

In these illustrations, the models are fitted only once to one generation of each dataset, and so the performance of the classifiers here cannot be compared to the performance of the classifiers in the 'Results' section; they are shown here purely for interest's sake.

The figures are generated by an adaptation of Scikit-learn's program 'Classifier Comparison'[17] [11].

B SAS logistic regression code and results

The SAS code in Listing 1 is used to perform logistic regression on exactly the same 'Linearly separable' dataset used in the test illustrated in Figure 23.

Figure 24 (generated using the same SAS program) uses a contour plot to show the probability of a point being classified as belonging to class 0. The red shading indicates a probability closer to 1 and the blue shading indicates a probability closer to 0. The red dots are actual class 0 points and the blue dots are actual class 1 points. Only the training data is shown in this figure.

The model is then used to make predictions for the test data, and the resulting confusion matrix is given in Figure 25. The MCC according to this confusion matrix can be calculated as

[16] SVMs work by creating hyperplanes from training data such that the distance from each hyperplane to the nearest set of training points belonging to different classes is maximised. Predictions are then made based on the classification outcomes associated with the various areas enclosed by the hyperplanes.

[17] *http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html*

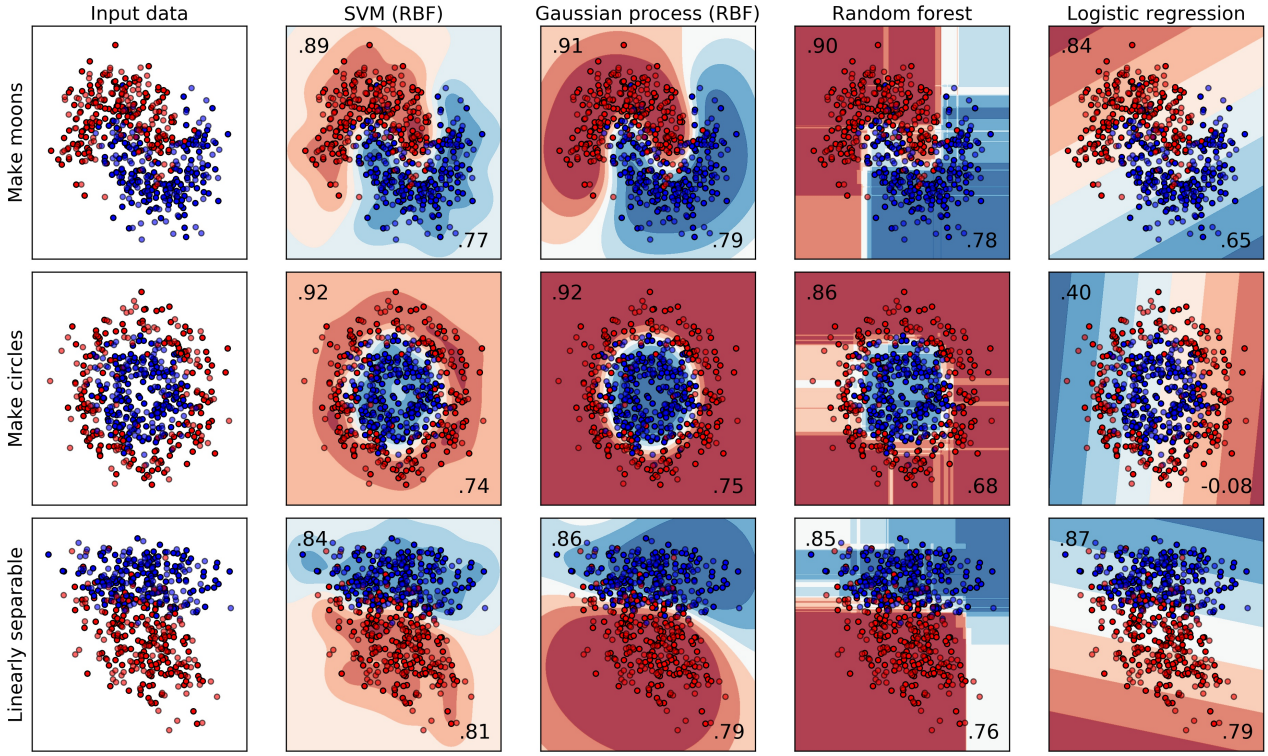


Figure 22: GP, SVM, RF and LR classification on 500 observations with classes equally represented

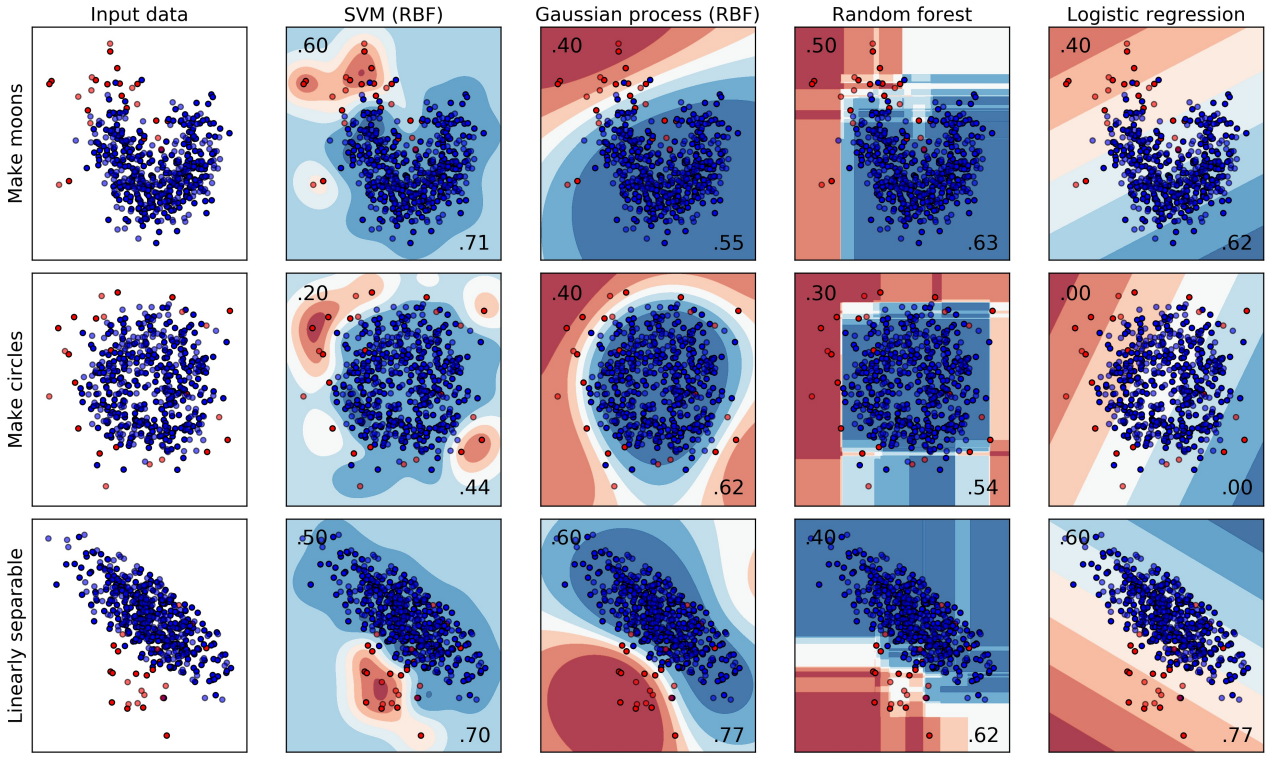


Figure 23: GP, SVM, RF and LR classification on 500 observations with a 5:95 class split

$$\frac{190 \cdot 6 - 4 \cdot 0}{\sqrt{(190+4)(190+0)(6+4)(6+0)}} = 0.77,$$

which is exactly the same as in Figure 23. Therefore, as expected, the LR classification function in Python and in SAS yield the same MCC result on the 'Linearly separable' dataset.

Listing 1: SAS code for logistic regression on the 'linearly separable' dataset

```
filename reffile '/folders/myfolders/research/test_data_for_sas.csv';

proc import datafile=reffile replace dbms=csv out=test_data;
        getnames=yes;
run;

filename reffile '/folders/myfolders/research/train_data_for_sas.csv';

proc import datafile=reffile replace dbms=csv out=train_data;
        getnames=yes;
run;

proc logistic data=train_data outmodel=the_model;
        model y=x1 x2;
        effectplot contour(x=x1 y=x2);
run;

proc logistic inmodel=the_model;
        score data=test_data out=results fitstat;
run;

proc freq data=results;
        table f_y*i_y / nocum nocol nopercent;
run;
```

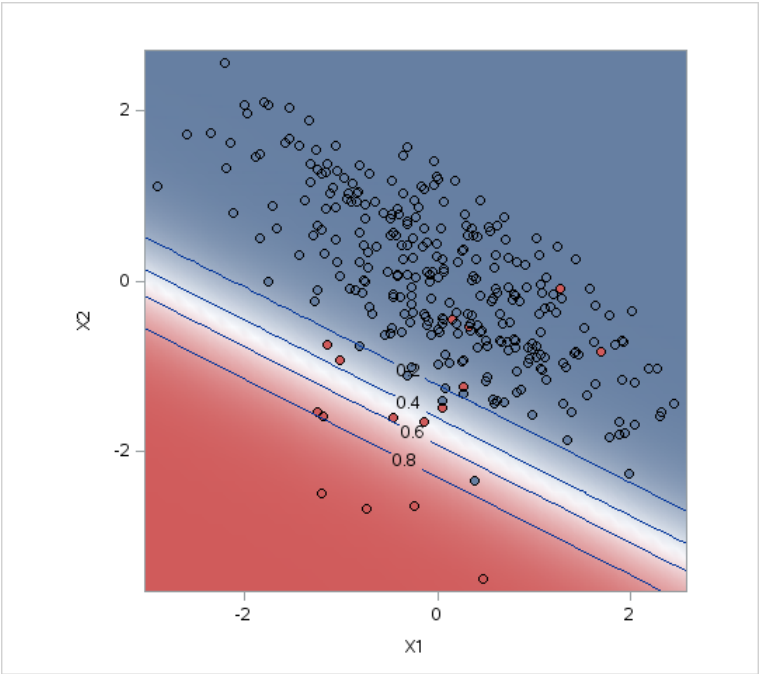


Figure 24: SAS LR classification probability contour plot on 'Linearly separable' dataset with 0.05 minority class proportion

The FREQ Procedure

Frequency
Row Pct

Table of F_y by I_y			
	I_y(Into: y)		
F_y(From: y)	0	1	Total
0	6 60.00	4 40.00	10
1	0 0.00	190 100.00	190
Total	6	194	200

Figure 25: Confusion matrix from SAS LR classification on 'Linearly separable' dataset with 0.05 minority class using PROC FREQ

Repeated measures ANOVA

Chiedza A. Segura 14015902

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr. L. Fletcher

Department of Statistics, University of Pretoria

30 October 2017

Abstract

This is a report on repeated measures analysis of variance, using parametric and non-parametric tests. Repeated measures designs are commonly used in longitudinal studies as opposed to comparing independent groups. Two practical examples, comparing two groups, at three different time periods, will be presented to show a repeated measures experiment as well as a discussion and calculation of effect sizes.

Declaration

I, *Chiedza Ashley Segura,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____

Chiedza Ashley Segura

_____

Dr. L. Fletcher

_____

30/10/2017

Acknowledgements

I would like to extend my heartfelt gratitude to my supervisor, Dr. Lizelle Fletcher, who dedicated a lot of her time assisting me throughout the course of my studies. Without her dedicated support, I would not have soared to these great heights.

I would also like to acknowledge the financial support from the Center for AI Research, Meraka Institute, CSIR.

Contents

List of Figures

List of Tables

1 Introduction

When data is collected in an experiment, a choice can be made between two methods: an independent design or a repeated measures design. In an independent design, also known as between-subjects or between-groups, the independent (qualitative) variable is manipulated by means of different subjects, i.e. various subject groups participate in each experimental condition. In a repeated measures design, also known as within-subjects or within-groups, the independent (qualitative) variable is manipulated using the same participants i.e. the same subject group takes part in each experimental condition. This report is focused on repeated measures design experiments.

In experimental work, including agricultural trials, the accuracy was always increased by repetition and this in turn suggested some form of reliability of results. Numerous layouts for repeated trials, some of which did increase accuracy reasonably well, had been developed from good judgement. Some agronomists who were statistically-minded, had been studying consistency trial data to learn the nature and size of the errors in field trials. There was, however, an absence in articulate theory on the approximation of errors from the results, except in the case of a comparison of only two conditions. Agronomists understood the size of the errors to which the plots of the trials were subjected but the need for repetition was not always recognised. In 1919, the Director of Rothamsted Experimental Station, Sir John Russell, decided to employ a mathematician. He employed Sir Ronald Fisher, an English statistician and biologist, to assist at the Rothamsted Farm in England. [2]

Sir Ronald Fisher directly encountered the problems faced by the other agricultural and biological research employees. The necessity of thorough and suitable methods for estimating the errors became evident to him. In 1922 Fisher first introduced analysis of variance in agricultural trials and it instantly cleared up the problem of estimation of experimental error. The difference between Fisher's experimental designs and existing designs was that Fisher's designs considered validity and efficiency. The first experiment was run at Rothamsted Farm by Thomas Eden, a soil scientist. Repeated measures ANOVA was first used in agricultural trials and now it is being used in a wide variety of fields including the medical field and the engineering field [2].

Repeated measures analysis of variance (ANOVA) is a technique used in statistics to analyse within-subjects or repeated measures designs using parametric tests under certain assumptions. When these assumptions are violated, a non-parametric test i.e. Friedman's test should be used. A parametric test requires the data to originate from one of the probability distributions, usually the normal distribution. For a repeated measures ANOVA to be valid, the assumptions are that the dependent variable is measured on an interval scale and is normally distributed, as well as sphericity (this is the situation where there is equality

of variances between all possible pairs of differences). Non-parametric tests are also called distribution-free tests because fewer or no assumptions are made about the distribution of the type of data on which they can be used. These tests work by ranking the data i.e. getting the lowest score and assigning a rank of 1 to it, then getting the next highest score and assigning it a rank of 2, etc. The ranks are then analysed instead of the actual measurements.

Repeated measures ANOVA, with only one group, is equivalent to a one-way ANOVA, with related observations instead of independent groups, and is an extension of the dependent samples t-test. The categorical independent variable is the repeated measure factor i.e. within-subjects and the quantitative variable, on which each subject is measured, is the dependent variable. Other factors, e.g. treatment vs. control, can also be introduced. It is a test to identify any overall differences between related means. Repeated measures is a cornerstone of scientific research as it proposes a less cumbersome way of assessing the effects of treatments upon subjects [11]. This method lessens the effects of natural variation between individuals upon results and requires less subjects and less resources.

The role of statistics is to uncover the variation in performance, then work out how much is systematic and how much is unsystematic [4]. Field [4] explains that systematic variation is variation caused by the experimenter having influenced all the subjects in one condition but not in another and unsystematic variation is variation caused by unexplained factors that exist between the treatment levels (such as the time of day).

In an independent design, two things may cause the differences between two conditions: the experimental manipulation done on the subjects and/or differences between characteristics of people in the various groups. The differences between characteristics of people are likely to cause significant variation, both within and between each condition.

In a repeated measures design, two things give rise to the differences between two conditions: the influence of the treatment on the subjects and any other factor affecting the way in which a subject performs from one time to the next. The influence of the experimental manipulation is likely to cause significant variation whereas the latter factor's influence should be insignificant.

Therefore, in a repeated measures design, the experimental manipulation effect is more likely to be evident than in an independent design. This is because in a repeated measures design, the unsystematic variation can only be caused by the differences in someone's behaviour at different times. In independent designs, there are differences in natural ability contributing to the unsystematic variation, thus this variation is likely to be much greater each time than if the same subjects had been used. When considering the effect of the experimental manipulation, there will always be other random variation caused by uncontrollable differences between the conditions [4]. In a repeated measures design, this kind of variation is minimised so that the

effect of the experiment is more likely to be brought to light. Thus, repeated measures designs have more capacity to identify effects than independent designs, other things being equal [4].

In this essay, two practical examples will be presented to demonstrate the use of a repeated measures experiment, using real life data from projects from the Department of Statistics' Internal Statistical Consultation Service. The first project is a repeated measures ANOVA with measurements taken over three time periods. The second project is a repeated measures ANOVA with measurements taken over three time periods for two different groups. The statistical software package to be used is SAS.

2 Background Theory

2.1 Model

Following [3] and [8], a response variable is determined for each of the n experimental units at each of the t conditions or time points. Suppose that this is a continuous and normally distributed variable. The general repeated measures ANOVA model is

$$y_{ij} = \mu_{ij} + \pi_{ij} + e_{ij} \tag{1}$$

where

- y_{ij} is the response at condition or time j from i^{th} subject for $i = 1, \ldots, n$, $j = 1, \ldots, t$,
- the mean μ_{ij} (fixed effects since it has a constant value, regardless of the specific individual) for randomly selected individuals from the same population as individual i at condition or time j,
- π_{ij} (random effects since it randomly changes over the population of individuals), the deviation of y_{ij} from μ_{ij} for the ith subject at condition or time j,
- e_{ij} (error terms) is the deviation from $\mu_{ij}+ \pi_{ij}$ for individual i at condition or time j.

The response y_{ij} has mean $\mu_{ij}+ \pi_{ij}$ under assumed recurrences from the same individual.

2.2 One-factor repeated measures ANOVA

For repeated measures obtained from one sample, the model can be written as

$$y_{ij} = \mu + \pi_i + \tau_j + e_{ij} \tag{2}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, t$ where

- y_{ij} is the response from the i^{th} subject at condition or time j,
- the overall mean is μ,
- π_i is the random effect for the i^{th} subject which is the same over all condition or time occurrences,
- τ_j is the fixed effect of condition or time j,
- e_{ij} the random error element at condition or time j specific to the i^{th} subject.

The random effects, π_i, are independent and $\pi_i \sim n(0, \sigma_\pi^2)$, while the random errors e_{ij} are also independently normally distributed with mean equal to zero and standard deviation equal to σ_e. The random effects and errors are independent. It is also assumed that the fixed effects τ_j are constrained to add up to zero.

In relation to the parameters in the model in (1),

- $\mu_{ij} = \mu + \tau_j$,
- $\pi_{ij} = \pi_i$ i.e. it is constant across time.

The observations' variances and the covariances are

- $var(y_{ij}) = \sigma_\pi^2 + \sigma_e^2$,
- $cov(y_{ij'} y_{i'j}) = 0$ for $i \neq i'$,
- $cov(y_{ij}, y_{ij'}) = \sigma_\pi^2$ for $j \neq j'$.

The covariance matrix of the vector $y_i = (y_{i1}, \ldots, y_{it})'$ is therefore

$$\sum = \begin{pmatrix} \sigma_\pi^2 + \sigma_e^2 & & \sigma_\pi^2 \\ & \ddots & \\ \sigma_\pi^2 & & \sigma_\pi^2 + \sigma_e^2 \end{pmatrix} = (\sigma_\pi^2 + \sigma_e^2) \begin{pmatrix} 1 & & \rho \\ & \ddots & \\ \rho & & 1 \end{pmatrix} \tag{3}$$

where $\rho = \frac{\sigma_\pi^2}{\sigma_\pi^2 + \sigma_e^2} = Corr(y_{ij}, y_{ij'})$.

The reasoning behind a repeated measures ANOVA is analogous to that of an independent ANOVA. For independent ANOVA, total variation, which is the sum of squares of the deviations of all the observations from their mean (SS_{Total}), is subdivided into variability between groups (SS_B), and variability within groups (SS_W). The variance within participants is the residual (error) variance (SS_R) and results from individual differences in performance. The experimental effect does not contaminate this variance, since whatever manipulation has been carried out has been done on different people. Dividing these sums of squares by

the suitable degrees of freedom yields a mean sum of squares for between groups (MS_B) and within groups (MS_W).

In a repeated measures ANOVA, the total variability SS_{Total} is also partitioned into conditions variability SS_C and within-groups variability SS_W as shown in Figure 1. The experimental effect in a repeated measures ANOVA appears in the within-participant variance rather than in the between-group variance. When manipulation is carried out on the same people, two things make up the within-participant variance: effect of the manipulation and the individual differences in performance at the different conditions or time periods. Therefore, some of the within-participant variation comes from the effects of the experimental manipulation. Any variation that cannot be explained by the manipulation must be due to random factors, not linked to the experimental manipulations, since the same manipulation is carried out on every participant within a particular condition. Furthermore, each subject is treated as a block, in other words each subject becomes a level of a factor called subjects. Therefore, the advantage of a repeated measures ANOVA is that it further divides this variability within groups into subject variability (SS_S) and error variability (SS_R). The ability to subtract SS_S will lead to a reduced SS_R i.e.

$$SS_R = SS_W - SS_S.$$

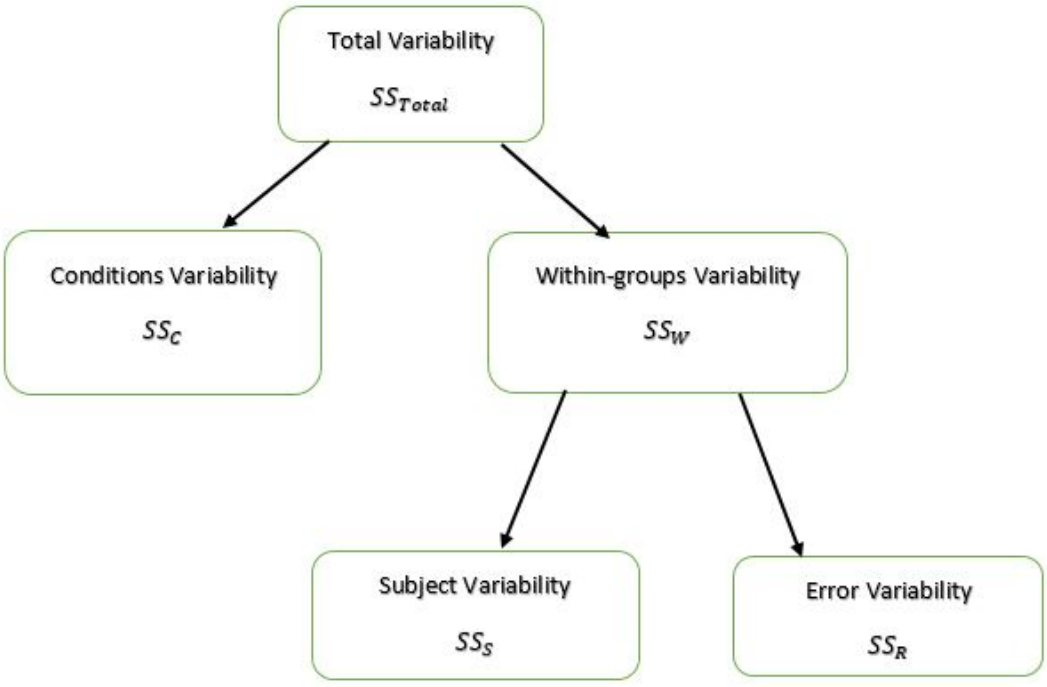


Figure 1: Total Sum of Squares partition

Table 1 is the ANOVA table for the repeated measures model.

Source	Sum of Squares (SS)	degrees of freedom (d.o.f)	Mean Squares (MS)
Conditions(Time)	SS_C	$t-1$	$MS_C = \frac{SS_T}{t-1}$
Subjects	SS_S	$n-1$	$MS_S = \frac{SS_S}{n-1}$
Residual	SS_R	$(n-1)(t-1)$	$MS_R = \frac{SS_R}{(n-1)(t-1)}$
Total	SS_{Total}	$nt-1$	

Table 1: One sample repeated measures ANOVA

The sum of squares (SS) in Table 1 are calculated using the observations y_{ij}, the overall mean

$$\bar{y}_{..} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{t} y_{ij}}{nt},$$

the means for each subject over time

$$\bar{y}_{i.} = \frac{\sum_{j=1}^{t} y_{ij}}{t},$$

and the means at each time point over all subjects

$$\bar{y}_{.j} = \frac{\sum_{i=1}^{n} y_{ij}}{n}.$$

The different sum of squares are defined as follows:

- $SS_C = \sum_{i=1}^{n}\sum_{j=1}^{t}(\bar{y}_{.j} - \bar{y}_{..})^2 = n\sum_{j=1}^{t}(\bar{y}_{.j} - \bar{y}_{..})^2$,
- $SS_S = \sum_{i=1}^{n}\sum_{j=1}^{t}(\bar{y}_{i.} - \bar{y}_{..})^2 = t\sum_{i=1}^{n}(\bar{y}_{i.} - \bar{y}_{..})^2$,
- $SS_R = \sum_{i=1}^{n}\sum_{j=1}^{t}(y_{ij} - \bar{y}_{i.} - \bar{y}_{.j} + \bar{y}_{..})^2$.

2.2.1 Repeated measures ANOVA hypothesis

The repeated measures ANOVA investigates if any differences exist between related population means. The null hypothesis

$$H_0 : \mu_1 = \mu_2 = \ldots = \mu_t \tag{4}$$

where μ is the population mean and t the number of related groups, is tested against the alternative hypothesis stating that at least one of the population means is different from the other. A repeated measures ANOVA is an omnibus statistical test and therefore will not tell where the differences between groups lie. In other words, it determines whether the explained variance in a data set is significantly larger than the unexplained

variance, overall. Should the repeated measures ANOVA be larger, then post hoc tests can be carried out to show exactly where the differences arise [8].

2.2.2 F-statistic

As in independent ANOVA, an F-ratio that compares the variation size brought about by the experimental manipulations to the variation size brought about by random factors is used. The only difference between the F-ratio of the independent ANOVA and repeated measures ANOVA is how we calculate the variances. It has been argued, in 1970 by [9], that two F-ratios can be used to gauge comparisons of treatment i.e. F' and F''. F' is the ratio obtained from the mean squares depending on the comparison being investigated and the specific error term for the comparison of interest. F'' is obtained from the total mean squares for all comparisons of repeated measures [4]. F' is usually used and is calculated as

$$F' = \frac{MS_C}{MS_R}. \tag{5}$$

The reduction in the size of the residual variability due to the division of the within-participants variability leads to an increase in the value of F' since

$$SS_R = SS_W - SS_S.$$

Because the between participants' variability has been removed, the new SS_R reflects only individual variability for each condition or time period. The power of the test to detect significant differences between means is thus increased. This is the major advantage of running a repeated measures ANOVA. If the assumptions of the model are met, then F' has a $F_{t-1,(n-1)(t-1)}$ distribution if H_0 is true. If the variance caused by the experimental manipulations is large relative to the variation caused by random factors, then a large F-value is obtained, and we can thus conclude that the observed results are unlikely to have occurred if there was no effect in the sample [4].

2.3 Multiple factor repeated measures ANOVA

For repeated measures obtained from multiple samples, suppose that from s groups of subjects, measurements are acquired at t time points [3] . Let n_h represent the number of subjects in the h^{th} group , and let $n = \sum_{h=1}^{s} n_h$. There are several possible models for this case, which all result in the same ANOVA table and

the simplest is

$$y_{hij} = \mu + \gamma_h + \tau_j + (\gamma\tau)_{hj} + \pi_{i(h)} + e_{hij}. \tag{6}$$

In the model,

- y_{hij} is the response at condition or time j from the i^{th} subject in the h^{th} group for $h = 1, \ldots, s$, $i = 1, \ldots, n_h$ and $j = 1, \ldots, t$,
- the overall mean is μ,
- γ_h is the fixed effect of the h^{th} group and the fixed effects γ_h are constrained to add up to zero,
- τ_j is the fixed effect of condition or time period j and the fixed effects τ_j are constrained to add up to zero,
- $(\gamma\tau)_{hj}$ is the fixed effect for the interaction of the j^{th} time with the h^{th} group and the fixed effects of the interaction are constrained to add up to zero,
- $\pi_{i(h)}$ are random effects for the i^{th} subject in the h^{th} group,
- e_{hij} are the random error terms.

According to [3], assume the random effects, $\pi_{i(h)}$, are independent and $\pi_{i(h)} \sim N(0, \sigma_\pi^2)$. The random errors terms, e_{hij}, are independent and normally distributed with mean zero and variance σ_e^2. In relation to the parameters in the model in (1),

- $\mu_{ij} = \mu + \gamma_h + \tau_j + (\gamma\tau)_{hj}$,
- $\pi_{ij} = \pi_{i(h)}$,
- $e_{ij} = e_{hij}$.

Table 2 is the multiple sample repeated measures ANOVA table.

Source	Sum of Squares	degrees of freedom	Mean Squares
Group	SS_G	$s-1$	$MS_G = \frac{SS_G}{s-1}$
Subjects(Group)	$SS_{S(G)}$	$n-s$	$MS_{S(G)} = \frac{SS_{S(G)}}{n-s}$
Time	SS_T	$t-1$	$MS_T = \frac{SS_T}{t-1}$
Group×Time	SS_{GT}	$(s-1)(t-1)$	$MS_{GT} = \frac{SS_{GT}}{(s-1)(t-1)}$
Residual	SS_R	$(n-s)(t-1)$	$MS_R = \frac{SS_R}{(n-s)(t-1)}$
Total	SS_{Total}	$nt-1$	

Table 2: Multiple sample repeated measures ANOVA

The following decomposition of the deviations of each observation about the overall mean is the basis of the sums of squares:

$$y_{hij} - \bar{y}_{...} = (\bar{y}_{h..} - \bar{y}_{...}) + (\bar{y}_{hi.} - \bar{y}_{h..}) + (\bar{y}_{..j} - \bar{y}_{...}) + (\bar{y}_{h.j} - \bar{y}_{h..} - \bar{y}_{..j} + \bar{y}_{...}) + (\bar{y}_{hij} - \bar{y}_{h.j} - \bar{y}_{hi.} + \bar{y}_{h..}), \quad (7)$$

where the overall mean is

$$\bar{y}_{...} = \frac{\sum_{h=1}^{s} \sum_{i=1}^{n_h} \sum_{j=1}^{t} y_{hij}}{nt},$$

the mean for the h^{th} group is

$$\bar{y}_{h..} = \frac{\sum_{i=1}^{n_h} \sum_{j=1}^{t} y_{hij}}{n_h t},$$

the mean at time j is

$$\bar{y}_{..j} = \frac{\sum_{h=1}^{s} \sum_{i=1}^{n_h} y_{hij}}{n},$$

the mean for the h^{th} group at time j is

$$\bar{y}_{h.j} = \frac{\sum_{i=1}^{n_h} y_{hij}}{n_h},$$

and the mean for the i^{th} subject in the h^{th} group is

$$\bar{y}_{hi.} = \frac{\sum_{j=1}^{t} y_{hij}}{t}.$$

The sums of squares are then defined as:

- $SS_G = \sum_{h=1}^{s} \sum_{i=1}^{n_h} \sum_{j=1}^{t} (\bar{y}_{h..} - \bar{y}_{...})^2 = t \sum_{h=1}^{s} n_h (\bar{y}_{h..} - \bar{y}_{..})^2,$
- $SS_{S(G)} = \sum_{h=1}^{s} \sum_{i=1}^{n_h} \sum_{j=1}^{t} (\bar{y}_{hi.} - \bar{y}_{h..})^2 = t \sum_{h=1}^{s} \sum_{i=1}^{n_h} (\bar{y}_{hi.} - \bar{y}_{h..})^2,$
- $SS_T = \sum_{h=1}^{s} \sum_{i=1}^{n_h} \sum_{j=1}^{t} (\bar{y}_{..j} - \bar{y}_{...})^2 = n \sum_{j=1}^{t} (\bar{y}_{..j} - \bar{y}_{...})^2,$
- $SS_{GT} = \sum_{h=1}^{s} \sum_{i=1}^{n_h} \sum_{j=1}^{t} (\bar{y}_{h.j} - \bar{y}_{h..} - \bar{y}_{..j} + \bar{y}_{...})^2,$
- $SS_R = \sum_{h=1}^{s} \sum_{i=1}^{n_h} \sum_{j=1}^{t} (y_{hij} - \bar{y}_{h.j} - \bar{y}_{hi.} + \bar{y}_{h..})^2.$

2.3.1 F-statistic

When testing for variation among groups, the F-statistic is

$$F = \frac{MS_G}{MS_{S(G)}}. \tag{8}$$

The assumption that there is equality in the within-group covariance matrices is required. Generally, this is a requirement for all tests of between-subjects effects. When testing for variation among time points, the F-statistic is

$$F = \frac{MS_T}{MS_R}. \tag{9}$$

Similarly, when testing for the significance of the interaction of group×time , the F-statistic is

$$F = \frac{MS_{GT}}{MS_R}. \tag{10}$$

The assumption of equality in the within-group covariance matrices is required for both tests, and that the condition of sphericity is met. Generally, this is a requirement for all tests of within-subjects effects [3].

Another repeated measures model for this case includes an extra random effect for the subject×time interaction. The assumption is that this effect is uncorrelated with the random subject effect. The sum of squares and test statistics are identical though the expected mean squares for this model are different from those displayed in Table 2.

2.4 Sphericity condition

For a repeated measures ANOVA, the sphericity assumption, also referred to as circularity, is analogous to the homogeneity of variance assumption in between-groups ANOVA i.e. it is assumed that the variation within treatment levels is reasonably alike and no two conditions are any more dependent than the others. To determine sphericity, the differences between pairs of scores is calculated in all combinations of the experimental conditions, then the differences of these variances are calculated. If the variances are equal or nearly equal, then the sphericity condition is satisfied.

Mauchly's test, which tests the null hypothesis that there is equality of variances of the differences between conditions, can be obtained in SAS, as well as in most other statistical packages [3]. If the test statistic is significant, then the conclusion is that the differences between the variances of differences are significant and so there is reason to be suspicious of the F-ratios. However, in the case of small samples, Mauchly's test has a low power [3]. It has been proved to be susceptible to deviations from normality and outliers [3].

The condition for compound symmetry is sufficient for F', but it is unnecessary. It is a special case of sphericity, a more general case, under which the F-test is valid [3]. The F-statistic has an approximate $F_{\varepsilon(t-1),\varepsilon(t-1)(n-1)}$ distribution when the sphericity condition is not met, where ε is a function of the covariance matrix. If the sphericity assumption is violated, several corrections on the F-ratio can be carried out. One of the corrections is adjusting the degrees of freedom of the F-statistic by means of the lower bound for ε. When $\varepsilon = \frac{1}{t-1}$, then the $F_{\varepsilon(t-1),\varepsilon(t-1)(n-1)}$ distribution becomes the $F_{1,(n-1)}$ distribution. However, this correction is too conservative [3]. There are two other corrections which are commonly used, namely the Greenhouse-Geisser correction and the Huynh-Feldt correction. These two estimates produce a correction factor that is related to the degrees of freedom which are used to evaluate the observed F-ratio. The Greenhouse-Geisser correction, $\hat{\varepsilon}$, alters between $\frac{1}{t-1}$ (where t is the number of time periods or conditions) and 1. As $\hat{\varepsilon}$ gets closer to 1, the variances of differences become more homogeneous and hence the closer the data are to being spherical. Huynh-Feldt reported that when the Greenhouse-Geisser estimate is larger than 0.75 and $n < 2t$, too many false null hypotheses fail to be rejected [4]. Therefore, Huynh and Feldt suggested a less conservative correction

$$\tilde{\varepsilon} = min\left(\frac{n(t-1)\hat{\varepsilon} - 2}{(t-1)(n-1-(t-1)\hat{\varepsilon})}\right). \tag{11}$$

$\tilde{\varepsilon}$ is derived from the unbiased estimators of ε and has less bias than $\hat{\varepsilon}$. It can be proved that $\tilde{\varepsilon} \geq \hat{\varepsilon}$ [3]. The estimate $\hat{\varepsilon}$ works better for $\varepsilon \leq 0.5$ and $\tilde{\varepsilon}$ works better when $\varepsilon \geq 0.75$. ε is unknown in practice. However, [7] reported that the Huynh-Feldt estimate overestimates sphericity. Therefore, [10] proposed adjusting the degrees of freedom of the two estimates by average of the two. Girden [5] suggested that $\tilde{\varepsilon}$ should be used when the estimates of sphericity are greater than 0.75, and $\hat{\varepsilon}$ should be used when sphericity estimates are less than 0.75 or there is completely no knowledge about sphericity. Greenhouse and Geisser proposed the following method be used for repeated measures ANOVA.

1. Assume that the repeated measures ANOVA assumptions are met and perform the univariate F-test.
2. Do not reject H_0 if the test is not significant.
3. Perform the conservative test using $\varepsilon = \frac{1}{t-1}$, if the test is significant. This will lead to the F-distribution with 1 and $n-1$ degrees of freedom.

- Reject the H_0 if the conservative test is significant,
- Then estimate ε and conduct an approximate test if the conservative test is not significant.

Another alternative option when the data violates the sphericity condition is the use of multivariate analysis of variance (MANOVA) test statistics, since they do not depend on the sphericity assumption. These multi-

variate test statistics can be produced automatically by the repeated measures procedures in many statistical packages [3].

2.5 Non-parametric test: Friedman's test

When there is only one independent variable and the normality assumption is violated or when measurements are made on an ordinal scale, a non-parametric test such as Friedman's ANOVA should be used, which is based on ranks. For factorial repeated measures designs, there is no non-parametric alternative test to use.

When there are more than two conditions or time periods and the same subjects are used in all conditions, Friedman's ANOVA is used to test for differences between these conditions. If some assumption of the parametric tests have been violated, then Friedman's ANOVA can be a useful way around the problem. The ranks are analysed instead of the actual measurements. Once all the data has be ranked, the total of the ranks is computed for each condition. R_i represents the total of the ranks, where i represents the specific condition or time period. When the sum of ranks for each condition or time period has been determined, then the statistic, F_r, is computed as

$$F_r = [\frac{12}{nt(t+1)}\sum_{i=1}^{t} R_i^2] - 3n(t+1) \tag{12}$$

where

- R_i is the total of the ranks for each condition or time period,
- n is the sample size,
- t is the number of time periods or conditions.

When the number of subjects tested is relatively large, i.e. bigger than ten, this test statistic has a χ^2 distribution with $t-1$ degrees of freedom [11].

2.6 Effect size

In a repeated measures ANOVA, as in most statistical tests, there is a need to measure the effect size, i.e. the magnitude or size of a treatment. According to [1], investigators should be encouraged to determine, produce and report effect size statistics regularly in an empirical report. Many effect size statistics have been established for repeated measures ANOVA. These include omega squared (ω^2), eta squared (η^2), partial eta squared (η_P^2) and generalized eta squared (η_G^2). The statistic commonly used in the ANOVA literature is η^2,

which is the ratio of the conditions variation to total variation. Therefore,

$$\eta^2 = \frac{SS_C}{SS_{Total}}. \tag{13}$$

The statistic η^2 is adequate with a one-way ANOVA, but when there is more than one source of variation in an ANOVA, it may become problematic. Usually η_P^2 is given as a solution to this problem. In factorial designs, η_P^2 takes away some of the factorial effects from the denominator. With a simple $A \times B$ factorial design, the total variation SS_{Total} is divided into four elements: SS_A, SS_B, SS_{AB} and the error term SS_R. Thus

$$\eta_P^2 = \frac{SS_C}{SS_C + SS_R} \tag{14}$$

where SS_C denotes the conditions i.e. either SS_A, SS_B , or SS_{AB}.[1]

For a repeated measures design which only has one within-subjects variable and no between-subjects variables, the total variability is divided into three elements: SS_S (the subjects variation), SS_P and SS_{Ps}. SS_S shows the portion of total variation that can be explained by knowledge of the specific subject. This fraction will be greater if the repeated scores correlate more within subjects. Each of the subjects signify a level of the factor. The error term will be smaller if scores correlate more within subjects. This is where the reputation for increased power of repeated measures designs emanates from. η_P^2 will generally be larger than η^2 since the subject effect would be eliminated from the denominator of η_P^2. [1]

The difference between η_G^2 and, η_P^2 and η^2 is in the denominator. η^2 includes all component sum of squares while η_G^2 and η_P^2 include only some of them, although η_G^2 includes more than η_P^2. η_G^2 assumes a traditional univariate ANOVA method, and not multivariate or multilevel methods to designs which involve repeated measures. η_G^2 is estimated as

$$\eta_G^2 = \frac{SS_C}{\delta \times SS_C + \sum_{measured} SS_{measured}} \tag{15}$$

where $SS_{measured}$ is the sum of squares due to individual differences.

The statistic η^2 and other measures are usually omnibus tests which are regarded as not specific enough to address the real concerns of researchers [1]. Bakeman [1] suggests that the generalized eta squared (η_G^2) is a generally more useful statistic for effect sizes as it allows comparability across between-subjects and within-subjects designs. He also argues that it can easily be determined from information given by standard statistical packages.

Field [4] suggests that the best measure for effect sizes for repeated measures design is omega squared

(ω^2) which is similar to the independent ANOVA. However, the equations for an independent design cannot be used with repeated measures data. The effect size will be slightly overestimated if the same equations are used on repeated measures data. The equation for omega squared for repeated measures ANOVA is

$$\omega^2 = \frac{[\frac{t-1}{nt}(MS_S - MS_R)]}{MS_R + \frac{MS_C - MS_R}{t} + [\frac{t-1}{nt}(MS_S - MS_R)]} \tag{16}$$

where

- t is the number of conditions in the experiment,
- n is the number of participants,
- MS_R is the mean square for residuals.

3 Applications

3.1 Single and multiple factor repeated measures ANOVA: SANDF data

3.1.1 Problem Statement

The number of female soldiers participating in the armed forces worldwide has increased dramatically. Approximately 10% to 20% of most armed forces around the world are made up by women. In the South African military, women's roles have evolved from the 1970s, when women could volunteer to serve in support functions of the South African military, to allowing women access to the full spectrum of career opportunities, including combat mustering. Women are more representative than men in the medical mustering of the South African National Defence Force (SANDF), as well as in the armour corps, artillery, infantry, combat navy, and as aircrew. A sharp increase in the number of women attending training and management courses at the SANDF's training institutions has been reported. Additionally, approximately 10% of the SANDF personnel deployed in peacekeeping operations are women, the highest percentage of women armed forces of all the African countries that contribute peacekeeping troops.

Many militaries still believe that women may compromise combat effectiveness based on physiological and psychological grounds and so they still do not allow them in direct frontline combat. In most militaries, including the SANDF, women are held to lower physical standards than men, meaning they must complete fewer exercise sets and are allowed longer run times during fitness tests. The physical limitations imposed by the female soldiers' lower muscle mass, higher percentage fat, lower aerobic and anaerobic capacity, and higher incidence of training injuries have been used to justify this practice. [12]

However, the argument still stands that if men and women are subject to the same absolute workloads, it may not be reasonable to lower one group's fitness standard. To aid in the integration of women into the SANDF, gender-mixed training has been adopted. Gender-integrated training has also been implemented by other militaries and supported by some. This in despite of growing evidence that mixed basic military training (BMT) is a greater risk factor for overuse injury in women. Considering a large amount of human and financial capital being invested into integrating women into militaries around the world, the purpose of this study was to determine gender differences before, during (12 weeks) and after a 20-week mixed BMT course and to determine if the course assisted in significantly reducing these differences. [12]

3.1.2 Study design

A total of 191 soldiers (115 male: mean age = 21.0 ± 1.1 year; 76 female: mean age = 20.5 ± 1.2 year) completed the BMT course and all anthropometric, physical fitness, explosive power, and hand grip strength measurements were taken. In this report only physical fitness was evaluated using the score of the standardised SANDF_PT test. The test comprised of five components namely a 2.4km run(time), maximum number of sit ups and push ups completed in 2 minutes, a shuttle run test and a 4km walk(time). Repeated measures analysis of variance was used to model BMT data with main effects for gender comparison between males and females, and time repeated measures effect for evaluation of differences between weeks 1, 12, and 20 of BMT, as well as an interaction effect for differences in changes over time for males and females.

The physical fitness measurements were taken at baseline (week 1=W1PT), after week 12 (W12PT) and after week 20 (W20PT). Alpha was set at $\alpha \leq 0.05$.

3.1.3 Results over the three time periods

Table 3 illustrates the descriptive statistics for the SANDF-PT test across the three time periods.

n=191	Week 1	Week 12	Week 20
Mean	54.005	84.346	83.718
Median	56.900	85.700	85.300
Standard deviation	17.260	8.505	10.582

Table 3: Descriptive statistics for SANDF-PT

	Females(76)			Males(115)		
	Week 1	Week 12	Week 20	Week 1	Week 12	Week 20
Mean	45.562	81.989	81.459	59.584	85.903	85.201
Median	44.900	84.000	83.550	61.100	86.400	86.300
Standard deviation	17.606	10.410	11.501	14.617	6.567	9.693

Table 4: Descriptive statistics for males and females

Table 4 shows the descriptive statistics for the SANDF-PT test by gender

Friedman test

From the SAS output, it follows that the data violate the assumption of normality since all the normality test statistics were significant (less than 0.05). This led to the use of the non-parametric test mentioned in Section 2.5. Friedman's test was obtained in SAS using the PROC FREQ procedure. The PROC FREQ procedure produces the test statistic for the Cochran-Mantel-Haenszel test. This is equivalent to the χ^2 statistic from the Friedman's test which is used to evaluate the null hypothesis that there are no differences with respect to SANDF-PT across the three time periods. The row lablelled $Row\,Means\,Scores\,Differ$ is the important row and the column labelled χ^2 value gives the value equivalent to Friedman's test statistic which is 270.105 with 2 degrees of freedom (cf. Fig. 11). This leads to the rejection of H_0 at the 5% level of significance ($p<0.0001$). We therefore conclude that there was a significant difference between the SANDF-PT levels.

Since the result is significant, it is necessary to investigate further which time periods differ from each other by performing post hoc analyses. The mean ranks are shown in Table 5 (cf. Appendix Fig 11).

Variable	Number of observations	Mean
time1 (W1PT)	191	115.450
time2 (W12PT)	191	374.770
time3 (W20PT)	191	370.780

Table 5: Mean ranks of Friedman's test

There are several ways to perform multiple non-parametric post hoc tests. We decided to use Wilcoxon signed rank tests and correcting for the number of tests done. One way to correct for the number of tests performed is using the Bonferroni correction where the significance level is adjusted by dividing by the number of tests. So instead of using 0.05 to compare the p-values, 0.05/3=0.0167 was used as the level of significance.

	median	t-statistic	p-value
W1PT-W12PT	-29.400	-28.243	<0.0001
W1PT-W20PT	-28.600	-27.164	<0.0001
W12PT-W20PT	0.300	0.893	0.379

Table 6: Post hoc tests

Table 6 shows the results from the Wilcoxon sign-rank test. The null hypothesis being tested is that the difference score between two time points is zero. There is a significant difference between measurements taken at baseline and week 12 since the p-value<0.0001 which is less than the adjusted level of significance 0.0167

(Bonferroni correction). Therefore the null hypothesis is rejected. The measurements taken at baseline and week 20 are also significantly different and so the null hypothesis is rejected. The measurements taken during week 12 and after week 20 are not significantly different since p-value=0.379. This means that the study done over 20 weeks would be similar to that done over 12 weeks and so it could have been shortened.

Repeated measures ANOVA

Since the data set is large, it can be argued that the central limit theorem can be used which states that the means follow an approximate normal distribution, even though the data used violate the assumption of normality. In Section 2.4, it was mentioned that SAS produces Mauchly's test which checks for the violation of the assumption of sphericity. This test should not be significant for the sphericity assumption to hold. Table 7 displays the results for Mauchly's test for the SANDF-PT test (cf. Appendix Fig. 13).

Variables	d.o.f	Mauchly's Criterion	Value	p-value
Orthogonal Components	2	0.739	57.114	$<$0.0001

Table 7: Sphericity test

The p-value for this data is less than 0.0001, which is less than the level of significance of 0.05. Therefore, the assumption that the variances of the differences between levels are equal is rejected i.e. the sphericity assumption has been violated.

It was also mentioned in Section 2.4 that SAS produces two corrections based on the estimates of sphericity supported by Greenhouse and Geisser, and Huynh and Feldt. If the Greenhouse-Geisser correction is closer to 1, the more homogenous the variances of difference and hence the more spherical the data becomes. Table 8 on the next page shows the estimates for the corrections. Since the Greenhouse-Geisser estimate is 0.813 and is close to 1, it can be assumed that the variances of differences are homogenous. Therefore the data is close to being spherical.

Correction	Estimate
Greenhouse-Geisser	0.813
Huynh-Feldt	0.819
Lower Bound	0.500

Table 8: Sphericity Corrections

The results of the ANOVA for the within-subjects are given in Table 9 (cf. Appendix Fig. 13). There is a sum of squares (SS_T) for the repeated-measures effect of time which tells us how much of the total variation is explained by the effect of the experiment. There is a residual term (SS_R), which is the amount of unexplained variation across the conditions of the repeated measures variable, which is time in this case.

Source	d.o.f	SS	MS	F-value	p-value	adj p-value Greenhouse-Geisser	adj p-value Huynh-Feldt
Time	2	114842.757	57421.378	634.12	<0.0001	<0.0001	<0.0001
Residual(time)	380	34410.255	90.553				

Table 9: Repeated measures ANOVA for within-subjects

The degrees of freedom for the effect of time is 2 i.e. $(t-1)$, where t is the number of repeated measures effects, and the degrees of freedom for the residuals is 380 i.e. $((n-1)(t-1))$ where n is the number of participants. The F-ratio is determined by dividing the mean squares for the experimental effect ($\frac{SS_T}{t-1}$) by the mean squares of error ($\frac{SS_R}{(n-1)(t-1)}$). SAS gives the exact significance level for the F-ratio. Since the p-value is less than 0.0001 which is less than 0.05, the F-ratio is significant. Therefore, the null hypothesis that there are no differences with respect to SANDF-PT across the three time periods is rejected and the conclusion is that there is a significant difference between the times.

Since initially, before the sphericity corrections, the assumption of sphericity had been violated, the F-ratio may be inaccurate. SAS produces adjusted p-values for the Greenhouse-Geisser and Huynh-Feldt corrections as illustrated in Table 9 above. They both show that the F-ratios for the corrections are significant. Therefore, the conclusion remains the same.

In Section 2.4 it was mentioned that another option to use when there is violation of sphericity is the multivariate analysis of variance (MANOVA) test statistics as they do not depend on the data being spherical. The column displaying the significance values (p-values) in Table 10 on the next page shows that the multivariate tests are significant since the p-values are less than 0.0001 (all < 0.05). Therefore, it supports the conclusion that there are significant differences between the three different periods.

Effect	Statistic	Value	F-value	Hypothesis d.o.f	Residual d.o.f	p-value
time	Wilks' Lambda	0.181	427.910	2	189	<0.0001
	Pillai's Trace	0.819	427.910	2	189	<0.0001
	Hotelling-Lawley Trace	4.528	427.910	2	189	<0.0001
	Roy's Greatest Root	4.528	427.910	2	189	<0.0001

Table 10: MANOVA statistics

The post hoc comparisons of the the three different time levels are shown in Table 11.

Source	time(level)	SS	d.o.f	MS	F-value	p-value
time	1 vs 2	175 828.189	1	175828.189	797.680	<0.0001
	1 vs 3	168 624.751	1	168624.751	737.900	<0.0001
	2 vs 3	75.330	1	75.330	0.800	0.3728
residual	1 vs 2	41880.881	190	41880.881		
	1 vs 3	43418.571	190	43418.571		
	2 vs 3	17931.313	190	17931.313		

Table 11: Contrasts for the three different time periods

The first contrast is the comparison of the baseline measurement, time 1 and measurements taken after 12 weeks, time 2. The F-statistic is significant (p-value<0.0001) which means that time 1 and time 2 differ significantly. The second contrast is the comparison of time 1 and time 3, measurements taken after 20 weeks. The F-statistic is significant (p-value<0.0001), indicating a significant difference in the baseline and time 3 values. The third contrast is the comparison of time 2 and time 3. The F-statistic is not significant (p-value<0.0001) i.e. there are no significant differences between the values of time 2 and time 3.

3.1.4 Effect sizes

The effect size that is going to be calculated is ω^2.

$$\begin{aligned}
\omega^2 &= \frac{[\frac{t-1}{nt}(MS_S - MS_R)]}{MS_R + \frac{MS_C - MS_R}{t} + [\frac{t-1}{nt}(MS_S - MS_R)]} \\
&= \frac{[\frac{3-1}{191\times 3}(57421.378 - 90.553)]}{90.553 + \frac{301.124-90.553}{3} + [\frac{3-1}{191\times 3}(57421.378 - 90.553)]} \\
&= 0.5545
\end{aligned}$$

The difference between the three time periods had a medium effect since $0 \leq \omega^2 \leq 1$.

3.1.5 Repeated measures ANOVA taking gender into account

	d.o.f	Mauchly's Criterion	χ^2-value	p-value
Orthogonal component	2	0.818	37.816	<0.0001

Table 12: Sphericity test

Table 12 shows that Mauchly's test is significant (p-value<0.0001). Therefore, the assumption that the variances of the differences between levels are equal is rejected i.e. the sphericity assumption has been violated.

The two corrections based on the estimates of sphericity advocated by Greenhouse and Geisser, and Huynh and Feldt are used again. Table 13 shows the estimates for the corrections. Since the Greenhouse-

Geisser estimate is 0.846 and is close to 1, it can be assumed that the variances of differences are homogenous. Therefore the data is close to being spherical.

Correction	Estimate
Greenhouse-Geisser	0.846
Huynh-Feldt	0.853

Table 13: Sphericity corrections

Table 22 shows that the F-statistic for gender is significant (p-value$<$0.0001). The null hypothesis that there are no differences between males and females is rejected and so we conclude that that there is a significant difference between the males and females.

Source	d.o.f	SS	MS	F-value	p-value
W1SEX(gender)	1	7174.253	7174.253	27.100	$<$0.0001
Residual	189	50039.245	264.758		

Table 14: Repeated measures ANOVA for between-subjects

The results of the ANOVA for the within-subjects is illustrated in Table 15. In this table, there is an extra term, an interaction term, which is variation caused by the interaction between time and gender (SS_{GT}).

Source	d.o.f	SS	MS	F-value	p-value	adj p-value Greenhouse-Giesser	adj p-value Huynh-Feldt
time	2	117808.452	58904.226	712.690	$<$0.0001	$<$0.0001	$<$0.0001
time$\times$W1SEX	2	3168.228	1584.114	19.170	$<$0.0001	$<$0.0001	$<$0.0001
residual(time)	378	31242.027	82.6509				

Table 15: Repeated measures ANOVA for within-subjects

The degrees of freedom for the effect of time is 2 i.e. $(t-1)$, where t is the number of repeated measures effects, the degrees of freedom for the interaction is 2 i.e. $((s-1)(t-1))$ where s is the number of groups and the degrees of freedom for the residuals is 380 i.e. $((n-1)(t-1))$ where n is the number of participants. Since the p-values for time and the interaction between time and gender are less than 0.0001, then the F-ratios are significant. Therefore, the conclusion is that there is a significant difference between the times and there is a significant difference in the interactions.

The adjusted p-values for the Greenhouse-Geisser and Huynh-Feldt corrections as illustrated in Table 15 on the previous page show that the F-ratios for the corrections are significant. Therefore, the conclusion remains the same.

The MANOVA results displayed in Table 16 below confirm the conclusion that there are significant differences between the times and there are significant differences between the interactions.

Effect	Statistic	Value	F-value	Hypothesis d.o.f	Residual d.o.f	p-value
time	Wilks' Lambda	0.159	498.480	2	188	<0.0001
	Pillai's Trace	0.841	498.480	2	188	<0.0001
	Hotelling-Lawley Trace	5.303	498.480	2	188	<0.0001
	Roy's Greatest Root	5.303	498.480	2	188	<0.0001
time×W1SEX	Wilks' Lambda	0.876	13.370	2	188	<0.0001
	Pillai's Trace	0.124	13.370	2	188	<0.0001
	Hotelling-Lawley Trace	0.142	13.370	2	188	<0.0001
	Roy's Greatest Root	0.142	13.370	2	188	<0.0001

Table 16: MANOVA statistics

The comparisons of the the three different time levels as well as that for gender are shown in Table 17.

Source	time(level)	SS	d.o.f	MS	F-value	p-value
time	1 vs 2	180155.974	1	180155.974	915.200	<0.0001
	1 vs 2	173200.928	1	173200.928	23.760	<0.0001
	2 vs 3	68.454	1	68.454	0.720	0.397
W1SEX(gender)	1 vs 2	4676.558	1	4676.558	848.240	<0.0001
	1 vs 3	4826.980	1	4826.980	23.640	<0.0001
	2 vs 3	1.190	1	1.190	0.100	0.911
residual	1 vs 2	37204.324	189	196.848		
	1 vs 3	38591.633	189	204.186		
	2 vs 3	17930.000	189	94.868		

Table 17: Contrasts for the three different time periods with gender taken into account

The value of F-statistic for the comparison between week 1 and week 12 is significant showing that there is a significant difference between the values of week 1 and week 12. The value of F-statistic for the comparison between week 1 and week 20 is significant showing that there is a significant difference between the values of week 1 and week 20. The value of F-statistic for the comparison between week 12 and week 20 is not significant (p-value=0.397) meaning that there is no significant difference between the values of week 12 and week 20.

The comparisons that follow have the gender incorporated. The value of the F-statistic for the comparison between week 1 and week 12 is significant showing that there is a significant difference between the values of week 1 and week 12. The value of the$F$-statistic for the comparison between week 1 and week 20 is significant showing that there is a significant difference between the values of week 1 and week 20. The value of the F-statistic for the comparison between week 12 and week 20 is not significant (p-value=0.911) indicating that there is no significant difference between the values of week 12 and week 20.

3.2 Multiple factor repeated measures ANOVA: cricketers

3.2.1 Problem Statement

To maximise performance in competitive sport, full recovery is important for athletes. Those who recover more rapidly and more efficiently are able to train harder and more intensely. Without proper recovery after training sessions or competitions, athletes are prone to risk of poorer performance and overuse injury. Studies have shown that there is an improvement in heart rate and blood pressure responses of athletes observed with the application of lower negative body pressure (LNBP) after 15 days of bed rest. Athletes often experience symptoms of discomfort, muscular soreness and stiffness within $12 - 24$ hours following excessive training. Imitating sports massages by stimulating the circulatory system and lymphatic vessels, LBNP is claimed to play an important role in the recovery of the athlete in order to maximise athletic performance in competitive sport. Different athletes have different endurance capacities, with vastly trained athletes performing at the most oxygen uptake presenting with least lactate accumulation. During exercise of increasing intensity, an increase in blood lactate concentration indicates a rise in glycogen metabolism within the muscle. However, the initial rise in the concentration of the lactate in the blood shows the net result of the production of lactate in the muscle and shows that the rate at which lactate in the blood appears is higher than the rate at which it disappears. This is the lactate threshold and is considered to be a good predictor of endurance exercise performance. [6]

3.2.2 Study design

A randomised cross-over study design with repeated measures was done to determine the effect LBNP treatment has on the recovery of cricketers. Twenty-two healthy male cricket players, aged 19.5± 0.09 years, weighing 79.63±8.17 kg with a height of 180±0.07 cm were invited to participate voluntarily. These cricketers are based at the TUKS Cricket Academy, at the High Performance Centre (HPC) of the University of Pretoria. The cricketers were randomly assigned to two groups with eleven volunteers each and evaluated. After the second week, the cricketers were crossed over and the study repeated over the following two weeks. Each player took part in the cross-over design during the four-week study period: once in the treatment group i.e. receiving LBNP treatment and once in the control group i.e. not receiving LBNP treatment. Participants were instructed to refrain from drinking alcohol and consuming caffeine products 24 hours prior to each study session. The two applications were separated by a washout period of 14 days. The cricketers continued to follow their regular training program consisting of 7 hours conditioning and 10 hours cricket specific skills with one competitive match on weekends throughout the study period. A small blood sample

of 0.3 µl was collected by means of lancet prick of the earlobe to determine the blood lactate concentration (mmol/l) of all athletes. Lactate was sampled at rest on day 1 (Lactate 1), immediately after the 1 hr exercise session (Lactate 2) and directly after the first 30 minutes LBNP treatment session (Lactate 3).

3.2.3 Results

The descriptive statistics of the three measurements for the two groups are presented in tables 18 and 19.

	Lactate 1	Lactate 2	Lactate 3
Mean	0.973	2.177	1.064
Median	0.950	1.900	1.000
Standard Deviation	0.249	1.107	0.192

Table 18: Control group descriptive statistics

	Lactate 1	Lactate 2	Lactate 3
Mean	0.973	2.218	1.519
Median	0.900	2.050	1.600
Standard Deviation	0.249	1.020	0.437

Table 19: Treatment group descriptive statistics

Table 20 displays results of the sphericity test for the cricketers. The data used violates the assumption of normality. Nonetheless, it was decided to proceed with the analyses since there is no non-parametric method available to compare the two groups in this repeated measures experiment.

Variables	d.o.f	Mauchly's criterion	Chi-square	p-value
Orthogonal components	2	0.446	33.077	<0.0001

Table 20: Sphericty test

Mauchly's test is significant (p-value<0.0001) as shown in table 20. Therefore, the assumption that the variances of the differences between levels are equal is rejected i.e. the sphericity assumption has been violated. Table 21 displays the options for sphericity corrections and their estimates. Since the Greenhouse-Geisser estimate is 0.644 and is close to 1, it can be assumed that the variances of differences are homogenous. Therefore the data is close to being spherical.

Correction	Estimate
Greenhouse-Geisser ($\hat{\varepsilon}$)	0.644
Huynh-Feldt ($\tilde{\varepsilon}$)	0.655
Lower bound	0.500

Table 21: Sphericity Corrections

Table 22 below shows that the F-statistic for gender is significant (p-value$<$0.0001). The null hypothesis that there are no differences between the control and treatment groups is rejected and so we conclude that that there is a significant difference between the two groups.

Source	d.o.f	SS	MS	F-value	p-value
Group	1	0.669	0.669	1.190	0.282
Residual	42	23.693	0.564		

Table 22: Repeated measures ANOVA for between-subjects

Table 23 displays results of the repeated measures ANOVA tests of the hypotheses for within-subject effects. There is a sum of squares for the repeated-measures effect of lactate which tells us how much of the total variability is explained by the experimental effect. There is also an interaction term, which is variation caused by the interaction between lactate and group. There is also a residual term, which is the amount of unexplained variation across the conditions of the repeated measures variable, which is lactate in this case.

Source	d.o.f	SS	MS	F-value	p-value	adj p-value Greenhouse-Geisser	adj p-value Huynh-Feldt
lactate	2	36.177	18.089	45.640	$<$0.0001	$<$0.0001	$<$0.0001
lactate$\times$group	2	0.991	0.496	1.250	0.292	0.280	0.281
residual	84	33.292	0.396				

Table 23: Repeated measures ANOVA for within-subjects

Since the p-value for lactate is less than 0.0001 which is less than 0.05, then the F-ratio is significant. Therefore, the conclusion is that there was a significant difference between the times. For the interactions, the F-statistic is not significant (p-value=0.292). This means that there was no significant difference between the interactions. But since the data is not spherical and violates the assumption of normality, the F ratios may be inaccurate.

Table 24displays the contrasts between the different levels of the lactate.

Source	(lactate level)	SS	d.o.f	MS	F-value	p-value
lactate	1 vs 2	66.028	1	66.028	57.150	$<$0.0001
	1 vs 3	3.551	1	3.551	17.032	$<$0.0001
	2 vs 3	38.954	1	38.954	38.411	$<$0.0001
lactate$\times$group	1 vs 2	0.018	1	0.018	0.016	0.900
	1 vs 3	1.642	1	1.642	7.876	0.008
	2 vs 3	1.313	1	1.313	1.294	0.262
residual	1 vs 2	48.524	42	1.155		
	1 vs 3	8.757	42	0.208		
	2 vs 3	42.594	42	1.014		

Table 24: Contrasts for the different levels of lactate

The value of F-statistic for the comparison between lactate1 and lactate2 is significant. This shows that

there is a significant difference between the values of lactate1 and lactate2. The value of F-statistic for the comparison between lactate1 and lactate3 is significant showing that there is a significant difference between the values of lactate1 and lactate3. The value of F-statistic for the comparison between lactate2 and lactate3 is significant showing that there is a significant difference between the values of lactate2 and lactate3.

The comparisons that follow have the control group and treatment group incorporated. The value of the F-statistic for the comparison between lactate1 and lactate2 is not significant indicating that there is no significant difference between the values of lactate1 and lactate2. The value of the$F$-statistic for the comparison between lactate1 and lactate3 is significant indicating that there is a significant difference between the values of lactate1 and lactate3. The value of the F-statistic for the comparison between lactate2 and lactate3 is not significant indicating that there is no significant difference between the values of lactate2 and lactate3.

The MANOVA results displayed in Table 25 confirm the conclusion that there are significant differences between the time periods and bu does not support the conclusion there are no significant differences between the interactions.

Effect	Statistic	Value	F-value	Hypothesis d.o.f	Residual d.o.f	p-value
lactate	Wilks' Lambda	0.414	29.04	2	41	<0.0001
	Pillai's Trace	0.586	29.04	2	41	<0.0001
	Hotelling-Lawley Trace	1.417	29.04	2	41	<0.0001
	Roy's Greatest Root	1.417	29.04	2	41	<0.0001
lactate×group	Wilks' Lambda	0.828	4.27	2	41	0.0207
	Pillai's Trace	0.172	4.27	2	41	0.0207
	Hotelling-Lawley Trace	0.208	4.27	2	41	0.0207
	Roy's Greatest Root	0.208	4.27	2	41	0.0207

Table 25: MANOVA statistics

4 Conclusion

The data that is used in practice often violates the assumption of normality and/or other assumptions for a parametric test. The SANDF data that was used in the first application of this report violated the normality assumption. This led to the use of the non-parametric test, Friedman's test. Since the data set was large, the central limit theorem was used to approximate the normal distribution. Repeated measures ANOVA was then used. Comparing the two methods, we found that the same results were obtained. This means that repeated measures ANOVA is robust to the normality assumption hence it is a better option to use as compared to an independent ANOVA since it requires data to be normal.

Since repeated measures ANOVA is robust, we could the proceed to investigate if there were gender

differences. We found that the males differ significantly from the females meaning that physical abilities in the military are affected by gender. The level of physical training done by men and that done by females is significantly different. In the second application, there was no non-parametric test to test between groups and since repeated measures ANOVA is robust, it was used since.

After doing some contrasts we found that the measurements taken at during week 12 and those taken after 20 weeks are not significantly different. This means there is no significant difference between a study done over 12 weeks and that done over 20 weeks. This implies that the study could have been done over 12 weeks and time could have been saved.

A few problems were encountered with the data used. The two groups in the SANDF data had unequal participants i.e. there were more males than females and this could have influenced the results obtained. There were also missing observations in the data set and SAS default setting is to not include that particular subject in the analysis. However, the data set was large and so these problems could be overlooked.

References

[1] Roger Bakeman. Recommended effect size statistics for repeated measures designs. *Behavior Research Methods*, 37(3):379–384, 2005.

[2] Joan Fisher Box. RA Fisher and the design of experiments, 1922-1926. *The American Statistician*, 34(1):1–7, 1980.

[3] Charles S Davis. *Statistical Methods for the Analysis of Repeated Measurements*. Springer Science & Business Media, 2002.

[4] Andy Field. *Discovering Statistics using SAS*. Sage publications, 2009.

[5] Ellen R Girden. *ANOVA: Repeated Measures*. Number 84. Sage, 1992.

[6] A Jansen van Rensburg, DC Janse van Rensburg, HE van Buuren, CC Grant, and L Fletcher. The use of negative pressure wave treatment in athlete recovery. *South African Journal of Sports Medicine*, 29:1–7, 2017.

[7] Scott E Maxwell and Harold D Delaney. *Designing Experiments and Analyzing Data: A Model Comparison Perspective*, volume 1. Psychology Press, 2004.

[8] Douglas C Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 2008.

[9] H Rouanet and D Lepine. Comparison between treatments in a repeated-measurement design: Anova and multivariate methods. *British Journal of Mathematical and Statistical Psychology*, 23(2):147–163, 1970.

[10] James P Stevens. *Applied Multivariate Statistics for the Social Sciences*. Routledge, 2002.

[11] AGW Steyn, CFT Smit, SHC Du Toit, and C Strashem. *Modern Statistics In Practice*. Van Schaik, 1994.

[12] Paola S Wood, Catharina C Grant, Peet J du Toit, and Lizelle Fletcher. Effect of mixed basic military training on the physical fitness of male and female soldiers. *Military Medicine*, 182(7), 2017.

Appendix

Code

Repeated measures ANOVA: SANDF

Descriptive statistics

```
proc univariate data=sandf normal plot;
var w1PT w12PT w20PT;
by W1SEX;
qqplot w1PT w12PT w20PT/normal(mu=est sigma=est color=green L=1);
inset mean std/cfill=blank format=5.2;
run;
```

Friedman test

```
 proc transpose data=sandf out=sandf1;
 var w1PT w12PT w20PT;
 by ID;
 run;
 proc freq data=sandf1;
   tables ID*_NAME_*COL1/
           cmh2 scores=rank noprint;
run;

proc rank data=sandf1 out=sandf1rank;
var COL1;
run;
```

```
proc means data=sandf1rank mean;
class _NAME_;
var COL1;
run;
```

Repeated measures ANOVA

```
proc glm data=sandf;
model w1PT w12PT w20PT=/nouni;
repeated time 3 contrast(1)/summary printe;
repeated time 3 contrast(2)/summary printe;
repeated time 3 contrast(3)/summary printe;
run;
```

Repeated measures with factor

```
proc glm data=sandf;
class W1SEX;
model w1PT w12PT w20PT=W1SEX/nouni;
repeated time 3 contrast(1)/summary printe;
repeated time 3 contrast(2)/summary printe;
repeated time 3 contrast(3)/summary printe;
run;
```

Multiple factor repeated measures ANOVA : cricketers

Descriptive statistics

```
/*control*/
proc univariate data=report normal plot;
var lactate1 lactate2 lactate3;
qqplot lactate1 lactate2 lactate3/normal(mu=est sigma=est color=green L=1);
inset mean std/cfill=blank format=5.2;
run;


/*treatment*/
proc univariate data=report normal plot;
var lactate1 lactate2 lactate3;
qqplot lactate1 lactate2 lactate3/normal(mu=est sigma=est color=green L=1);
inset mean std/cfill=blank format=5.2;
run;
```

Repeated measures ANOVA

```
proc glm data=report;
class group;
model lactate1 lactate2 lactate3=group/nouni;
repeated lactate 3 contrast(1)/summary printe;
repeated lactate 3 contrast(2)/summary printe;
repeated lactate 3 contrast(3)/summary printe;
run;
```

Output

Repeated measures ANOVA

Descriptive statistics

The UNIVARIATE Procedure
Variable: W1PT (W1PT)

Moments			
N	191	Sum Weights	191
Mean	54.004712	Sum Observations	10314.9
Std Deviation	17.2602473	Variance	297.916136
Skewness	-0.4965325	Kurtosis	-0.2330154
Uncorrected SS	613657.27	Corrected SS	56604.0658
Coeff Variation	31.960632	Std Error Mean	1.24890766

Basic Statistical Measures			
Location		Variability	
Mean	54.00471	Std Deviation	17.26025
Median	56.90000	Variance	297.91614
Mode	23.00000	Range	86.60000
		Interquartile Range	26.00000

Note: The mode displayed is the smallest of 8 modes with a count of 3.

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.975023	Pr < W	0.0017
Kolmogorov-Smirnov	D	0.074811	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.23427	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	1.353062	Pr > A-Sq	<0.0050

Figure 2: Basic statistical measures and normality tests for week 1

The UNIVARIATE Procedure
Variable: W12PT (W12PT)

Moments			
N	191	Sum Weights	191
Mean	84.3455497	Sum Observations	16110
Std Deviation	8.50500587	Variance	72.3351248
Skewness	-0.9505616	Kurtosis	1.71983599
Uncorrected SS	1372550.48	Corrected SS	13743.6737
Coeff Variation	10.0835265	Std Error Mean	0.61540063

Basic Statistical Measures			
Location		Variability	
Mean	84.34555	Std Deviation	8.50501
Median	85.70000	Variance	72.33512
Mode	81.60000	Range	50.80000
		Interquartile Range	10.70000

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.953741	Pr < W	<0.0001
Kolmogorov-Smirnov	D	0.089246	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.270512	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	1.670129	Pr > A-Sq	<0.0050

Figure 3: Basic statistical measures and normality tests during week 12

The UNIVARIATE Procedure
Variable: W20PT (W20PT)

Moments			
N	191	Sum Weights	191
Mean	83.7175393	Sum Observations	15990.05
Std Deviation	10.582014	Variance	111.97902
Skewness	-1.0696595	Kurtosis	1.46223989
Uncorrected SS	1359923.65	Corrected SS	21276.0137
Coeff Variation	12.6401398	Std Error Mean	0.76568766

Basic Statistical Measures			
Location		Variability	
Mean	83.71754	Std Deviation	10.58201
Median	85.30000	Variance	111.97902
Mode	85.30000	Range	55.60000
		Interquartile Range	11.40000

Note: The mode displayed is the smallest of 2 modes with a count of 4.

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.931296	Pr < W	<0.0001
Kolmogorov-Smirnov	D	0.106027	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.499983	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	3.016526	Pr > A-Sq	<0.0050

Figure 4: Basic statistical measures and normality tests after week 20

The UNIVARIATE Procedure
Variable: W1PT (W1PT)

W1SEX=1

Moments			
N	115	Sum Weights	115
Mean	59.5843478	Sum Observations	6852.2
Std Deviation	14.6174349	Variance	213.669402
Skewness	-0.617975	Kurtosis	0.13813316
Uncorrected SS	432642.18	Corrected SS	24358.3118
Coeff Variation	24.5323401	Std Error Mean	1.36308283

Basic Statistical Measures			
Location		Variability	
Mean	59.58435	Std Deviation	14.61743
Median	61.10000	Variance	213.66940
Mode	61.10000	Range	69.90000
		Interquartile Range	17.40000

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.968784	Pr < W	0.0087
Kolmogorov-Smirnov	D	0.079557	Pr > D	0.0739
Cramer-von Mises	W-Sq	0.15273	Pr > W-Sq	0.0223
Anderson-Darling	A-Sq	0.980903	Pr > A-Sq	0.0144

Figure 5: Basic statistical measures and normality tests for males during the first week

The UNIVARIATE Procedure
Variable: W12PT (W12PT)

W1SEX=1

Moments			
N	115	Sum Weights	115
Mean	85.9026087	Sum Observations	9878.8
Std Deviation	6.56670845	Variance	43.1216598
Skewness	-0.2114384	Kurtosis	-0.2623724
Uncorrected SS	853530.56	Corrected SS	4915.86922
Coeff Variation	7.64436441	Std Error Mean	0.61234872

Basic Statistical Measures			
Location		Variability	
Mean	85.90261	Std Deviation	6.56671
Median	86.40000	Variance	43.12166
Mode	81.60000	Range	31.10000
		Interquartile Range	9.00000

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.990803	Pr < W	0.6380
Kolmogorov-Smirnov	D	0.044215	Pr > D	>0.1500
Cramer-von Mises	W-Sq	0.036584	Pr > W-Sq	>0.2500
Anderson-Darling	A-Sq	0.259103	Pr > A-Sq	>0.2500

Figure 6: Basic statistical measures and normality tests for males during week 12

The UNIVARIATE Procedure
Variable: W20PT (W20PT)

W1SEX=1

Moments			
N	115	Sum Weights	115
Mean	85.2104348	Sum Observations	9799.2
Std Deviation	9.69335433	Variance	93.9611182
Skewness	-1.3542191	Kurtosis	2.9627347
Uncorrected SS	845705.66	Corrected SS	10711.5675
Coeff Variation	11.3757832	Std Error Mean	0.90390995

Basic Statistical Measures			
Location		Variability	
Mean	85.21043	Std Deviation	9.69335
Median	86.30000	Variance	93.96112
Mode	93.40000	Range	55.60000
		Interquartile Range	11.40000

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.912662	Pr < W	<0.0001
Kolmogorov-Smirnov	D	0.102247	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.313148	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	2.031218	Pr > A-Sq	<0.0050

Figure 7: Basic statistical measures and normality tests for males after week 20

The UNIVARIATE Procedure
Variable: W1PT (W1PT)

W1SEX=2

Moments			
N	76	Sum Weights	76
Mean	45.5618421	Sum Observations	3462.7
Std Deviation	17.6060972	Variance	309.974658
Skewness	-0.1429044	Kurtosis	-0.416226
Uncorrected SS	181015.09	Corrected SS	23248.0993
Coeff Variation	38.6421979	Std Error Mean	2.01955785

Basic Statistical Measures			
Location		Variability	
Mean	45.56184	Std Deviation	17.60610
Median	44.90000	Variance	309.97466
Mode	19.00000	Range	80.00000
		Interquartile Range	25.90000

Note: The mode displayed is the smallest of 7 modes with a count of 2.

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.988129	Pr < W	0.7051
Kolmogorov-Smirnov	D	0.053954	Pr > D	>0.1500
Cramer-von Mises	W-Sq	0.035686	Pr > W-Sq	>0.2500
Anderson-Darling	A-Sq	0.232182	Pr > A-Sq	>0.2500

Figure 8: Basic statistical measures and normality tests for females during week 1

The UNIVARIATE Procedure
Variable: W12PT (W12PT)

W1SEX=2

Moments			
N	76	Sum Weights	76
Mean	81.9894737	Sum Observations	6231.2
Std Deviation	10.4096824	Variance	108.361488
Skewness	-0.8279243	Kurtosis	0.6643611
Uncorrected SS	519019.92	Corrected SS	8127.11158
Coeff Variation	12.6963645	Std Error Mean	1.19407246

Basic Statistical Measures			
Location		Variability	
Mean	81.98947	Std Deviation	10.40968
Median	84.00000	Variance	108.36149
Mode	75.00000	Range	50.00000
		Interquartile Range	13.05000

Note: The mode displayed is the smallest of 5 modes with a count of 2.

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.954708	Pr < W	0.0085
Kolmogorov-Smirnov	D	0.119682	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.142392	Pr > W-Sq	0.0303
Anderson-Darling	A-Sq	0.891004	Pr > A-Sq	0.0225

Figure 9: Basic statistical measures and normality tests for females during week 12

The UNIVARIATE Procedure
Variable: W20PT (W20PT)

W1SEX=2

Moments			
N	76	Sum Weights	76
Mean	81.4585526	Sum Observations	6190.85
Std Deviation	11.5009054	Variance	132.270826
Skewness	-0.7236544	Kurtosis	0.41783927
Uncorrected SS	514217.993	Corrected SS	9920.31194
Coeff Variation	14.1187206	Std Error Mean	1.31924433

Basic Statistical Measures			
Location		Variability	
Mean	81.45855	Std Deviation	11.50091
Median	83.55000	Variance	132.27083
Mode	88.10000	Range	51.00000
		Interquartile Range	14.82500

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.951454	Pr < W	0.0056
Kolmogorov-Smirnov	D	0.108769	Pr > D	0.0245
Cramer-von Mises	W-Sq	0.173429	Pr > W-Sq	0.0117
Anderson-Darling	A-Sq	0.981393	Pr > A-Sq	0.0140

Figure 10: Basic statistical measures and normality tests for females after week 20

Friedman's test

The FREQ Procedure

Summary Statistics for _NAME_ by COL1
Controlling for ID

Cochran-Mantel-Haenszel Statistics (Based on Rank Scores)				
Statistic	Alternative Hypothesis	DF	Value	Prob
1	Nonzero Correlation	1	0.2628	0.6082
2	Row Mean Scores Differ	2	270.1051	<.0001

Total Sample Size = 573

The MEANS Procedure

Analysis Variable : COL1 Values of COL1 Were Replaced by Ranks		
NAME OF FORMER VARIABLE	N Obs	Mean
W12PT	191	374.7696335
W1PT	191	115.4502618
W20PT	191	370.7801047

Figure 11: Friedman's Test

Partial Correlation Coefficients from the Error SSCP Matrix / Prob > |r|			
DF = 190	W1PT	W12PT	W20PT
W1PT	1.000000	0.510310 <.0001	0.496518 <.0001
W12PT	0.510310 <.0001	1.000000	0.499660 <.0001
W20PT	0.496518 <.0001	0.499660 <.0001	1.000000

Figure 12: Partial Correlations

Sphericity Tests				
Variables	DF	Mauchly's Criterion	Chi-Square	Pr > ChiSq
Transformed Variates	2	0.3759163	184.91547	<.0001
Orthogonal Components	2	0.7699927	49.399724	<.0001

Greenhouse-Geisser Epsilon	0.8130
Huynh-Feldt Epsilon	0.8190

Figure 13: Sphericity tests and corrections

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no time Effect H = Type III SSCP Matrix for time E = Error SSCP Matrix S=1 M=0 N=93.5					
Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.18089176	427.91	2	189	<.0001
Pillai's Trace	0.81910824	427.91	2	189	<.0001
Hotelling-Lawley Trace	4.52816772	427.91	2	189	<.0001
Roy's Greatest Root	4.52816772	427.91	2	189	<.0001

Figure 14: MANOVA statistics

The GLM Procedure
Repeated Measures Analysis of Variance
Univariate Tests of Hypotheses for Within Subject Effects

Source	DF	Type III SS	Mean Square	F Value	Pr > F	Adj Pr > F	
						G - G	H - F
time	2	114842.7566	57421.3783	634.12	<.0001	<.0001	<.0001
Error(time)	380	34410.2551	90.5533				

Figure 15: Repeated measures ANOVA for within-subjects

The GLM Procedure
Repeated Measures Analysis of Variance
Analysis of Variance of Contrast Variables

time_N represents the contrast between the nth level of time and the 1st

Contrast Variable: time_2

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	175828.1885	175828.1885	797.68	<.0001
Error	190	41880.8815	220.4257		

Contrast Variable: time_3

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	168624.7514	168624.7514	737.90	<.0001
Error	190	43418.5711	228.5188		

Figure 16: Contrast with time 1

The GLM Procedure
Repeated Measures Analysis of Variance
Analysis of Variance of Contrast Variables

time_N represents the contrast between the nth level of time and the 2nd

Contrast Variable: time_1

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	175828.1885	175828.1885	797.68	<.0001
Error	190	41880.8815	220.4257		

Contrast Variable: time_3

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	75.32986	75.32986	0.80	0.3728
Error	190	17931.31264	94.37533		

Figure 17: Contrast with time 2

The GLM Procedure
Repeated Measures Analysis of Variance
Analysis of Variance of Contrast Variables

time_N represents the contrast between the nth level of time and the 3rd

Contrast Variable: time_1

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	168624.7514	168624.7514	737.90	<.0001
Error	190	43418.5711	228.5188		

Contrast Variable: time_2

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	75.32986	75.32986	0.80	0.3728
Error	190	17931.31264	94.37533		

Figure 18: Contrasts with time 3

Repeated Measures with factor

Partial Correlation Coefficients from the Error SSCP Matrix / Prob > |r|			
DF = 189	W1PT	W12PT	W20PT
W1PT	1.000000	0.470436 <.0001	0.472981 <.0001
W12PT	0.470436 <.0001	1.000000	0.479897 <.0001
W20PT	0.472981 <.0001	0.479897 <.0001	1.000000

Figure 19: Partial Correlations

Sphericity Tests				
Variables	DF	Mauchly's Criterion	Chi-Square	Pr > ChiSq
Transformed Variates	2	0.416821	164.51852	<.0001
Orthogonal Components	2	0.8177903	37.816064	<.0001

Greenhouse-Geisser Epsilon	0.8459
Huynh-Feldt-Lecoutre Epsilon	0.8527

Figure 20: Sphericity tests and corrections

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no time Effect H = Type III SSCP Matrix for time E = Error SSCP Matrix S=1 M=0 N=93					
Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.15865599	498.48	2	188	<.0001
Pillai's Trace	0.84134401	498.48	2	188	<.0001
Hotelling-Lawley Trace	5.30294505	498.48	2	188	<.0001
Roy's Greatest Root	5.30294505	498.48	2	188	<.0001

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no time*W1SEX Effect H = Type III SSCP Matrix for time*W1SEX E = Error SSCP Matrix S=1 M=0 N=93					
Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.87550366	13.37	2	188	<.0001
Pillai's Trace	0.12449634	13.37	2	188	<.0001
Hotelling-Lawley Trace	0.14219968	13.37	2	188	<.0001
Roy's Greatest Root	0.14219968	13.37	2	188	<.0001

Figure 21: MANOVA statistics

The GLM Procedure
Repeated Measures Analysis of Variance
Tests of Hypotheses for Between Subjects Effects

Source	DF	Type III SS	Mean Square	F Value	Pr > F
W1SEX	1	7174.25339	7174.25339	27.10	<.0001
Error	189	50039.24477	264.75791		

Figure 22: Repeated measures ANOVA for between-subjects

The GLM Procedure
Repeated Measures Analysis of Variance
Univariate Tests of Hypotheses for Within Subject Effects

Source	DF	Type III SS	Mean Square	F Value	Pr > F	Adj Pr > F	
						G - G	H-F-L
time	2	117808.4519	58904.2260	712.69	<.0001	<.0001	<.0001
time*W1SEX	2	3168.2284	1584.1142	19.17	<.0001	<.0001	<.0001
Error(time)	378	31242.0266	82.6509				

Figure 23: Repeated measures ANOVA for within-subjects

The GLM Procedure
Repeated Measures Analysis of Variance
Analysis of Variance of Contrast Variables

time_N represents the contrast between the nth level of time and the 1st

Contrast Variable: time_2

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	180155.9735	180155.9735	915.20	<.0001
W1SEX	1	4676.5578	4676.5578	23.76	<.0001
Error	189	37204.3236	196.8483		

Contrast Variable: time_3

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	173200.9282	173200.9282	848.24	<.0001
W1SEX	1	4826.9377	4826.9377	23.64	<.0001
Error	189	38591.6334	204.1885		

Figure 24: Contrast with time 1 with gender taken into account

The GLM Procedure
Repeated Measures Analysis of Variance
Analysis of Variance of Contrast Variables

time_N represents the contrast between the nth level of time and the 2nd

Contrast Variable: time_1

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	180155.9735	180155.9735	915.20	<.0001
W1SEX	1	4676.5578	4676.5578	23.76	<.0001
Error	189	37204.3236	196.8483		

Contrast Variable: time_3

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	68.45394	68.45394	0.72	0.3967
W1SEX	1	1.18985	1.18985	0.01	0.9109
Error	189	17930.12279	94.86837		

Figure 25: Contrast with time 2 with gender taken into account

The GLM Procedure
Repeated Measures Analysis of Variance
Analysis of Variance of Contrast Variables

time_N represents the contrast between the nth level of time and the 3rd

Contrast Variable: time_1

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	173200.9282	173200.9282	848.24	<.0001
W1SEX	1	4826.9377	4826.9377	23.64	<.0001
Error	189	38591.6334	204.1885		

Contrast Variable: time_2

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	68.45394	68.45394	0.72	0.3967
W1SEX	1	1.18985	1.18985	0.01	0.9109
Error	189	17930.12279	94.86837		

Figure 26: Contrasts with time 3 with gender taken into account

Multiple factor repeated measures ANOVA

The UNIVARIATE Procedure
Variable: LACTATE1 (LACTATE1_B)

Moments			
N	22	Sum Weights	22
Mean	0.97272727	Sum Observations	21.4
Std Deviation	0.24915441	Variance	0.06207792
Skewness	1.68351851	Kurtosis	5.07453384
Uncorrected SS	22.12	Corrected SS	1.30363636
Coeff Variation	25.6140052	Std Error Mean	0.0531199

Basic Statistical Measures			
Location		Variability	
Mean	0.972727	Std Deviation	0.24915
Median	0.950000	Variance	0.06208
Mode	0.900000	Range	1.20000
		Interquartile Range	0.10000

Note: The mode displayed is the smallest of 2 modes with a count of 6.

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.850758	Pr < W	0.0035
Kolmogorov-Smirnov	D	0.229146	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.176796	Pr > W-Sq	0.0094
Anderson-Darling	A-Sq	1.028784	Pr > A-Sq	0.0086

Figure 27: Basic statistical measures and normality tests for control 1

The UNIVARIATE Procedure
Variable: LACTATE2 (LACTATE2_0H)

Moments			
N	22	Sum Weights	22
Mean	2.17727273	Sum Observations	47.9
Std Deviation	1.10709049	Variance	1.22564935
Skewness	1.91750773	Kurtosis	3.45815446
Uncorrected SS	130.03	Corrected SS	25.7386364
Coeff Variation	50.8475799	Std Error Mean	0.23603249

Basic Statistical Measures			
Location		Variability	
Mean	2.177273	Std Deviation	1.10709
Median	1.900000	Variance	1.22565
Mode	1.900000	Range	4.10000
		Interquartile Range	1.00000

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.755551	Pr < W	0.0001
Kolmogorov-Smirnov	D	0.248967	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.325207	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	1.926321	Pr > A-Sq	<0.0050

Figure 28: Basic statistical measures and normality tests for control 2

The UNIVARIATE Procedure
Variable: LACTATE3 (LACTATE3)

Moments			
N	22	Sum Weights	22
Mean	1.06363636	Sum Observations	23.4
Std Deviation	0.19159843	Variance	0.03670996
Skewness	0.26182415	Kurtosis	0.06422633
Uncorrected SS	25.66	Corrected SS	0.77090909
Coeff Variation	18.0135272	Std Error Mean	0.04084892

Basic Statistical Measures			
Location		Variability	
Mean	1.063636	Std Deviation	0.19160
Median	1.000000	Variance	0.03671
Mode	1.000000	Range	0.80000
		Interquartile Range	0.20000

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.964019	Pr < W	0.5745
Kolmogorov-Smirnov	D	0.175561	Pr > D	0.0773
Cramer-von Mises	W-Sq	0.083585	Pr > W-Sq	0.1824
Anderson-Darling	A-Sq	0.432692	Pr > A-Sq	>0.2500

Figure 29: Basic statistical measures and normality tests for control 3

The UNIVARIATE Procedure
Variable: LACTATE1 (LACTATE1_B)

Moments			
N	22	Sum Weights	22
Mean	0.97272727	Sum Observations	21.4
Std Deviation	0.24915441	Variance	0.06207792
Skewness	0.76912865	Kurtosis	-0.4055682
Uncorrected SS	22.12	Corrected SS	1.30363636
Coeff Variation	25.6140052	Std Error Mean	0.0531199

Basic Statistical Measures			
Location		Variability	
Mean	0.972727	Std Deviation	0.24915
Median	0.900000	Variance	0.06208
Mode	0.700000	Range	0.80000
		Interquartile Range	0.30000

Note: The mode displayed is the smallest of 2 modes with a count of 5.

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.888599	Pr < W	0.0177
Kolmogorov-Smirnov	D	0.205726	Pr > D	0.0165
Cramer-von Mises	W-Sq	0.131386	Pr > W-Sq	0.0405
Anderson-Darling	A-Sq	0.844406	Pr > A-Sq	0.0246

Figure 30: Basic statistical measures and normality tests for treatment 1

The UNIVARIATE Procedure
Variable: LACTATE2 (LACTATE2_0H)

Moments			
N	22	Sum Weights	22
Mean	2.21818182	Sum Observations	48.8
Std Deviation	1.02010101	Variance	1.04060606
Skewness	0.77007444	Kurtosis	-0.1124339
Uncorrected SS	130.1	Corrected SS	21.8527273
Coeff Variation	45.9881601	Std Error Mean	0.21748627

Basic Statistical Measures			
Location		Variability	
Mean	2.218182	Std Deviation	1.02010
Median	2.050000	Variance	1.04061
Mode	1.200000	Range	3.60000
		Interquartile Range	1.40000

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.923356	Pr < W	0.0892
Kolmogorov-Smirnov	D	0.136834	Pr > D	>0.1500
Cramer-von Mises	W-Sq	0.074889	Pr > W-Sq	0.2348
Anderson-Darling	A-Sq	0.543402	Pr > A-Sq	0.1475

Figure 31: Basic statistical measures and normality tests for treatment 2

The UNIVARIATE Procedure
Variable: LACTATE3 (LACTATE3)

Moments			
N	22	Sum Weights	22
Mean	1.45	Sum Observations	31.9
Std Deviation	0.53519022	Variance	0.28642857
Skewness	-0.5945633	Kurtosis	1.26543321
Uncorrected SS	52.27	Corrected SS	6.015
Coeff Variation	36.9096703	Std Error Mean	0.11410294

Basic Statistical Measures			
Location		Variability	
Mean	1.450000	Std Deviation	0.53519
Median	1.550000	Variance	0.28643
Mode	1.100000	Range	2.40000
		Interquartile Range	0.70000

Note: The mode displayed is the smallest of 2 modes with a count of 4.

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.93906	Pr < W	0.1893
Kolmogorov-Smirnov	D	0.110366	Pr > D	>0.1500
Cramer-von Mises	W-Sq	0.07629	Pr > W-Sq	0.2263
Anderson-Darling	A-Sq	0.510396	Pr > A-Sq	0.1839

Figure 32: Basic statistical measures and normality tests for treatment 3

Partial Correlation Coefficients from the Error SSCP Matrix / Prob > |r|			
DF = 42	LACTATE1	LACTATE2	LACTATE3
LACTATE1	1.000000	0.075164 0.6319	0.075645 0.6297
LACTATE2	0.075164 0.6319	1.000000	0.327855 0.0319
LACTATE3	0.075645 0.6297	0.327855 0.0319	1.000000

Figure 33: Partial Correlations

Sphericity Tests				
Variables	DF	Mauchly's Criterion	Chi-Square	Pr > ChiSq
Transformed Variates	2	0.4522719	32.532338	<.0001
Orthogonal Components	2	0.4463043	33.076926	<.0001

Greenhouse-Geisser Epsilon	0.6436
Huynh-Feldt-Lecoutre Epsilon	0.6552

Figure 34: Sphericity tests and corrections

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no lactate Effect
H = Type III SSCP Matrix for lactate
E = Error SSCP Matrix

S=1 M=0 N=19.5

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.41377382	29.04	2	41	<.0001
Pillai's Trace	0.58622618	29.04	2	41	<.0001
Hotelling-Lawley Trace	1.41677929	29.04	2	41	<.0001
Roy's Greatest Root	1.41677929	29.04	2	41	<.0001

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no lactate*GROUP Effect
H = Type III SSCP Matrix for lactate*GROUP
E = Error SSCP Matrix

S=1 M=0 N=19.5

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.82759121	4.27	2	41	0.0207
Pillai's Trace	0.17240879	4.27	2	41	0.0207
Hotelling-Lawley Trace	0.20832603	4.27	2	41	0.0207
Roy's Greatest Root	0.20832603	4.27	2	41	0.0207

Figure 35: MANOVA statistics

The GLM Procedure
Repeated Measures Analysis of Variance
Tests of Hypotheses for Between Subjects Effects

Source	DF	Type III SS	Mean Square	F Value	Pr > F
GROUP	1	0.66939394	0.66939394	1.19	0.2822
Error	42	23.69303030	0.56411977		

Figure 36: repeated measures ANOVA for between-subjects

The GLM Procedure
Repeated Measures Analysis of Variance
Univariate Tests of Hypotheses for Within Subject Effects

Source	DF	Type III SS	Mean Square	F Value	Pr > F	Adj Pr > F	
						G - G	H-F-L
lactate	2	36.17742424	18.08871212	45.64	<.0001	<.0001	<.0001
lactate*GROUP	2	0.99106061	0.49553030	1.25	0.2917	0.2802	0.2808
Error(lactate)	84	33.29151515	0.39632756				

Figure 37: repeated measures ANOVA for within-subjects

The GLM Procedure
Repeated Measures Analysis of Variance
Analysis of Variance of Contrast Variables

lactate_N represents the contrast between the nth level of lactate and the 1st

Contrast Variable: lactate_2

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	66.02750000	66.02750000	57.15	<.0001
GROUP	1	0.01840909	0.01840909	0.02	0.9002
Error	42	48.52409091	1.15533550		

Contrast Variable: lactate_3

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	3.55113636	3.55113636	17.03	0.0002
GROUP	1	1.64204545	1.64204545	7.88	0.0076
Error	42	8.75681818	0.20849567		

Figure 38: Contrasts with first level of lactate

The GLM Procedure
Repeated Measures Analysis of Variance
Analysis of Variance of Contrast Variables

lactate_N represents the contrast between the nth level of lactate and the 2nd

Contrast Variable: lactate_1

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	66.02750000	66.02750000	57.15	<.0001
GROUP	1	0.01840909	0.01840909	0.02	0.9002
Error	42	48.52409091	1.15533550		

Contrast Variable: lactate_3

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	38.95363636	38.95363636	38.41	<.0001
GROUP	1	1.31272727	1.31272727	1.29	0.2617
Error	42	42.59363636	1.01413420		

Figure 39: Contrasts with second level of lactate

The GLM Procedure
Repeated Measures Analysis of Variance
Analysis of Variance of Contrast Variables

lactate_N represents the contrast between the nth level of lactate and the 3rd

Contrast Variable: lactate_1

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	3.55113636	3.55113636	17.03	0.0002
GROUP	1	1.64204545	1.64204545	7.88	0.0076
Error	42	8.75681818	0.20849567		

Contrast Variable: lactate_2

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	38.95363636	38.95363636	38.41	<.0001
GROUP	1	1.31272727	1.31272727	1.29	0.2617
Error	42	42.59363636	1.01413420		

Figure 40: Contrasts with third level of lactate

Nested vs non-nested models: Model selection

Ruth Seema 12097901

STK795 Research Report

Submitted in partial fulfillment of the degree BCom(Hons) Statistics

Supervisor: Dr. J. Kleyn

Department of Statistics, University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

29 September 2017 (draft 2)

Abstract

This report distinguishes between nested and non-nested models. A model is nested within another models if it can be derived as a special case of the other. Two models are non-nested if it is not possible to transform one into the other by way of parametric restriction or a limiting process. We do so by means of hypothesis testing theory. Moreover we discuss the issue of model selection. There are many different explanatory variables that can be identified when investigating a theory. Consequently this may give rise to multiple competing models. We look at two approaches for selecting the best model, namely the discrimination approach and the discerning approach. One major function is to test the validity of a model (goodness-of-fit), which is commonly neglected. A few methods to measure goodness of fit are discussed. Furthermore practical example will be used to illustrate the different approaches.

Declaration

I, *Ruth Seema,* declare that this essay, submitted in partial fulfillment of the degree *Statistics BCom(Hons) Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____

Ruth Seema

_____

Dr. Judy Kleyn

_____

Date 29 September 2017

Acknowledgements

Firstly, I would like to thank my supervisor , Dr Judy Kleyn, for cheering me on when I felt like giving up, and having tolerated me throughout.

Secondly, I am thankful for my family, friends and colleagues for always being supportive, for believing in me and their encouragement.

Lastly, I am thankful to the Centre of Artificial Intelligence and Research (CAIR) for funding my postgraduate studies this year.

Contents

List of Figures

List of Tables

1 Introduction

Statistical modelling is a rigorous process of working to achieve a correct model to explain a given phenomenon. The topic of model specification and evaluation is vast, however some of the essential issues involved are model specification errors or models specification bias, the detection thereof and testing as well as the evaluation of competing models [9].
In order to carry out specification testing, we first need to distinguish between nested and non-nested models. Nested models are models that can be reduced to other models by placing one or more restrictions onto their parameters, i.e . Whereas with non-nested models, given (say) two models, one cannot be derived from the other.

> "Statisticians are often faced with the problem of choosing the appropriate dimentionality of a model that will fit a given set of observations [21]."

Very often in econometric literature, model selection criteria and model specification tests are treated as very closely related or even as rival procedures. This is particularly misleading as these procedures serve different purposes. On the on hand model selection criteria are suitable for picking a model among several competing models. Non-nested hypothesis tests, on the other hand model are tests for model specification, just like tests for serial correlation or missing variables. The only difference with these tests from more classical tests is that they are conducted given the existence of alternative non-nested models [14].
The full reality of a phenomenon cannot be captured in a model [18]. In multiple regression, the objective of the analysis is (a) to explain the variation in a response variable and (b) to predict future performance on a response variable.

According to Olejnik and Keselman [19], when researchers investigate a particular phenomenon they identify a set of explanatory and predictor variables based on some combination of theory, experience and convenience. The number of the identified variables may exceed the number of individuals available to evaluate them. While there may not be enough time and resources to identify all the variables, an extensive list of identified variables can complicate the analysis and the researcher desires simplification. In any of these events a researcher will seek to reduce the number of the identified variables. Model selection is the task of choosing the best subset of these variables. In other words if we have competing models each made up of a subset of the listed variables, the best model will provide the best explanation of variation in the response variable or

it will have a high predictive power of future performance of the response variable, or both.

A good model is parsimonious (simple), with the least assumptions and variables, not under fitted and not overfitted [9]. However in real life, problems are much more complex and they require a little more than a simple model. Therefore there is always a trade-off between simple models and complex models. At times model selection could mean deselecting false models and only retaining a subset of the models that are close to the truth [11]. To obtain a model that is correct there are several criteria and methods developed to carry out the task.

Olejnik and Keselman [19] suggest that the stepwise procedures could be the most popular procedures to select the best subset of variables, to explain the response variable. There are there are two types of stepwise procedures. The backward elimination and the foreward selection method.
The forward selection porcedure adds predictors successively. The terms that enhance the fit of the model are selected at each stage. This selection is based on the p-value, i.e terms with a smaller p-value are prefered. The process stops when additional terms do not add significantly to the fit of the model. Backwards elimination starts with an elaborate model and deletes predictors subsequently. Terms that does not harm the fit of the model are removed to a point where deletion leads to a significantly poorer fit [1].
Other model selection methods include the exhaustive search, cross validations as well as Bayes factors of various flavours (partial, intrinsic, pseudo, fractional, posterior), Bayesian model averaging to name a few popular methods [18].
This report will focus on the following two approaches for model selectiom: the discrimination approach and the discerning approach.

Firstly within the context of the discrimination approach, two competing models with the same dependent variable will be compared to with other based on the following criteria namely, the Akaike's information criterion, Schwarz's information criterion (SIC), as well as the Mallows's C_p criterion. The criteria rank models according to a score. The score is calculated by introducing a penalty term which is harsher on models that have more parameters. This is in attempt to mitigate the risk of over-fitting. In comparing two models, the information criteria prefer the model which scores a lower value [9].

The Akaike information criterion was introduced by a Japanese Statistician, Hirotugu Akaike in 1973 and formally published in 1974 [2]. Akaike's original work for IID (identical independent distributed) data, however it is also extended to regression type setting. According to Kubokawa and Srivastave [13] the AIC is

developed for selecting the variables of nested error regression models where an unobservable random effect is present. In regression analysis, models could be over-fitted or under-fitted. The AIC is a way to balance the drawbacks of these circumstances. Akaike developed the AIC score for choosing the best model for both in-sampling and out-of sampling forecasting.

The Schwarz's information criterion, also known as the Bayes Information Criterion (BIC), prefers simpler models compared to the AIC, i.e the penalty imposed on models with more regressands is higher. The model selection tool was named after the Israeli professor Gideon Ernst Schwarz, who developed it. His work on criterion was published in 1978 [21]. The criterion has been widely used for model identification in time series and linear regression. Just like the AIC, It has a wide application to any set of maximum likelihood-based models [21].

Another model selection technique we will discuss, is Mallows's C_p criterion. The model selection tool is named after Collin Lingwood Mallows and it was introduced in 1973 [15]. The C_p criterion is used to assess the fit of a regression model that has beem estimated using ordinary least squares. When a model is underfitted, or missing important predictors, it yields biased regression coefficients and biased predictions of the response. The C_p criterion estimates the magnitude of the bias that is present in the predicted responses, as a result of underfitting the model. A high C_p value indicates a large bias. A Model with less bias, hence a lower C_p, is preffered [9].

Secondly within the context of the discerning approach evaluates models taking into account information provided by other contestant models. We need two estimated models, A and B. We then add the dependent variable from Model A as a regressand to Model B. The idea is to evaluate whether the variables unique to Model A, offer more explanatory power to Model B. If we reject the hypothesis that the additional regressand adds explanatory power to Model B, we will prefer the Model B over Model A. The hypothesis tests can be done using the non-nested F-test. There are problems encountered with using only the F-test to test the model. The Davidson and Mackinnon J-test will be discussed as an alternative to the F-test.

The J-test , was developed by Russel Davidson and James MacKinnon on 1981 [4]. The test was developed to test non-nested model specification, however it is widely used to choose between model specifications. Bremmer [3] states that in application, the F-test is prefered over the J-test because it requires estimation of only two regressions while the J-test requires the estimation of four different regressions. However the J-test is commonly used in the literature, it has been cited in just under 500 seperate articles between 1984 and

2004. Another indicator of its increased acceptance in econometric practice is the number of textbooks that discuss this test [3].

Lastly, model validation is possibly the most important step in model building. An important issue is whether the results obtained from a sample can be extrapolated to the population from which the sample was drawn [16]. R^2, is the most common measure of goodness-of-fit with respect to linear regression models. It is routinely given by software packages such as R and SAS. Unfortunately a high R^2 does not necessarily signify that the model fits well. Incidently, there is no one perfect measure of goodness of fit for statistical models, hence we will look at a variety of concepts that fall into the category of goodness of fit, including the adjusted R^2, residual analysis, and the ROC curve.

Furthermore in setion 2, a discussion of the tests for nested and non-nested models will be presented. This section will also include a discussion of model selection criteria as well a few tools for model validation. In section 3, we will use practical examples to illustrate the tests and criteria discussed in the previous section.

2 Background Theory

2.1 Nested vs non-nested models

In this section the difference between nested and non-nested models will be discussed. We will use the t-test as well as the non-nested F-test to test whether a model is nested or non-nested.

2.1.1 Nested models

To test that a model whether a model is nested or non-nested consider the following models:

$$Y_i = \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4i} + \beta_5 X_{5i} + u_i \tag{1}$$

$$Y_i = \beta X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + u_i \tag{2}$$

In Model A, if we test that

$$H_0 : \beta_4 = \beta_5 = 0$$

$$H_1 : \beta_4 \neq 0 \text{ or } \beta_5 \neq 0$$

According to Gujarati and Porter [9] we employ the restricted F-test to test whether the parameters are significant. We compare the value of the test statistics to the critical value. If the F test statictic is smaller than the critical value, we do not reject the null hypothesis. We can therefore conclude that the model (1) can reduce to model (2). In other words, model (2) is nested in Model (1). If we add another variable, X_4, to model (2), model (1) reduces to model (2) if β_5 is equal to zero. This we conclude on the basis of a t-test that the coefficient of X_5 is equal to zero. In regression analysis that may be linear or not, specification error tests, more generally used are the likelihood test ratio, or the Wald test, or the LaGrange multiplier. Because reseachers often work with small and finite samples , the F-test suffices [9].

Now consider the following models:

$$Y_i = \alpha_1 + \alpha_2 X_{2i} + \alpha_3 X_{3i} + u_i \tag{3}$$

$$Y_i = \beta_1 + \beta_2 Z_{2i} + \beta_3 Z_{3i} + v_i \tag{4}$$

where the X's and Z's are different variables. Neither model is the subset of the other. Therefore the models are non-nested, since one cannot be derived from the other. The models may have some regressors in common, for example, X_3 may be included to model (3). Still the models are non-nested because model (4) doesn't have Z_2, Z_3 and model (4) does not have X_2.

2.1.2 Non-nested Models

The non-nested tests of hypotheses arise in situations when the alternative hypothesis cannot be derived as a special case of the null hypothesis. This may happen as a result of either completely different sets of regressors in competing models or different distributions in the stochastic term [20]. Non-nested hypothesis tests provide a way to test model specifications against the evidence made available by one or more of the alternative hypotheses.

To do model selection using the non-nested F-test, suppose we estimate the following nested model:

$$Y_i \quad = \quad \lambda_1 + \lambda_2 X_{2i} + \lambda_3 X_{3i} + \lambda_4 Z_{2i} + \lambda_5 Z_{3i} + u_i \tag{5}$$

Consider model (3) and model (4) in section (2.2.1), model (5) encompases model(3) and (4). It is important to note that model (3) and (4) are non-nested. If model (3) is correct then $\lambda_4 = \lambda_5 = 0$, otherwise if model(4) is correct $\lambda_2 = \lambda_3 = 0$. This testing can be done using the F-test. The F test proceeds as Gujarati and Porter [9] indicate that there are problems encountered using this test procedure .

Firstly, it is the case where multicollinearity is present between the X's and the Z's. If the X's and the Z's are highly correlated then there is great possibility that one or more $\lambda's$ are individually statistically insignificant. However, on the basis of the F-test one could conclude that the coefficients are all statistically insignificant.

Secondly, there is another problem. Suppose we choose model (3) as a reference model and establish that all the coefficients are significant. Then we add Z_2 or Z_3 or both and conclude, on the basis of the F-test, that their additional contribution to the explained sum of squares (SSE) is statistically insignifcant. As a result we choose model(3) to be the correct model. Likewise, if we take model(4) as the reference hypothesis and find that all its coefficents are significant and similarly add X_2 or X_3 or both. If we conclude that the incremental contribution of the additional coefficients, to the ESS is statistictally insignificant, we will therefore decide to choose model(4).

Gujarati and Porter [9] further illustrate that the choice of the reference model may determine the chosen model especially if severe multicollinearity is present in the competing regressors, rendering the F-test meaningless. Because of the problems presented using the non-nested F-test, alternatives have been suggested. One is the Davidson-Mackinnon J-test.

2.2 Model Selection

According to Gilmour [6], if there are k explanatory/ predictor variables there is a possibility of $2^k - 1$ regression models. Computing all the regression models today is possible but selecting a model is still an issue.

In model selection there are m models $M_1, ..., M_m$ where usually $m > 2$. Instead of testing multiple hypotheses with two models at a time, to see if we reject one or the other, it is more convenient to have a criterion to select one of the models. There are two approaches in which model selection is done. Here we will discuss the criteria as well as the tests used to choose between competing models with respect to the two approaches.

2.2.1 The Discrimiantion approach

When we have more than one competing model, several criteria may be used to select the right model. Since the models have the same dependent variable, we can choose between models by using some criteria. The criteria include the Akaike's Information Criteria, Schwarz's Information Criteria and the Mallow's C_p criterion.

Akaike's Information Criterion (AIC)

According to Dziak et al. [10] the AIC estimates the relative Kulberg-Leibler (KL) distance (non-parametric

distance measure) between the likelihood function estimated by a candidate model and the unknown true likelihood function that generated the data. The model that is closest to the truth in terms of the KL would not necesarrily be the best fit to an observed sample, as samples may be arbitrarily well fitted to the data by adding more parameters to the model. A model with the best KL distance will however, describe the population distribution accurately, and hence the future samples.

The KL distance can be written as

$$E_t\left(l_t\left(y\right)\right) - E\left(l_t\left(y\right)\right) \tag{6}$$

where E_t is the expected value under the unknown true distribution function, l is the log likelihood of the data under the fitted model being considered, and l_t is the log-likelihood of the data under the unknown true distribution. $E_t\left(l_t\left(y\right)\right)$ will be the same for all the candidate models, so KL is minimised by choosing a model with the highest $E_t\left(l\left(y\right)\right)$.

The AIC criteria introduces the idea of imposing a penalty . It is defined as

$$AIC = e^{\frac{2k}{n}} \frac{\sum \hat{u}^2}{n} = e^{\frac{2k}{n}} \frac{SSR}{n} \tag{7}$$

where k is the number of parameters and n is the number of observations. For mathematical convenience

$$ln\ AIC = \frac{2k}{n} + ln\left(\frac{SSR}{n}\right) \tag{8}$$

where $ln\ AIC$ is the natural log of AIC and $\frac{2k}{n}$ is the penalty factor. The AIC is smaller for bigger models because of the penalty imposed for overparameterization. This model selection criterion maximizes the predictive accuracy of the chosen probability distribution [12]. There is the familiar discourse that the AIC is not a consistent estimator of the number of parameters of the smallest model containing the true value. Kiessepa [12] validates this claim and adds that this feature of the AIC is compatible with the purpose for which it is used. Models that have a lower AIC are preferred.

Schwarz's Information Criterion (SIC)

In Bayesian model selection, a prior probability is set for each model M_i, and the prior distributions are set for the nonzero coefficients in each model. If we assume that one and only one model, together with its associated priors, is true, we can use Bayes' Theorem to find the posterior probability of each model given the data. Let $Pr\left(M_i\right)$ be the set prior probability, and $Pr\left(\boldsymbol{y}|M_i\right)$ be the probability of the density of the data under M_i, calculated as the expected valued of the likelihood function of $\boldsymbol{y}$ given the model and the parameters, over the prior distribution of parameters. According to Bayes Theorem the posterior probability is proportional to $Pr\left(M_i\right) Pr\left(y|M_i\right)$. The degree to which the data support M_i over another model M_j is

given by the ratio of the posterior odds to the prior odds:

$$\frac{Pr\left(M_i|\boldsymbol{y}\right)/Pr\left(M_j|\boldsymbol{y}\right)}{Pr\left(M_i\right)/Pr\left(M_j\right)}.$$

If we assume equal prior probabilities for each model then this simplifies to the "Bayes factor"

$$B_{ij} = Pr\left(M_i|\boldsymbol{y}\right)/Pr\left(M_j|\boldsymbol{y}\right) = Pr\left(\mathbf{y}|M_i\right)/Pr\left(\boldsymbol{y}|M_j\right)$$

so that the the model with the higher Bayes factor also has the higher posterior probability. Schwarts showed that in different types of models B_{ij}can be roughly approximated by $exp\left(-\frac{1}{2}BIC_i + \frac{1}{2}BIC_j\right)$, where BIC (also called SIC) is 1, especially if a certain "unit information is used for the coefficients". The model with the highest posterior probability is likely the one with the lowest BIC (SIC).

The SIC is given by:

$$SIC = n^{\frac{k}{n}}\frac{\sum \hat{u}^2}{n} = n^{\frac{k}{n}}\frac{SSR}{n} \tag{9}$$

or in log form:

$$ln\ SIC = \frac{k}{n}ln\ n + ln\ \frac{SSR}{n} \tag{10}$$

where $\left[\frac{k}{n}ln\left(n\right)\right]$ is the penalty factor. Comparing eq(6) to eq(4), it is obvious that SIC imposes a harsher penalty than the AIC, it replaces 2 by $log\left(n\right)$ as a multiple of the number of parameters. Hence the selected model has to be less complex than the one selected under the AIC [1]. As noted by Agresti [1] the SIC is derived based on the Bayesian argument for determining the model that will have the highest posterior probability, out of a set of contestant models. When comparing two models based on a SIC score, any difference between the values relates to a Bayes factor. It has a property of selecting the "correct model " with probability converges to 1 as $n \to \infty$. The Bayesian structure provides justification for this approach. Schwarz [21] indicates that the SIC is consistent, meaning that if one of the contestant models has the true value, the SIC will maximize the probability of selecting the correct model, namely the smallest model containing the true value. The AIC is not consistent in this sense. As it may seem that the SIC is preferable, however, in the case that none of the competing models are correct, it is unclear which criteria to use [12].

Mallows's C_p Criterion (C_p)

A good model should have a small mean square error of prediction. The C_p as a measure of bias, is used to compare models that have been estimated using the least square estimator method [9]. Suppose we have a model with k regressors including the intercept. Let $\widehat{\sigma}^2$ be the estimator of the true σ^2. Suppose we only

choose p regressors (i.e $p \leq k$) . The criterion is given as:

$$C_p = \frac{SSR_p}{\hat{\sigma^2}} - (n - 2p) \tag{11}$$

Where SSR_p denotes the regression sum of squares obtained from the regression using p regressors. The definition of the C_p-statistic was intended to ensure that C_p had the expected value p for a model including all possible regressors [6]. If the model does not suffer a lack of fit , such a model will have

$$E\left(SSR_p\right) = (n - p)\,\sigma^2 \tag{12}$$

consequently it is true that

$$E\left(C_p\right) = \frac{(n - p)\,\sigma^2}{\sigma^2} - n + 2p \approx p \tag{13}$$

Gujarati and Porter [9] state that in choosing a model with respect of the C_p criterion, we would choose one that has a low C_p value just about equal to p. So, following the principle of parsimony we select a model with p ($p < k$) regressors that gives a good fit to the data. In practice one would construct a plot of C_p computed using Eq (7) against p. An adequate model will appear as a point near the $C_p = p$ line, as can be seen in the figure (2).

2.2.2 Discerning Approach

Instead of trying to find the best model from a set of competing models, based on some criteria, one can use the approach of gathering information from other models to see if it can improve a given model and vice versa. Suppose that model (3) and model (4) are theoretically plausible for explaining a phenomenon, and both models have the same dependent variable, Y. The J-test proceeds as follows:

1. The model (3) is estimated and the variable and the predicted values of the depended variable , $\hat{y}_i^C$, $i = 1, ..., N$, are obtained.

2. Similar to Step 1, model (4) is is estimated and the predicted values from this model, $\hat{y}_i^D$, $i = 1, ..., N$, are obtained.

3. The predicted values from model (3) are included as an additional regressor in model (4). Likewise the values of model(4) are added as an explanatory variable to model(3), as shown in equation (14) and

equation (15), respectively.

$$Y_i = \beta_1 + \beta_2 Z_{2i} + \beta_3 Z_{3i} + \beta_4 \hat{Y}_i^C + v_i \tag{14}$$

$$Y_i = \alpha_1 + \alpha_2 X_{2i} + \alpha_3 X_{3i} + \alpha_4 \hat{Y}_i^D + u_i \tag{15}$$

4. The idea is to test whether the predicted values from one model add predictive power to the other model. Using the t-test we test whether each of the coefficient $\alpha_4 = 0$ and $\alpha_5 = 0$.

Consider equation (14). If we can accept the model (3) as the true value, that is, if we do not reject the hypothesis that $\alpha_4 = 0$, it would mean that $\hat{Y}_i^D$ included in eq (15), which represents the effect of the predictor variables in model (4), does not offer additional explanatory power beyond that offered by model (3). In other terms model (3) encompasses model (4), in the sense that model(4) does not contain information that will improve the performance of model (3). If we reject the null hypothesis, it would mean that model (3) is not the true model.

On the other hand, if we reject the hypothesis that $\beta_4 = 0$, we therefore conclude that model(3) does not improve the explanatory power of model (4) and thus choose model (4) over model (3). By the same token, if we do not reject the null hypothesis we then prefer the model (3) over model (4).
The J-test may fail to discriminate between the true and false models specification when the alternative hypothesis fits the given data well. Either both specifications will be rejected or neither will be rejected. In this case testing a model agaisnt the evidence provided by the other will not allow the investigator to choose between the two models. It could however indicate that either one or both models have been misspecified. "When both models are rejected we must conclude that neither model is satisfactory, a result that will not be welcome but will perhaps spur us to develop better models" [5]. Godfrey and Pesaran [7] state the following situations where the J test will over reject the true hypothesis: (i) when the true model is poorly fitted (ii) low or moderate correlation of the regressors of the two models; and (iii) when the false model includes more regressors than the true model. Davidson and Mackinnon [14] agree that if (i) and (ii) are obtained as well as in small, finite samples the J test will "over-reject".

Rao, Gali and Krieg [20] provide some theoretical reasons why the J test may lack power in testing model specifications that fits a given data set well. Firstly, when the sample size is small the difference in regressors in the contestant models will influence the size of the test statistic. This is in agreement with the simulation-

based findings of Godfey and Pesaran [7] and also with the conclusion drawn by Gourieroux and Monford that the test is "very sensitive to the relative number of regressors in the two hypothesis; in particular the power of the J test is poor when the number of regressors in the null hypothesis is smaller than the number of regressors in the alternative one" [8]. However the effect of the difference in regressors will become insignificant as sample sizes increase.

When classical procedures lead to inconsistent results there are many non-standard testing of hypotheses used as alternatives. Rao, Gali and Krieg [20] suggest the Bayesian approach which is more consistent and provides meaningful results however there are very few applications of it that have been found.

2.3 Model fit

After selecting a model, model validation should be done to evaluate if the model fits the data well. This is possibly the most important step in model building. Goodness of fit measures are used to assess how well a model fits a given set of data. Pregibon [17] indicates that in practice however, this step is usually neglected and rarely carried out. The basic reasons are

(i) The lack of routine methods.

(ii) The high costs of an analyst and computer time.

A model that is well fitted is one that when it is applied to different data samples it consistently produces reliable estimates and predictions. There are numerical and graphical tools which can be used to assess how well a model fits a given data set. Graphical methods have an advantage over numerical methods as they readily illustrate a broad range of relationships between the model and the data. Numerical methods tend to be narrowly focused on a particular aspect of the relationship between the model and the data and often try to compress that information into a single descriptive number or test result. Nevertheless numerical methods do play an important role as confirmatory methods for graphical techniques. There are also a few modelling situations in which graphical methods cannot be used. Logistic regression with binary data is another area in which graphical methods can be difficult. In these cases numerical methods provide a fallback position for model validation [16].

The R^2 Criterion

The R^2 criterion is a global measure of variance, which is routinely given in some software output such as R and SAS.

We estimate a regression function $E\{Y_i\} = \beta_0 + \beta_1 X_i$ with $\hat{Y} = b_0 + b_1 X_1$ $i = 1, 2, ...n$ where

$$Y_i = response$$

$$E\{Y_i\} = mean\ response$$

$$\hat{Y}_i = fitted\ value$$

The measure of total variation, the total sum of squares is

$$SSTO = \sum_{i=1}^{n} \left(Y_i - \overline{Y}\right)^2$$

The total sum of squares can be decomposed into two components:

$$SSTO = ESS + RSS$$

where

$$ESS = \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2$$

the variation of Y around the fitted regresson line, the error sum of squares and

$$RSS = \sum_{i=1}^{n} \left(\hat{Y}_i - \bar{Y}\right)^2$$

the regression sum of squares.

R^2 also known as the coefficient of determination, measures the goodness of fit of a model. The criterion works for in-sample forecasting, that is, how close the dependent variable of the given data estimates the true value, but does not perform well for out-of-sample forecasting. R^2 is defined as

$$R^2 = \frac{ESS}{SSTO} = 1 - \frac{RSS}{SSTO} \tag{16}$$

where $0 < R^2 < 1$. As per definition, R^2 is a ratio of the error sum of squares over the total sum of squares. The problem with this measure, especially for comparing models of different sizes, is that the sum of squares of the regression, and hence R^2, increase the more variables there are in the model . In addition there is a temptation to add more variables just to increase the score. The result will be a higher R^2 but also a larger variance of the forecast error [9]. For this reason the adjusted $\overline{R}^2$ is used instead, as it takes into account the number of parameters in the model, including the intercept term.

Adjusted R^2

The adjusted R^2 is a measure of variation explained which takes into account the effect of the number of parameters in a model. It features a penalty term for adding more regressors to increase the R^2 value, defined as

$$\overline{R}^2 = 1 - \frac{RSS/(n-k)}{SSTO/(n-1)} = 1 - (1 - R^2)\frac{n-1}{n-k} \tag{17}$$

where n is the sample size and k is the number of regressors

As in the formula, $\bar{R}^2 < R^2$ as a result of penalizing the addition of more regressors.

Residual analysis

Some diagnostic tests for linear regression models, such as the t-test, F-test and R^2 are based on the assumptions that the relationship between the outcomes of the dependent variable and the predictor variables is (approximately) linear and that the error term is normally distributed. If any of these underlying assumptions are violated the fitted model may depart from true value. This departure may not be detected by the R^2 measure of goodness of fit. Residual analysis plots the statistic

$$e_i = y_i - \hat{y} \tag{18}$$

against $\hat{y}$; where y_i is the observed value and $\hat{y}$ is the fitted value. The pattern of the resulting graphical representation is of interest. If the points on the graph form a random pattern the model fits the data well. Any other resulting pattern indicates that the model is a poor fit to the data.

The residual analysis diagnostic test also applies to logistic regression models. Binary logistic regression models how response variable Y depends on a set of k explanatory variables, $X = (X_1, X_2, ...X_k)$. For a binary dependent variable Y and explanatory variable X, let $\pi(x) = P(Y = 1 \mid X = x) = 1 - P(Y = 0 \mid X = x)$. The logistic regression model is

$$\pi(x) = \frac{e^{(\alpha+\beta x)}}{1 + e^{(\alpha+\beta x)}} \tag{19}$$

the logit (log of odds) has a linear relationship

$$logit[\pi(x)] = log\frac{\pi(x)}{1 - \pi(x)} = \alpha - \beta x \tag{20}$$

Let Y_i denote the binomial outcome for n_i where $i = 1, 2..., N$. Let $\hat{\pi}_i$ denote the model estimate $P(Y = 1)$. Then $\hat{\mu}_i = n_i\hat{\pi}_i$. For a general linear model with a binomial linear component, the Pearson residual is

$$e_i = \frac{y_i - n_i\hat{\pi}_i}{\sqrt{v\hat{a}r(Y_i)}} = \frac{y_i - n_i\hat{\pi}_i}{\sqrt{[n_i\hat{\pi}_i(1 - \hat{\pi}_i)]}} \tag{21}$$

divides the residual $y_i - n_i\hat{\pi}_i$ by the estimated binomial standard deviation of y_i.

The Pearson statistic for testing model fit is defined as

$$\chi^2 = \sum_{i-1}^{N} e_i^2 \tag{22}$$

One of the functions of this Pearson χ^2 statistic is to test whether the distibution of events observed in a sample is consistent with a particular theoretical distribution i .e. may detect a lack of fit. This is done

through hypothesis testing, where the null hypothesis will state that there is no difference between the distributions. The null hypothesis is rejected if the test statistic exceeds the critical value.

Although the χ^2 is very useful it does have limitations:

(i) The test does not give information about the strength of the relationshiop between the variables, it only indicates if there is an association.

(ii) The size of the χ^2 is directly proportional to the sample size, and therefore it is sensitive to sample size. This means that strong associations may not be detected as significant when a small sample size is used. Larger sample sizes, on the other hand may yield statistical significance when in actual fact the findings are small and uninteresting.

ROC curve

Another way of assessing the validity of a model, with respect of logistic regresson models, is by splitting the available data into a training and testing samples. The samples are taken from the same population but are distinct and independent of each other. Firstly the training set is used to fit the model. The fitted model is then applied to the testing sample to evaluate the model's performance on it. If the testing data is too small, this might result in an unsatisfactory testing data set. According to Giancristofaro and Salmaso [18] we should expect a lower performance of the model on the testing set.

The ROC (Receiver Operating Characteristics) curve and the area below the ROC curve are commonly used to assess binary response models . The testing set is used to estimate the area below the ROC curve. The ROC curve is created by plotting the probability of correctly classifying a positive subject against the probability of incorrectly classifying a negative subject. These probabilitites can be computed using the model's regression equation. The area below the ROC curve ranges between 0 and 1. A curve will be judged according to its ability to measure the predictive error. That is the ability of the model to distinguish between subjects with different responses.

3 Application

This section has three parts. Firstly, we do hypothesis testing for nested hypotheses. The second part deals with model selection of non-nested hypotheses, with respect to the discerning approach and the discrimnation approach. Lastly, we will look at model fit tests.

3.1 Tests of nested models

The models given by (23) and (24) explain major league baseball players' salaries. The data was obtained from the sas.help database.

In this example

$Y = salary\, of\, major\, league\, baseball\, players$

$X_1 = years\, in\, major\, league$

$X_2 = times\, at\, bat\, in$ 1986

$X_3 = hits\, in$ 1986

$X_5 = home\, runs$

$X_6 = runs\, in$ 1986

$X_7 = RBIs\, in$ 1986

$X_8 = walks\, in$ 1986

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_5 x_5 + \beta_6 X_6 + \beta_7 X_7 + \beta_8 X_8 + \varepsilon \tag{23}$$

is the unrestricted original model.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_5 x_5 + \beta_7 X_7 + \epsilon \tag{24}$$

A restriction is placed on (23) to produce (24), a restricted model because we had to control for β_3, β_6, β_7. The hypothesis test proceed as follows:

$$H_0 : \beta_3 = \beta_6 = \beta_8 = 0$$

$$H_1 : \beta_3 \neq 0\ or\ \beta_6 \neq 0\ or\ \beta_8 \neq 0$$

From output (i) in table 1, it can be seen that the F statistic= 3.39 corresponds to a p-value= $0.0212 < 0.05$. At a 5% level of significance we do not reject the null hypothesis. There is enough evidence to conclude that the model (24) is nested within model (23).

If we add X_7 to model (24), then model (23) will reduce to model (24) if $\beta_8 = 0$. Using the t-test we test that

$$H_0 : \beta_8 = 0$$

$$H_1 : \beta_8 \neq 0$$

On the basis of the t-test with a $p - value = 0.0012$, shown in the output in table 1(ii), we do not reject the null hypothesis at a level of significance $\alpha = 0.05$. It can be concluded that if X_7 is added to model (18) , model (18) reduces to model (17) given that $\beta_8 = 0$.

Test 1 Results for Dependent Variable y				
Source	DF	Mean Square	F Value	Pr > F
Numerator	3	1.12981	3.29	0.0212
Denominator	255	0.34322		

Table 1: Nested hypothesis output: Unrestricted F-test

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	4.13133	0.12972	31.85	<.0001
x1	1	0.09840	0.00787	12.51	<.0001
x2	1	-0.00166	0.00097205	-1.71	0.0880
x3	1	0.01278	0.00365	3.50	0.0005
x5	1	0.00557	0.00958	0.58	0.5614
x6	1	-0.00087622	0.00446	-0.20	0.8446
x7	1	-0.00005292	0.00423	-0.01	0.9900
x8	1	0.00812	0.00248	3.27	0.0012

Table 2: Nested hypotheses output : Parameter estimates

3.2 Tests of non-nested hypotheses

Consider the following two models:

$$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + {}_5x_5 + \varepsilon \tag{25}$$

$$y = \beta_0 + \beta_6 x_6 + \beta_7 x_7 + \beta_8 x_8 + \upsilon \tag{26}$$

Note that *model* (25) and (26) are subsets of *model* (23). These models are non-nested. Each model has predictor variables that uniquely discribe the dependent variable, i.e. both models involve the same dependent variable.

3.2.1 Information criteria

To test these two models we compare the following criteria, namely, the AIC, BIC as well as Mallows's C_p criterion. A model is chosen on the basis of one or more these criteria. Table(2) and table(3) shows the summaries of the fit diagnostics of *model* (25) and *model* (26) respectively. *Model* (25) has lower AIC and BIC scores compared to *model* (26). The C_p criteria however, had different results. *model* (25)'s $C_P = 5$, turned out to be higher than that of *model* (21), $C_p = 4$. Two out of the three criteria are in favour of *model* (25). In this case. *model* (25) is prefered over *model* (26). Therefore we choose *model* (25) to be the right model.

Observations	263
Parameters	5
Error DF	258
MSE	0.3486
R-Square	0.5659
Adj R-Square	0.5591
AIC	-272.2
BIC	-270
CP	5

Table 3: Non-nested model selection diagnostics output: Model (25)

Observations	263
Parameters	4
Error DF	259
MSE	0.5657
R-Square	0.2928
Adj R-Square	0.2846
AIC	-145.9
BIC	-143.8
CP	4

Table 4: Non-nested model selection diagnostics output : Model (26)

3.2.2 Davidson MacKinnon J-test

For the J-test we use models (25) and (26) to do the following hypothesis test

Parameter Estimates						
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	0.10224	0.36117	0.28	0.7773
x6		1	-0.00224	0.00256	-0.87	0.3833
x7		1	-0.00123	0.00234	-0.52	0.6002
x8		1	0.00794	0.00230	3.45	0.0006
yhat1	Predicted Value of y	1	0.95867	0.07089	13.52	<.0001

Table 5: Model selection, non-nested Mackinnon J-test : Model (25) parameter estimates output

$$H_0 : Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_5 X_5 + \beta_6 \hat{y}^{(25)}$$

$$H_0 : Y = \beta_0 + \beta_6 X_6 + \beta_7 X_7 + \beta_8 X_8 + \beta_9 \hat{y}^{(26)}$$

where $\hat{y}^{(25)}$ and $\hat{y}^{(25)}$ are fitted values of model (25) and model (26) respectively.

The output in table (6) shows the estimator for $\alpha_6 = 0.55472$, with a p-value of 0.0009. Thus we do not reject H_0 in favor of H_1. In table (5), the output from the of model (26) gives the parameter estimate for $\beta_9 = 0.95867$

with a p-value of 0.0001. Thus we do not reject the alternative hypothesis either. model (25) improves the explanatory power model (26) and vice versa.

Parameter Estimates						
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	1.46568	0.80409	1.82	0.0695
x1		1	0.09818	0.00769	12.77	<.0001
x2		1	-0.00166	0.00097218	-1.71	0.0886
x3		1	0.01084	0.00300	3.62	0.0004
x5		1	-0.00482	0.00613	-0.79	0.4326
yhat2	Predicted Value of y	1	0.55472	0.16519	3.36	0.0009

Table 6: Model selection, non- nested Mackinnon J-test : Model (26) parameter estimates output

3.3 Model fit

First we look at the R^2 and the $Adj\ R^2$ obtained from the output in table (2), generated from the 1986 Baseball data. In addition the Titanic dataset will be used to illustrate the ROC curve method of model validation. A logistic regression model is fitted to the data to predict the the number of survivors of the shipwreck.

3.3.1 R^2 and $\bar{R}^2$

Consider the outputs of *model* (23) and *model* (24) in table (3) and table (4) respectively. The R^2 of model (23) is calculated to 0.569; that is to say that only 57 percent of the variation in the dependent variable can be explained by the predictor variable. $\bar{R}^2 = 0.5591$, which penalises for additional parameters. The R^2 as well as $\bar{R}^2$ for *model* (24), a competing model, are 0.2928 and 0.2846 respectively. Usually an R^2 of 70 percent is considered to be good. In this case neither of the models are an excellent fit to the data, however it can be seen that *model* (23) is a better fit than *model* (24), because it has a higher $\bar{R}^2$ value.

3.3.2 The ROC curve

The Titanic data set is split into a training set and a testing set using a $80 : 20$ ratio. The fitted model has a binary dependent variable i.e. survived vs did not survive. The predictor variables: class, age, gender, parch, sibs are used to observe how they affect the outcome.

To test the performance of the model, first it is applied to the training set and then after, to the testing set. The objective of the test is to determine how well the model will correctly classify the dependent variable on the "unseen" data i.e the training set. The following are models used to classify whether a passenger survived or did not survive:

$Y = Survive$

$X_1 = Class$

$X_2 = Age$

$X_3 = Gender$

$X_4 = Parch$

$X_5 = Sibs$

where "parch" stands for a passenger who is a parent who had children aboard the ship and "sibs" is the number of siblings a passenger has aboard the Titanic.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon \tag{27}$$

$$Y = \beta_0 + \beta_2 X_2 + \beta_4 X_4 + \beta_5 X_5 + \epsilon \tag{28}$$

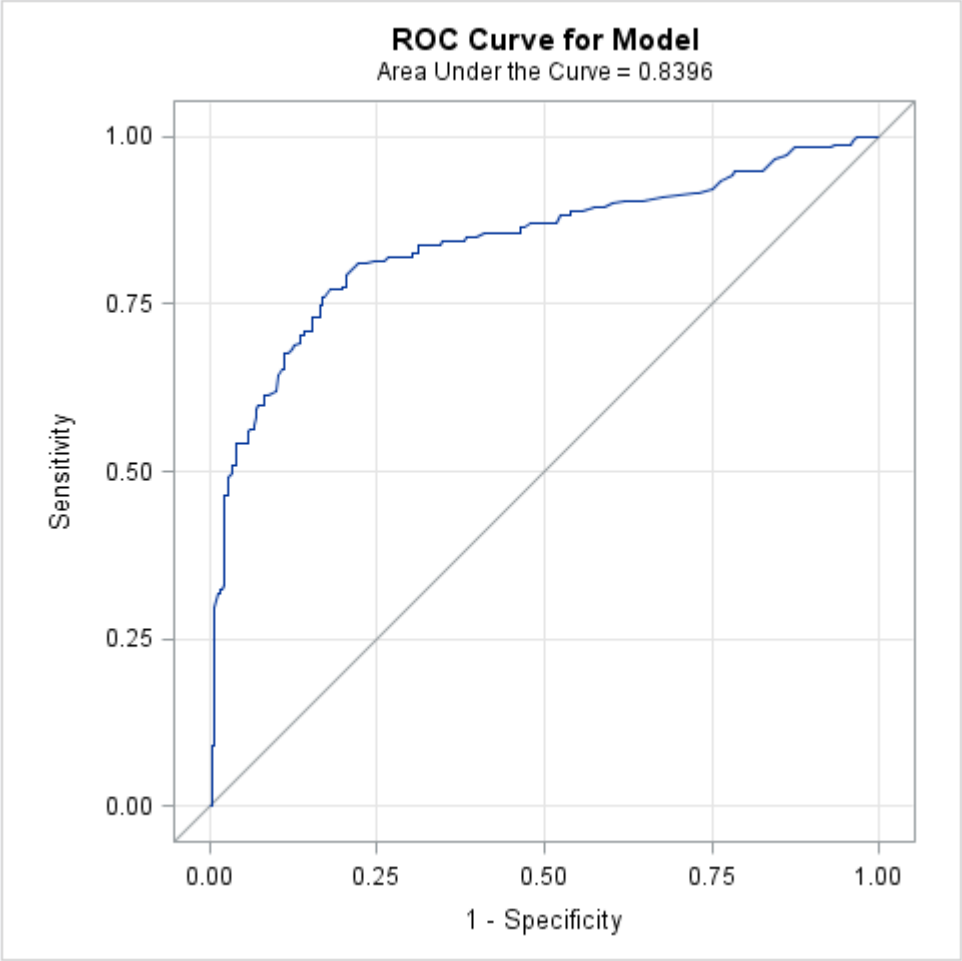


Table 7: Logistic regression output, ROC curve: *Model* (27)

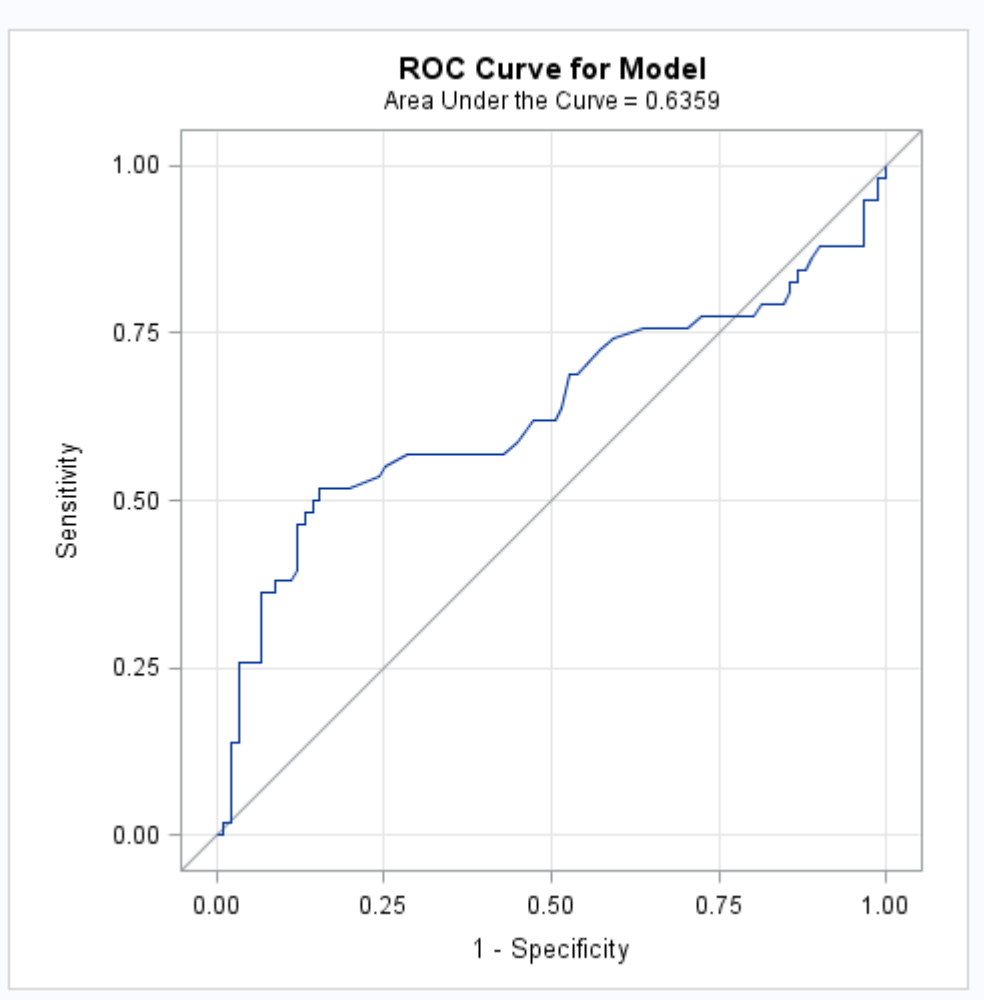


Figure 1: Logistic regression output, ROC curve:*Model* (28)

The output in figures (1) and (2) a show ROC curves that were used to test the ability of the model to distinguish between different responses. The testing set was used to estimate the area below the curves. The area between 0 and 1. The ROC curves are outputs of models that were estimated using different sets of variables. It can be seen that *model* (27), with are under the curve = 0.8940, performs better than *model* 28 in classifying the outcomes. Therefore we conclude that *model* (27) has a higher predictive accuracy, and

thus a better fit.

3.3.3 Residual Analysis

With reference to model (27) and model (28), suppose we want to further investigate whether the models fit the distribution from which the data was generated. The hypothesis is formulated as follows:

$$H_0 : The\ model\ fits$$

$$H_1 : The\ model\ lacks\ fit$$

This test is done on the basis of $\chi^2_{0.05}$. Table (8) and table (9), below show output statistics for the test. Using the Pearson statistic in table (8), on a 5% level of significance, and p-value 0.8379 >0.05 , the decision is to not reject the H_0 . There was enough evidence to conclude that model (27) is a good fit.
Consider the output in table (9). At a 5% level of significance we conclude that model (28) lacks fits . This is based on the evidence provided by p-value = 0.0214 < 0.0 5.

Deviance and Pearson Goodness-of-Fit Statistics				
Criterion	Value	DF	Value/DF	Pr > ChiSq
Deviance	90.7368	110	0.8249	0.9095
Pearson	92.1679	110	0.8379	0.8904

Table 8: Goodness of fit statistics: Model (27)

Deviance and Pearson Goodness-of-Fit Statistics				
Criterion	Value	DF	Value/DF	Pr > ChiSq
Deviance	127.4706	90	1.4163	0.0058
Pearson	119.2063	90	1.3245	0.0214

Table 9: Goodness of fit statistics: Model (28)

4 Conclusion

This essay set out to distinguish between nested and non-nested models. A restricted F-test was used to test the hypothesis that one model is nested within another. By examining the 1986 Baseball data we concluded that the two modules that were fitted to the data were nested, as the restriction placed on the one model, reduced the model to the other specified model. Models are non-nested if it is not possible to derive one model from another. According to the discrimination approach, model selection criteria are used to discriminate among non-nested competing models. A model is selected if it scores lower than other contestant models with respect of a given criteria. In the practical example we considered two models to illustrate this approach. Two out of three criteria scores (the AIC and the BIC) were in favour of $model(24)$, while the results observed with the Cp score would have led one to choose different. $Model(24)$ is chosen based on the popular vote. In this essay's illustration of the discerning approach we had a case where we could not reject neither the H_0 nor the H_1. Both Models had additional information which improved the other models explanatory power. These results may be improved by developing better models since the sample used was very large.

Models can be built for forecasting and/ or predicting certain outcomes. Depending on the purpose to which the model is to be used, after selecting a model, its important to validate the model's predictive power and/or accuracy.

Several model fit tests were used to assess how well the selected models fit the data. It was apparent that the selected models fit the data better the contestant models. The observed $\bar{R}$ and R^2 of $model\,(24)$ is far below 70%, but it is the higher of the two models and therefore a better fit. We looked at the ROC curve technique as well . The area below the curve is used to assess the predictive power of the model. In the example $model(27)$ shows that it will classify the data correctly 89% of the time. And finally residuals analysis technique of testing goodness of fit yielded resuslts that were consistent with the previous technique's results. Althought it may not be practical or feasible for researchers to do more that one out-of sample goodness-of-fit test, to maximise a chosen model's performance at least one test should be conducted.

References

[1] A. Agresti. *Categorical Data Analysis*. John Wiley & Sons Incorporated, 3rd edition, 2013.

[2] H. Akaike. A new look at statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):714–723, 1974.

[3] D. S. Bremmer. J-tests: To nest or not to nest, that is the question. In *Quantitative Methods*, 2003.

[4] R. Davidson and J. MacKinnon. Several tests for model specification in the presence of alternative hypothesis. *Econometrica*, 49(3):781–793, 1981.

[5] R. Davidson and J. Mackinnon. *Econometric theory and methods*. Oxford University Press, 2004.

[6] S.G Gilmour. The interpretation of mallow's c_p statistic. *The Statistician*, 45(1):49–56, 1996.

[7] L. Godfrey and M. Pesaran. Test of non-nested regression models. *Journal of Econometrics*, 21(1):133–154, 1983.

[8] C. Gourieroux and A. Monfort. *Handbook of Econometrics*. Elsevier, 1994.

[9] D.N. Gujarati and D.C. Porter. *Basic Econometrics*. McGraw-Hill/Irwin, 5 edition, 2009.

[10] S. Lanza J. Dziak, D. Coffman and R. Li. Sensitivity and specificity of information. pages 2–5, 2012.

[11] J. B. Kadane and N. A. Lazar. Methods and criteria for model selection. *Journal of the American Association*, 99(465):279–290, 2004.

[12] I. A. Kieseppa. Aic and large samples. *Philosophy of Science*, 70(5):1265–1276, 2003.

[13] T. Kubokawa and M. S. Srivastava. Akaike information criteria for selecting variables in the non-nested error regression model. *Communication in Statistics; Theory and Methods*, 41(15):2626–2642, 2012.

[14] J. Mackinnon. Model specification tests against non-nested alternatives. Queen's economics Department Working paper.

[15] C. L. Mallows. Some comments on c_p. *Technometrics*, 15(4):661–675, 1973.

[16] NIST/SEMATECH. e-handbook of statistics methods, April 2012.

[17] D. Pregibon. Logistic regression diagnostics. *Annals of Statistics*, 9(4):705–724, 1981.

[18] L. Salmaso R. Arborretti Giancristofaro. Model performance and analysis and model validation in logistic regression. *Statistica*, 63(2):366–396, 2003.

[19] H. Keselman S. Olejnik. Using wherry's adjusted r^2 and mallow's c_p for model selection from all possible regressions. *The Journal of Experimental Education*, 68(4):365–380, 2000.

[20] M. Ghali S. Rao and J. Krieg. On the j test for non-nested hypotheses and bayesian extension. *MPRA*, pages 3–, January 2008.

[21] G. E. Schwarz. Estimating the dimensions of a model. *Annals of Statistics*, 6(2):461–464, 1978.

Appendix

A Data sets

Sashelp.Baseball

Alphabetic List of Variables and Attributes				
#	Variable	Type	Len	Label
10	CrAtBat	Num	8	Career Times at Bat
15	CrBB	Num	8	Career Walks
11	CrHits	Num	8	Career Hits
12	CrHome	Num	8	Career Home Runs
14	CrRbi	Num	8	Career RBIs
13	CrRuns	Num	8	Career Runs
23	Div	Char	16	League and Division
17	Division	Char	8	Division at the End of 1986
16	League	Char	8	League at the End of 1986
1	Name	Char	18	Player's Name
18	Position	Char	8	Position(s) in 1986
22	Salary	Num	8	1987 Salary in $ Thousands
2	Team	Char	14	Team at the End of 1986
9	YrMajor	Num	8	Years in the Major Leagues
24	logSalary	Num	8	Log Salary
20	nAssts	Num	8	Assists in 1986
3	nAtBat	Num	8	Times at Bat in 1986
8	nBB	Num	8	Walks in 1986
21	nError	Num	8	Errors in 1986
4	nHits	Num	8	Hits in 1986
5	nHome	Num	8	Home Runs in 1986
19	nOuts	Num	8	Put Outs in 1986
7	nRBI	Num	8	RBIs in 1986
6	nRuns	Num	8	Runs in 1986

Obs	Name	Team	nAtBat	nHits	nHome	nRuns	nRBI	nBB	YrMajor	CrAtBat	CrHits	CrHome	CrRuns	CrRbi	CrBB	League	Division	Position	nOuts	nAssts	nError	Salary	Div	logSalary
1	Allanson, Andy	Cleveland	293	66	1	30	29	14	1	293	66	1	30	29	14	American	East	C	446	33	20	.	AE	.
2	Ashby, Alan	Houston	315	81	7	24	38	39	14	3449	835	69	321	414	375	National	West	C	632	43	10	475.00	NW	6.16331
3	Davis, Alan	Seattle	479	130	18	66	72	76	3	1624	457	63	224	266	263	American	West	1B	880	82	14	480.00	AW	6.17379
4	Dawson, Andre	Montreal	496	141	20	65	78	37	11	5628	1575	225	828	838	354	National	East	RF	200	11	3	500.00	NE	6.21461
5	Galarraga, Andres	Montreal	321	87	10	39	42	30	2	396	101	12	48	46	33	National	East	1B	805	40	4	91.50	NE	4.51634
6	Griffin, Alfredo	Oakland	594	169	4	74	51	35	11	4408	1133	19	501	336	194	American	West	SS	282	421	25	750.00	AW	6.62007
7	Newman, Al	Montreal	185	37	1	23	8	21	2	214	42	1	30	9	24	National	East	2B	76	127	7	70.00	NE	4.24850
8	Salazar, Argenis	Kansas City	298	73	0	24	24	7	3	509	108	0	41	37	12	American	West	SS	121	283	9	100.00	AW	4.60517
9	Thomas, Andres	Atlanta	323	81	6	26	32	8	2	341	86	6	32	34	8	National	West	SS	143	290	19	75.00	NW	4.31749
10	Thornton, Andre	Cleveland	401	92	17	49	66	65	13	5206	1332	253	784	890	866	American	East	DH	0	0	0	1100.00	AE	7.00307
11	Trammell, Alan	Detroit	574	159	21	107	75	59	10	4631	1300	90	702	504	488	American	East	SS	238	445	22	517.14	AE	6.24832
12	Trevino, Alex	Los Angeles	202	53	4	31	26	27	9	1876	467	15	192	186	161	National	West	C	304	45	11	512.50	NW	6.23930
13	Van Slyke, Andy	St Louis	418	113	13	48	61	47	4	1512	392	41	205	204	203	National	East	RF	211	11	7	550.00	NE	6.30992
14	Wiggins, Alan	Baltimore	239	60	0	30	11	22	6	1941	510	4	309	103	207	American	East	2B	121	151	6	700.00	AE	6.55108
15	Almon, Bill	Pittsburgh	196	43	7	29	27	30	13	3231	825	36	376	290	238	National	East	UT	80	45	8	240.00	NE	5.48064

Titanic

A	B	C	D	E	F	G	H	I
P_Id	surv	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
1	0	3	male	22	1	0	7.25	S
2	1	1	female	38	1	0	71.2833	C
3	1	3	female	26	0	0	7.925	S
4	1	1	female	35	1	0	53.1	S
5	0	3	male	35	0	0	8.05	S
6	0	3	male		0	0	8.4583	Q
7	0	1	male	54	0	0	51.8625	S
8	0	3	male	2	3	1	21.075	S
9	1	3	female	27	0	2	11.1333	S
10	1	2	female	14	1	0	30.0708	C
11	1	3	female	4	1	1	16.7	S
12	1	1	female	58	0	0	26.55	S
13	0	3	male	20	0	0	8.05	S
14	0	3	male	39	1	5	31.275	S
15	0	3	female	14	0	0	7.8542	S
16	1	2	female	55	0	0	16	S
17	0	3	male	2	4	1	29.125	Q
18	1	2	male		0	0	13	S
19	0	3	female	31	1	0	18	S
20	1	3	female		0	0	7.225	C
21	0	2	male	35	0	0	26	S
22	1	2	male	34	0	0	13	S
23	1	3	female	15	0	0	8.0292	Q
24	1	1	male	28	0	0	35.5	S
25	0	3	female	8	3	1	21.075	S
26	1	3	female	38	1	5	31.3875	S

B SAS code

Model selection

```
data a;
set sashelp.baseball;
y=logsalary;
x1=yrmajor;
x2=natbat;
x3=nhits; x5=nhome;
x6=nruns;
x7=nrbi;
x8=nbb;
run;
/*nested models hypothesis testing, restricted f-test*/
proc reg data=a;
modlsel: model y=x1 x2 x3 x5 x6 x7 x8;
test x3=0, x5=0, x7=0;
run;
/*non-nested model selection*/
proc reg plots=diagnostics (stats=(default aic bic cp));
model1: model y= x1 x2 x3 x5;
run;
proc reg plots=diagnostics (stats=(default aic bic cp));
model2: model y= x6 x7 x8; run;
```

MacKinnon Jtest

```
/* Mackinnon Jtest*/
/*model1*/
ods graphics off;
proc reg data=a;
model1:model y= x1 x2 x3 x5;
output out=b pred=yhat; run;
/*model2*/
ods graphics off;
proc reg data=a;
model2:model logsalary=x6 x7 x8 ;
output out=c pred=yhat1;
run; quit;
data jtest;
set b;
run;
proc reg data=jtest;
jtest: model logsalary=x1 x2 x3 x5 yhat1 ;
run;
data jtest1;
set c;
run;
proc reg data=jtest1;
jtest1: model logsalary= x6 x7 x8 yhat;
run;
```

Logistic regression

```
data train; set titan (obs=718);
run;
data test;
set titan(firstobs=719);
run;
proc logistic data=train desc plots=roc;
class class gender embark;
model survive=class age gender;
run;
proc logistic data=test desc plots=roc;
class class gender embark;
model survive=class age gender/ influence;
run;
proc logistic data=train c;
class class gender embark;
model survive=age parch sibs;
run;
proc logistic data=test desc plots=roc;
class class gender embark;
model survive= age parch sibs/influence;
run;
```

Logistic regression (residual analysis)

```
/*Residual analysis*/
proc logistic data=train desc plots=roc;
class class gender embark;
model survive=class age gender;
run;
proc logistic data=test desc plots=roc;
class class gender embark;
model survive(desc)=class age gender/ scale=none;
run;
proc logistic data=train c;
class class gender embark;
model survive=age parch sibs;
run;
proc logistic data=test desc plots=roc;
class class gender embark;
model survive(desc)= age parch sibs/scale=none;
run;
```

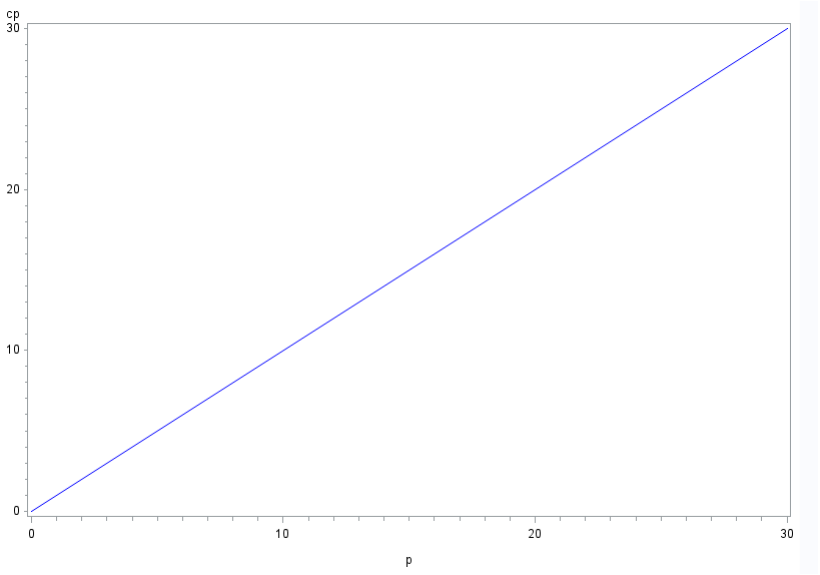


Figure 2: C_p plot

Goodness-of-fit tests for normality

Reginald Sethosa 14051304

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor(s): Dr Paul J van Staden

Department of Statistics, University of Pretoria

30 October 2017

Abstract

Goodness-of-fit tests are conventional statistical tests that are typically used for collating a random sample with a theoretical probability distribution. This report will specifically study how goodness-of-fit tests are used to compare random samples to the normal distribution. Commonly used goodness-of-fit tests such as the Kolmogorov-Smirnov test, the Cramér-von Mises test, the Anderson-Darling test, the Shapiro-Wilk test and the Jarque-Bera test will be investigated. A comprehensive power comparison on goodness-of-fit tests will be discussed in this report. This study will particularly focus on distributions that are not symmetric mesokurtic distributions to assess the aforementioned goodness-of-fit tests.

Declaration

I, *Madimetja Reginald Sethosa,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Madimetja Reginald Sethosa

_ _

Dr Paul J van Staden

_ _

30 October 2017

Acknowledgements

I acknowledge the financial support from National Student Financial Aid Scheme (NSFAS) for the National Research Fund, NRF. I would like to thank my supervisor, Dr Paul van Staden, for pointing me in the right direction for my research. Lastly, I would like to thank my family and friends for all the support during the whole research period.

Contents

List of Figures

List of Tables

1 Introduction

Inferential statistics, unlike descriptive statistics, is classified into two main sections, namely parametric and non-parametric statistics. Furthermore, Thas (2010), [21], reiterates that parametric tests tend to be more reliable than non-parametric tests, because parametric tests are based on a known distribution with it's own assumptions. A fundamental statistical assumption in parametric statistics is the normal distribution, especially in parameter estimation and hypothesis testing [6]. When data does comply with a normal distribution, parametric tests will have more power and efficiency than non-parametric tests [6]. To determine whether a specific data set follows a normal distribution, graphical methods such as histograms, box plots, PP plots and QQ plots can be utilized to get a graphical representation of the shape of the distribution sample observations. These methods are visually effective in summarising a data set, however they are not objective tools. Statisticians have proposed several tests, called goodness-of-fit tests, that can be used to check the form of a distribution. These hypothesis tests determine whether a set of data conforms with a distribution under the null hypothesis against the alternative hypothesis that the data set does not conform with the distribution. This study will focus on goodness-of-fit tests for normality and will examine the power of these tests via a simulation study conducted in Mathematica and SAS.

The goodness-of-fit tests that are to be discussed in this report are the Kolmogorov-Smirnov test, the Cramér-von Mises test, the Anderson-Darling test, the Shapiro-Wilk test and the Jarque-Bera test. The Kolmogorov-Smirnov, the Cramér-von Mises and the Anderson-Darling tests can be applied to any probability distribution, whereas the Shapiro-Wilk and Jarque Bera tests have been specifically developed to test for normality. Thas (2010), [21], describes the Kolmogorov-Smirnov test, which originates from the work of Kolmogorov [11] and Smirnov [19]. This test uses the divergence function between the hypothesised distribution function and the empirical distribution function to assess how well a data fits a distribution [21]. The Anderson-Darling test, introduced by Anderson and Darling [1], is also based on the empirical distribution function. It calculates a test statistic that uses an integral with a weight function [2]. A special case of the Anderson Darling test arises when the weight function equals 1, simplifying the Anderson-Darling test statistic to a statistic that is now recognised and referred to as the Cramér-von Mises statistic. The resulting test originates from the work of Cramér [23] and von Mises[22]. Shapiro (1965), [15], proposed an analysis of variance test for normality. Jarque and Bera (1987), [8], developed a test for normality, which is based on the shape, specifically the skewness and kurtosis, of the data set.

In SAS, the UNIVARIATE procedure is a Base SAS procedure that provides descriptive statistic measures and properties of the distribution of the data set. The UNIVARIATE procedure examines and calculates the respective test statistic and p-values of the Kolmogorov-Smirnov test, the Cramér-von Mises test, the Anderson-Darling test and the Shapiro-Wilk test. The AUTOREG procedure in SAS/ETS

gives the Jarque-Bera test statistic value and p-value. In Mathematica, the Kolmogorov-Smirnov test, the Anderson-Darling test, the Cramér-von Mises test, the Shapiro-Wilk test and the Jarque-Bera test are available for univariate data under distribution fit tests. The tests return the p-value by default and Mathematica allows the tests to be modified so that they return test statistic values and descriptions of the test conclusions.

2 Background Theory

2.1 Moments and L-moments

Moments and L-moments are typically used to describe the location and the spread of a probability distribution. For a random variable X , two categories of moments exist, namely the raw moments and the central moments. The definitions of the raw moments and central moments are given in Definition 1 and Definition 2 respectively.

Definition 1. The raw moments (commonly referred to as moments about 0) of a distribution are defined as $u_i^{'} = E(X^i)$. Hence, the raw moments are computed as follows

$$u_i^{'} = \int_{-\infty}^{\infty} x' f(x) dx$$

for a continuous distribution with a probability denstity function $f(x)$ and

$$u_i^{'} = \sum_{k=0}^{n} x_k^{'} p_k$$

for a discrete distribution with a probability mass function p_i [3].

Definition 2. The central moments (commonly referred to as moments about the mean) of a distribution are defined as $u_i = E([X - u]^i)$. Hence, the central moments are computed as follows

$$u_i = \int_{-\infty}^{\infty} (x - u)^i f(x) dx \quad (i \geq 2)$$

for a continuous distribution with a probability denstity function f(x) and

$$u_i = \sum_{k=0}^{n} (x_k - u)^i p_k \quad (i \geq 2)$$

for a discrete distribution with a probability mass function p_i [3].

The first raw moment is conventionally known as the mean of X. The second central moment is the variance of X hence it is an indication of the spread of the distribution of X. Note that the lower central moments are directly related to the variance, skewness and kurtosis of the random variable X. It can be noted that the second, third and fourth central moments can be expressed in terms of the raw moments as follows:

$$u_2 = u_2^{'} - u^2 = \sigma^2$$

$$u_3 = u_3^{'} - 3uu_2^{'} + 2u^3$$

$$u_4 = u_4^{'} - 4uu_3^{'} + 6u^2u_2^{'} - 3u^4$$

Other popular measures that are used to describe the shape of a distribution are the Pearson coefficient of skewness and the Pearson coefficient of kurtosis. Definition 3 provides a definition of the two measures.

Definition 3. The Pearson coefficient of skewness is defined as

$$\alpha_3 = \frac{\mu_3}{\mu_2^{1.5}} = \frac{\mu_3}{\sigma^3}$$

while the Pearson coefficient of kurtosis is defined as

$$\alpha_4 = \frac{\mu_4}{\mu_2^2} = \frac{\mu_4}{\sigma^4}$$

The Pearson coefficient of skewness gives an indication of the skewness of a distribution. Skewness gives a measure of how symmetric the observations are about the mean [4]. The Pearson coefficient of kurtosis gives an indication of the peakedness and tail-weight of a distribution. Kurtosis gives a measure of the thickness in the tails of a probability density function. Literary work compiled by Hosking (1990),[7], describes L-moments and the work encapsulates theoretical results and techniques described by Sillito (1951,1964,1969). The definition of L-moments is widely accepted as the expectation of a linear combination of order statistics. A formal definition of L-moments is given in Definition 4.

Definition 4. Given that $X_{1:n} \leq X_{2:n} \leq ... \leq X_{n:n}$ denotes order statistics for a random sample of size n from the distribution of X, the r^{th} order L-moment of X is defined as

$$L_r = r^{-1} \sum_{k=0}^{r-1} (-1)^k \begin{pmatrix} r-1 \\ k \end{pmatrix} E(X_{r-k:r}) \quad r = 1, 2, 3, ..$$

Expressions for the first four L-moments are given in Table 1.

Table 1: First four L-moments

Moment	Expression
L_1	$E[X]$
L_2	$\frac{1}{2}E[X_{2:2} - X_{1:2}]$
L_3	$\frac{1}{3}E[X_{3:3} - 2X_{2:3} + X_{1:3}]$
L_4	$\frac{1}{4}E[X_{4:4} - 3X_{3:4} + 3X_{2:4} - X_{1:4}]$

The first two L-moments are mostly used and by convention, L-moments of order 3 or more are usually standardised and then used in the definitions of L-moment ratios. The first two L-moment ratios of a distribution are of particular interest because they are also used to describe the shape of a distribution. A formal definition of the L-moment ratios is outlined in Definition 5.

Definition 5. L-moment ratios are denoted by τ_r and are defined as

$$\tau_r = \frac{L_r}{L_2} \ , r = 3, 4, 5, ...$$

The first two L-moment ratios, which are utilised the most, are referred to as L-skewness (τ_3) and L-kurtosis (τ_4). Expressions of the L-skewness and $L-$kurtosis are given as

$$\tau_3 = \frac{\frac{1}{3}E[X_{3:3} - 2X_{2:3} + X_{1:3}]}{\frac{1}{2}E[X_{2:2} - X_{1:2}]}$$

and

$$\tau_4 = \frac{\frac{1}{4}E[X_{4:4} - 3X_{3:4} + 3X_{2:4} - X_{1:4}]}{\frac{1}{2}E[X_{2:2} - X_{1:2}]}$$

respectively.

Beyond their simplicity, L-moments are preferred over the traditional moments because L-moment ratios are bounded. This property gets rid of the complex interpretation element when working with the L-moment ratios of a distribution. Work done by Hosking (1990) and Jones (2004) proves that the L-skewness ratio is bounded as follows

$$-1 \leq \tau_3 \leq 1$$

while the L-kurtosis ratio is bounded as follows

$$\frac{1}{4}\left(5\tau_3^2 - 1\right) \leq \tau_4 \leq 1$$

2.2 The normal distribution

The normal distribution, or sometimes referred to as the gaussian distribution, is one of the most important and useful continuous probability distribution in probability theory. The central limit theorem makes the normal distribution very useful because it allows the distribution to be used to describe natural and social random variables with unknown distributions. The central limit basically states that for an adequately large sample size n, the sample average of independently drawn observations will converge in distribution to the gaussian distribution[3]. Using customary notation, if X is a real-value random variable, the normal distribution is a continuous distribution that is defined for all x values, it is symmetrical about the mean and it is bell shaped. A random variable X that is normally distributed, denoted as $X \sim N(u, \sigma^2)$, has the probability density function

$$f(x; u, \sigma^2) = \frac{1}{\sqrt{2\Pi\sigma^2}} e^{-(x-u)^2/2\sigma^2}$$

where

$$E(X) = u$$

and

$$var(X) = \sigma^2$$

are the location parameter and scale parameter respectively [9]. It is known that the location and scale parameters sufficiently specify the normal distribution. When the location parameter and scale parameter are 0 and 1 respectively, X is said to have a standard normal distribution, denoted as $X \sim N(0, 1)$. The probability density function of the standard normal distribution is

$$f(x; 0, 1) = \frac{1}{\sqrt{2\Pi}} e^{-x^2/2}$$

for a random variable X [9]. The location parameter of the normal distribution indicates where the distribution is centered and a change in the location parameter shifts the probability density function left or right on the horizontal axis. The scale parameter describes the spread of the distribution hence a larger σ^2 value produces a probability density function that is very wide as compared to a smaller value of σ^2 which produces a probability density function that is more narrow. Graphical representations of how the location parameter and scale parameter affect the shape and symmetry of the normal distribution are given on the next page.

Mathematica Code

```
Plot[Table[PDF[NormalDistribution[[u, 1.5]], x], {u, -2, 0, 2}}]//Evaluate, {x, -6, 6},
Filling -> Axis, PlotLegends-> "μ = -2","μ = 0","μ = 2"},
Plotstyle-> {"Dashed","BGrey","Dotted"}]]
```

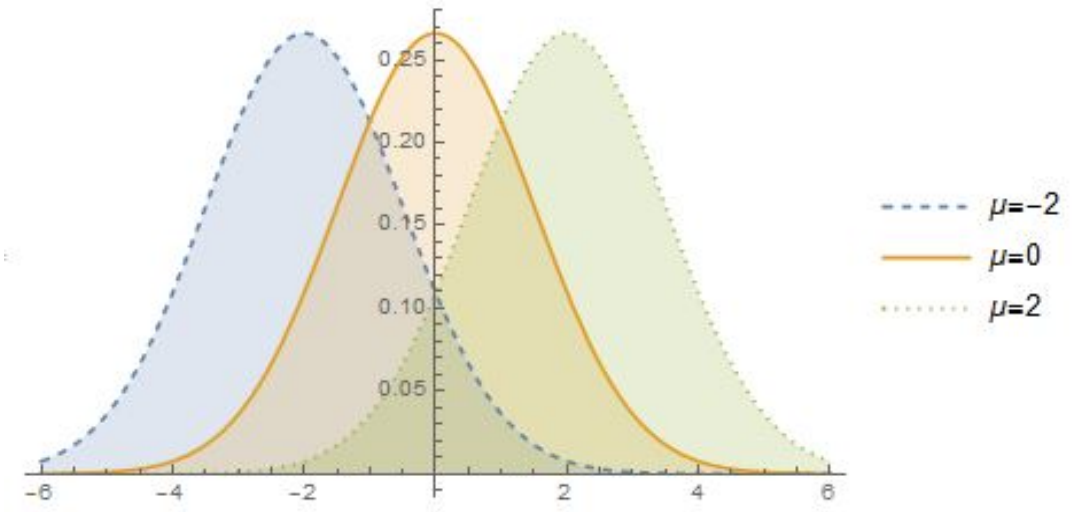


Figure 1: Graphical representation of how the location parameter affects the normal distribution

Mathematica Code

```
Plot[Table[PDF[NormalDistribution[0, σ], {σ, {.75, 1, 2}}]//Evaluate, {x, -6, 6},
Filling -> Axis, PlotLegends-> "σ = 0.75","σ = 1","σ = 2"
Plotstyle-> {"Dashed","BGrey","Dotted"}]]
```

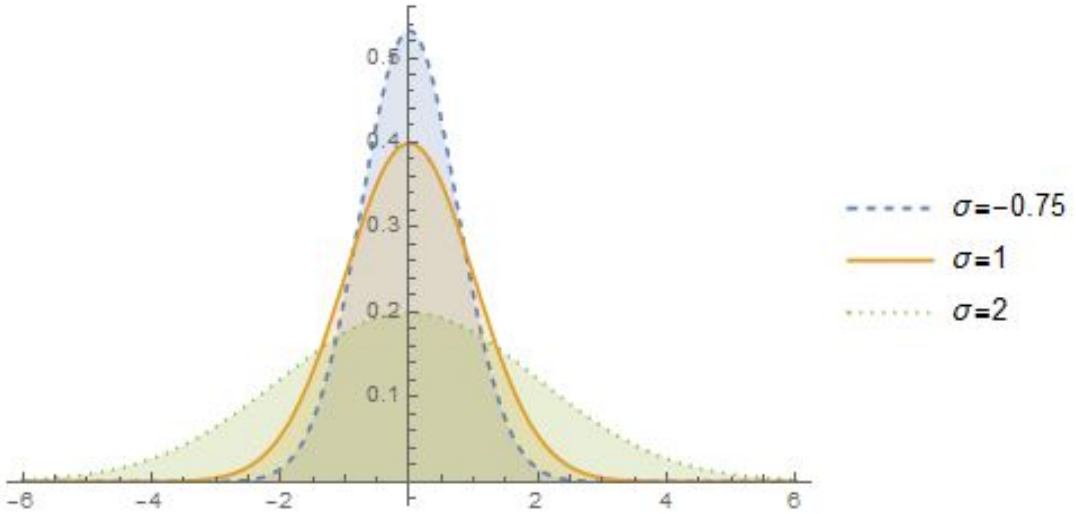


Figure 2: Graphical representation of how the scale parameter affects the normal distribution

A normal distribution has a Pearson coefficient of kurtosis of 3 and a Pearson coefficient of skewness of 0. Excess kurtosis of a distribution is defined in terms of the Pearson coefficient of kurtosis and it is given by Equation (1),

$$\text{Excess kurtosis} = 3 - \text{Pearson coeffecient of kurtosis} \tag{1}$$

Thus the normal distribution has an excess kurtosis of 0. The distribution with a Pearson coefficient of kurtosis of 3 is called mesokurtic hence the normal distribution is called a mesokurtic distribution. A platykurtic distribution is a distribution with an excess kurtosis < 0. In comparison to a normal distribution, a platykurtic distribution will generally have shorter and thinner tails [4]. On the contrary, a distribution with excess kurtosis > 0 is called leptokurtic. In comparison to a normal distribution, a platykurtic distribution will generally have longer and fatter tails [4].

Considered the most important statistical distribution because of it's extensive usage, the normal distribution is computationally easy to use. Standard normal tables exist to aid in the computation of probabilities. Standard normal tables tabulate different ϕ values where ϕ are cumulative distribution function values of a standard normal distribution. Since there exists infinitely many normal distributions, any probability calculation requires a normal distribution (a normal distribution that is not standard) to be transformed to the standard normal distribution first so that the probability calculation can be computed. The normal distribution is widely used in statistical inference and social, physical and biological measurement situations because normality arises naturally in those situations. As a result, different statistical tests called goodness-of-fit tests for normality have been developed over the years to be able to test for the normality assumption.

2.3 The empirical distribution function

The empirical distribution function (EDF) is a pivotal component that is utilised in many goodness-of-fit techniques. Thas (2010), [21], describes the EDF (denoted as$\hat{F}_n(x)$) as an estimator of a distribution function F of a random variable, say X. Since the EDF estimates a cumulative distribution, it also exhibits the basic properties that characterize distribution functions. The fundamental properties are that the EDF is a right-continuous and a non-decreasing function.

Furthermore

$$lim_{x\to-\infty}\hat{F}_n(x) = 0$$

and

$$lim_{x\to\infty}\hat{F}_n(x) = 1$$

are properties that are exhibited by the EDF. The EDF is a function that is directly constructed from the

probability interpretation of F. Hence, for all x and for a random sample of size n, the EDF is defined as

$$\hat{F}_n(x) = \frac{\text{number of observations} \leq x}{n} = \frac{1}{n}\sum_{i=1}^{n} I(X_i \leq x) \tag{2}$$

From the construction of the empirical function, it is imperative to see that it is a non-decreasing step function where each step is a multiple of $\frac{1}{n}$. To indicate this property of the EDF, a graphical illustration of the EDF of a random sample of size 5 with observations 1, 2 , 3, 4 and 8 is depicted in Figure 3.

Mathematica code:

```
data = 1, 2, 3, 4, 8;
D = EmpiricalDistribution[data];
DiscretePlot[CDF[D, x], x, 0, 9, .01, AxesLabel− > "x", "F_n(x)"]
```

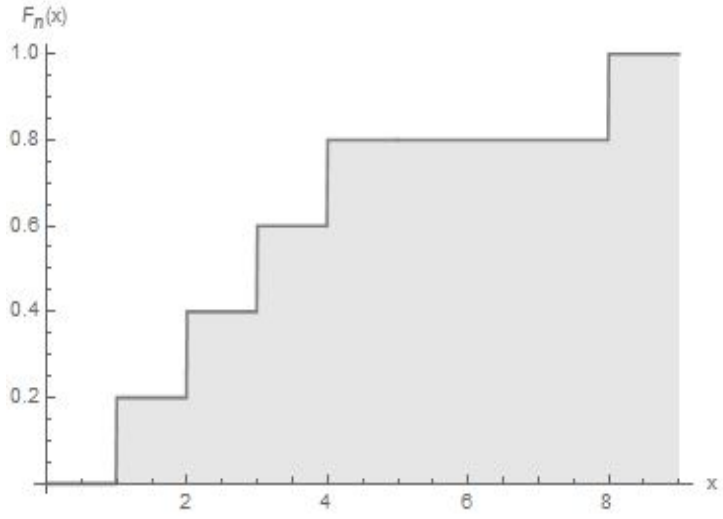


Figure 3: An EDF graphical representation of a random sample

An alternative definition of the empirical distribution function considers the use of order statistics. Assuming a random sample of size n is given and all the observations of the random sample are different, the n observations can be ordered such that $X_1 < X_2 < ... < X_n$. An alternative definition of the EDF using the order statistics is given by:

$\hat{F}_n(x) = 0$ for $x \leq X_1$

$\hat{F}_n(x) = \frac{i}{n}$ for $X_i \leq x \leq X_{i+1}, i = 1, ..., n-1$

$\hat{F}_n(x) = 1$ for $X_n \leq x$

Comparison between the EDF and the binomial distribution yields results that suggest that the EDF and the binomial distribution have a relation. Thas (2010), [21], shows that using Equation (2), $n\hat{F}_n(x)$ has a binomial distribution with parameters n and $F(x)$ and as a result, the following 3 properties hold

for the EDF.

Property 1

$\hat{F}_n(x)$ is an unbiased estimator of $F(x)$. That is, $E\hat{[}F_n(x)] = F(x)$ for all x and for all n.

Property 2

Using the central limit theorem, $\sqrt{n}[\hat{F}_n(x) - F(x)]$ converges pointwise to a normal distribution with mean 0 and variance $F(x)[1 - F(x)]$ for all x.

Property 3

$\hat{F}_n(x) \overset{a.s.}{\to} F(x)$ for every x. This can be rewritten as $\sup_x \left|\hat{F}_n(x) - F(x)\right| \overset{a.s.}{\to} 0$ and the result is formally known as the *Glivenko-Cantelli theorem* [21]. Thas (2010), [21], further describes the error of the EDF as being regulated by the Dvoretzky-Kiefer-Wolfowitz inequality, which affirms that for all $\varepsilon > 0$,

$$Pr\left\{\sup_x \left|\hat{F}_n(x) - F(x)\right| > \varepsilon\right\} \leq 2exp(-2n\varepsilon^2)$$

From the aforementioned properties of the EDF, it can be deduced that $\hat{F}_n(x)$ and $F(x)$ are homogeneous for large sample sizes [21]. Consequently, considering a goodness-of-fit test with null hypothesis

$$H_0 : F(x) = G(x)$$

is equivalent to determining if there exists a difference between the EDF and the hypothesised distribution G. EDF test statistics use this idea as they measure the difference between the hypothesised distribution function G and the sample-based EDF. EDF test statistics usually take the form

$$T_n = c(n)d(\hat{F}_n, G) \tag{3}$$

where $d(,)$ indicates a distance and $c(n)$ denotes a scaling factor that utilises the sample size to ensure the asymptotic null distribution of T_n is non-degenerate [21]. $d(\hat{F_n}, G)$ is a consistent estimator of $d(F, G)$ and $d(F, G) = 0 \iff H_0$ is true [21].

2.4 The van Staden-Loots distribution

The van Staden-Loots distribution is a special type of the generalized linear distribution (GLD) with four parameters, namely, a location parameter α, a scale parameter $\beta > 0$ and two shape parameters $0 \leq \delta \leq 1$ and λ. A formal definition of the van Staden-Loots distribution is outlined in Lemma 6.

Lemma 6. *A real-valued random variable X is said to have the van Staden-Loots type of the GLD, denoted $X \sim GLD_{vSL}(\alpha, \beta, \delta, \lambda)$, if it's quantile function is given by*

$$Q(p) = \begin{cases} \alpha + \beta\left((1-\delta)\left(\frac{p^{\lambda}-1}{\lambda}\right) - \delta\left(\frac{(1-p)^{\lambda}-1}{\lambda}\right)\right) & \lambda \neq 0 \\ \alpha + \beta\left((1-\delta)\,log\,[p] - \delta log\,[1-p]\right) & \lambda = 0 \end{cases} \tag{4}$$

where α is the location parameter, $\beta > 0$ is the scale parameter and $0 \leq \delta \leq 1$ and λ are the shape parameters.

Special cases of the van Staden-Loots distribution are incorporated in the distribution and they are obtainable via a special combination of shape parameter values of the van Staden-Loots distribution. Table 2 gives the combination of these shape parameter values and the corresponding special distribution that is obtained.

Table 2: Table with the GLD_{GPD} special distributions

Distribution	Shape parameter Values
Exponential	$\lambda = 0$, $\delta = 1$
Generalized pareto	$-\infty < \lambda < \infty$, $\delta = 1$
Logistic	$\lambda = 0$, $\delta = \frac{1}{2}$
Skew logistic	$\lambda = 0$, $0 \leq \delta \leq 1$
Tukey's λ	$-\infty < \lambda < \infty$, $\delta = \frac{1}{2}$
Uniform	$\lambda = 1$,$0 \leq \delta \leq 1$ and $\lambda = 2$,$\delta = \frac{1}{2}$

The L-moments and L-moment ratios of the GLD_{vSL} have simple expressions hence are conventionally preferred over other moments. The GLD_{vSL} expressions for the L-location and L-scale are given by

$$L_1 = \alpha + \frac{\beta(2\delta - 1)}{\lambda + 1}$$

and

$$L_2 = \frac{\beta}{(\lambda+1)(\lambda+2)}$$

respectively.

The expressions of the first two L-moment ratios, the L-skewness ratio (τ_3) and the$L$-kurtosis ratio (τ_4), of the GLD_{vSL} are given by

$$\tau_3 = \frac{(2\delta - 1)(1-\lambda)}{\lambda + 3}$$

and

$$\tau_4 = \frac{(\lambda-1)(\lambda-2)}{(\lambda+3)(\lambda+4)}$$

respectively. Note that a distribution that has a L-skewness ratio of 0 is symmetric.

The van Staden-Loots distribution is exceptional at approximating the normal distribution. Using the estimated parameter values $\hat{\alpha} = 0, \hat{\beta} = 2.4449, \hat{\delta} = 0.5$ and $\hat{\lambda} = 0.1416$ for the parameters of the quantile function in Lemma 6, the van Staden-Loots distribution yields a density curve that approximates the normal distribution very well. By specifying a specific combination of L-moments and L-moment ratios, the aforementioned estimated parameters can be obtained. The aforementioned estimated parameter values of the parameters α, β, δ and λ , that result in an approximation of the normal distribution, are obtained given that the L-moments and L-moment ratios are specified as follows

- The$L$-location

$$L_1 = 0$$

- The$L$-scale

$$L_2 = 1$$

- The L-skewness ratio

$$\tau_3 = 0$$

- The L-kurtosis ratio

$$\tau_4 = \frac{30}{\Pi} arctan\left(\sqrt{2}\right) - 9$$

This GLD_{vSL} approximation will approximate a normal approximation with mean 0 and variance 1.7725. Figure 4 depicts the density curve of both the normal distribution and it's respective GLD_{vSL} approximation using the aforementioned combination of parameter values. Since the GLD_{vSL} approximation is very good, the density curve of the normal distribution and the GLD_{vSL} approximation plot very close to each other. Hence, to differentiate between the two curves, the normal distribution is plotted using a dotted line while the GLD_{vSL} approximation is plotted in red on Figure 4.

Mathematica code:

$L_1 = 0;$

$L_2 = 1;$

$\tau_3 = 0;$

$\tau_4 = N\left[\frac{30}{\Pi} arctan\left(\sqrt{2}\right) - 9\right]$

$\lambda = If\left(\tau_4 == N\left[\frac{1}{6}\right], 0, \frac{3+7\tau_4-\sqrt{\tau_4+98\tau_4-1}}{2(1-\tau_4)}\right);$

$\delta = If\left(\lambda == 1, 0.5, 0.5\left(1 - \frac{\tau_3(\lambda+3)}{\lambda-1}\right)\right);$

$\beta = L_2(\lambda+1)(\lambda+2);$

$\alpha = L_1 + \frac{\beta(1-2\delta)}{\lambda+1};$

$ParametricPlot\left[If\left[\lambda == 0, \left\{\alpha + \beta\left((1-\delta)\, log\left[p\right] - \delta log\left[1-p\right]\right), \frac{p(1-p)}{\beta(\delta p+(1-\delta)(1-p))}\right\}\right]\right]$

$\alpha + \beta\left((1-\delta)\left(\frac{p^{\lambda}-1}{\lambda}\right) - \delta\left(\frac{(1-p)^{\lambda}-1}{\lambda}\right)\right), \frac{1}{\beta\left((1-\delta)p^{\lambda-1}+\delta(1-p)^{\lambda-1}\right)}, p, 0.0000001, 0.9999999$

$AspectRatio->0.75, PlotRange->MaxRecursion->15, Frame->True,$

$PlotStyle->Red, Thickness[0.005]]$

$Epilog->Black, Dashed, Thickness[0.5], First@Plot[PDF[NormalDistribution[0,1], x], x, -7, 7]];$

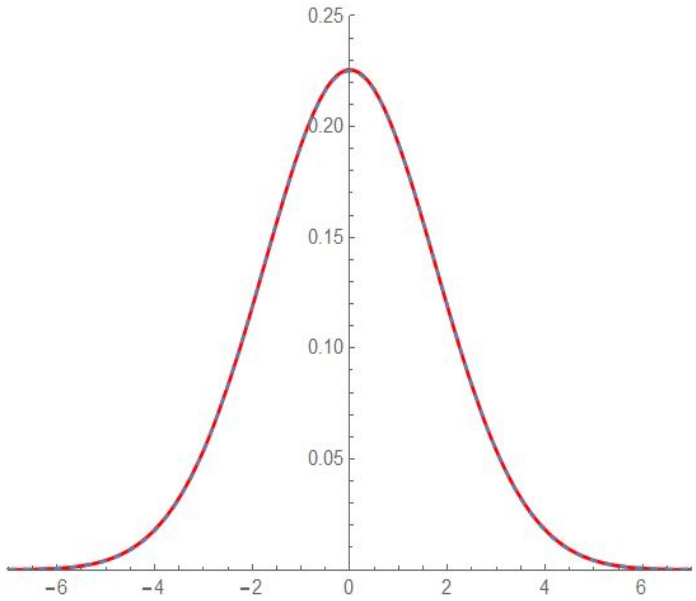


Figure 4: Density curve of a normal distribution and a GLD_{vSL} approximation of the normal distribution

From Figure 4, it is vividly clear that there exists no difference between the approximation of the normal distribution and the normal distribution. A tiny drawback of using the GLD_{vSL} approximation approach to approximate the normal distribution is that the GLD_{vSL} approximation has a bounded support whereas the support of a normal distribution is unbounded. For the GLD_{vSL} approximation, setting $L_1 = 0$ and $L_2 = 1$, results in a support of $[-8.6323, 8, 6223]$.

3 Goodness-of-fit tests

3.1 The Kolmogorov-Smirnov test

The Kolmogorov-Smirnov test is among the oldest non-parametric goodness-of-fit techniques that is used to test for normality. Thas (2010), [21], describes the Kolmogorov-Smirnov test as a test that is based on the empirical distribution function and is consequently referred to as an EDF test. Empirical distribution functions are often utilized to test for a variety of non-normal distributions. The Kolmogorov-Smirnov test, which originates from the work of Kolmogorov [11] and Smirnov[19], belongs to the supremum class of EDF statistics and it is a test that utilizes divergence between the hypothesised distribution function and the EDF to evaluate the quality of the fit [21].

To test the the null hypothesis $H_0 : F = G$ against the alternative hypothesis $H_1 : F \neq G$, the Kolmogorov-Smirnov test statistic is given and calculated by

$$D_n = \sqrt{n} sup_{x \varepsilon S} \left| \hat{F}_n(x) - G(x) \right|$$

which is in the form of Equation (3) with d as the supremum of the function. The formula for D_n indicates that the Kolomogorov-Smirnov test statistic is the largest absolute deviation between the hypothesised distribution G and the EDF. In 1939, Nikolai Smirnov extended the study of the Kolmogorov Smirnov test statistic by considering two statistics that are equivalent to the Kolmogorov Smirnov test statistic. The two test statistics are formally formulated as:

$$D_n^+ = \sqrt{n} sup_{x \varepsilon S} (\hat{F}_n(x) - G(x))$$

and

$$D_n^- = \sqrt{n} sup_{x \varepsilon S} (G(x) - \hat{F}_n(x))$$

(D_n^+) calculates the largest positive deviation between the hypothesised distribution G and the EDF $\hat{F}_n(x)$, while (D_n^-) calculates the largest negative deviation between the hypothesised distribution G and $\hat{F}_n(x)$. (D_n^+) and (D_n^-) are primarily used in directional tests because (D_n^+) is only considered to be large when $F(x) > G(x)$. Hence, it is used to test $H_0 : F = G$ against $H_1 : F > G$. Likewise (D_n^-) is only considered to be small when $F(x) < G(x)$, thus it is used to test $H_0 : F = G$ against $H_1 : F < G$.

A graphical representation of the Kolmogorov-smirnov test statistic and the test statistics studied by Smirnov for a random sample generated from a normal distribution is depicted below in Figure 5.

Mathematica code:

```
Clear["Global`*"];
data = RandomVariate[NormalDistribution[], 50];
D = EmpiricalDistribution[data];
DiscretePlot[CDF[D, x], {x,-3,3,.01},
Epilog->{First@Plot[CDF[NormalDistribution[], x]//Evaluate,{x,-3,3}]}]
```

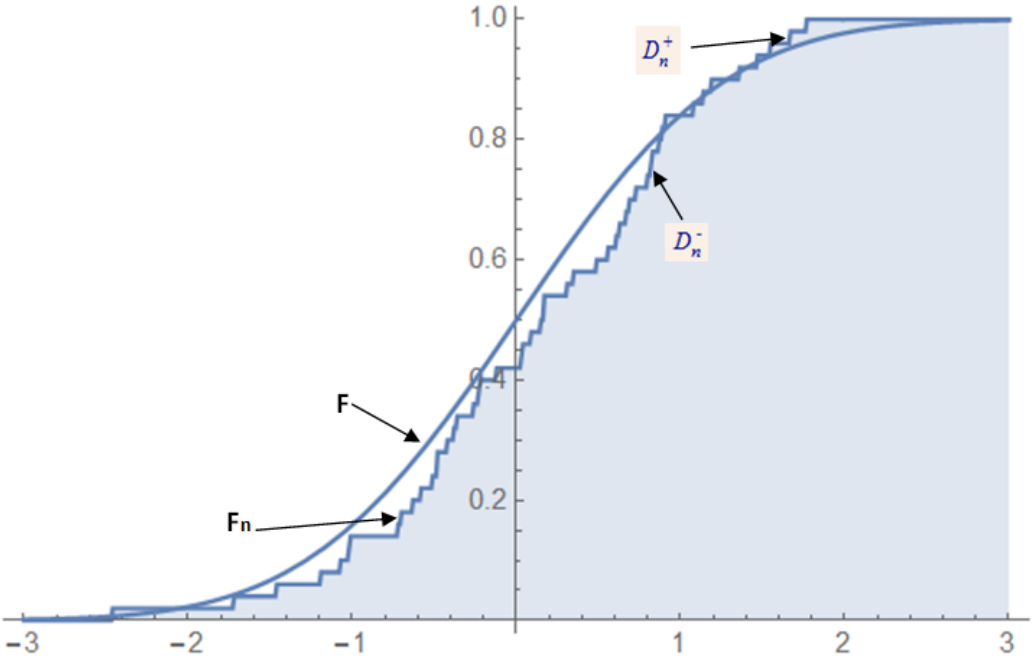


Figure 5: Comparison of empirical and theoretical distributions for a random sample from a normal distribution

The statistics D_n, D_n^- and D_n^+ are distribution free; hence, the null distribution of the different test statistics is the same irrespective of the underlying hypothesised distribution G. Although a closed distribution of the statistics exist, many studies use the asymptotic distribution of the statistics

To test the the null hypothesis $H_0 : F = G$ against the alternative that $H_1 : F \neq G$, $H_1 : F < G$ or $H_1 : F < G$ a p-value is usually calculated using the test statistics D_n, D_n^+ and D_n^- respectively. The null hypotheses will be rejected if the p-value is less than α for the aforementioned tests, where α is the specified significance level.

3.2 The Anderson-Darling Test

In 1952, another important class of EDF statistics called the quadratic class of EDF statistics was invented. Both the Anderson-Darling statistic and the Cramér-von Mises statistic belong to this class of EDF statistics [2]. This class of the EDF statistics is also used for testing distributional assumptions and

the quadratic loss class primarily utilizes the quantity

$$[F_n(x) - G(x)]^2$$

The general form of the quadratic statistics is given by:

$$Q = n \int_{-\infty}^{\infty} w(G(x)) \{\hat{F}_n(x) - G(x)\}^2 dG(x) \tag{5}$$

where w(.) denotes a weight function [23]. The Anderson-Darling statistic named after Theodore Wilbur Anderson and Donald A Darling is given by

$$A^2 = n \int_{-\infty}^{\infty} \frac{\{\hat{F}_n(x) - G(x)\}^2}{G(x)\{1 - G(x)\}} dG(x) \tag{6}$$

which is identical to the general form outlined in Equation (5) with the weight function being equal to $[G(x)\{1 - G(x)\}]^{-1}$ [21]. The hypotheses for the Anderson-Darling test are:

H_0: The data comes from a specified distribution.

H_1: The data does not come from the specified distribution.

The simplest form of an Anderson-Darling test does not require the estimation of distribution parameters although cases do exist when distribution parameters are required to be estimated. Considering an Anderson-Darling test for normality, four different cases can exist based on the given normal distribution assumptions.

Table 3: Table with the four cases for the Anderson-Darling test

Case	u	σ^2
1	Known	Known
2	Known	Unknown
3	Unknown	Known
4	Unknown	Unknown

Cases 2 , 3 and 4 require estimation of parameters for the hypothesised normal distribution. Classical statistical techniques used for the estimation process typically order the n observations of the sample data as $x_1 < x_2 < ... < x_n$ and estimate the unknown parameters using the following formulas:

$$\hat{u} = \begin{cases} u & \text{if mean known} \\ \bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i & \text{otherwise} \end{cases}$$

$$\hat{\sigma}^2 = \begin{cases} \sigma^2 & \text{if variance is known} \\ \frac{1}{n}\sum_{i=1}^{n}(x_i - u)^2 & \text{if variance is unknown but the mean is known} \\ \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 & \text{otherwise} \end{cases}$$

The specified distribution is used to calculate critical values and the test statistic for the Anderson-Darling test. Tables of critical values are available for many distributions such as the normal, uniform, lognormal, exponential and many more distributions. These tables have been discussed comprehensively in the literature by Stephens [20]. Stephens (1974), [20], used a simulation study to compute the table of critical values below:

Table 4: Stephen's Table of critical values used for the Anderson-Darling test

Case	n	15%	10%	5%	2.5%	1%
1	$\geqslant 5$	1.610	1.933	2.492	3.070	3.5857
2			0.908	1.106	1.304	1.573
3	$\geqslant 5$		1.760	2.323	2.904	3.690
4	10	0.514	0.578	0.683	0.779	0.926
	20	0.528	0.591	0.704	0.815	0.969
	50	0.546	0.616	0.735	0.861	1.021
	100	0.559	0.631	0.754	0.884	1.047
	∞	0.576	0.656	0.787	0.918	1.092

The calculation of the test statistic can be quite cumbersome due to the presence of the integral. However, by using the properties of order statistics and a suitable transformation, the calculation of the Anderson-Darling test statistic becomes easier. Given sample data of size n (i.e. $x_1, x_2, ..., x_n$), each observation is standardized by

$$y_i = \frac{x_i - u}{\sigma}$$

and then ordered such that $y_1 < y_2 < ... < y_n$. Then, using the standard normal cumulative distribution function, Φ, A^2 is calculated using

$$A^2 = -n - \frac{1}{n}\sum_{i=1}^{n}(2i-1)\left[ln(\Phi(y_i)) + ln(1 - \Phi(y_{n+i-1}))\right] \tag{7}$$

where n is the sample size and i represents the i^{th}sample of the order statistics. Equation (7) is a closed formula that can be utilized to compute the test statistic when no estimation of parameters of the underlying distribution is required. In contrast, Equation (6) cannot be used to compute the test statistic if estimation of parameters of the underling distribution is required. Through the simulation study, Stephens concluded that the estimation of parameters for the underlying hypothesized distribution from the given sample improves the test even if they are known [20]. He further computed a modified

Anderson-Darling statistic typically used for case 4. The formula for the modified statistic is given by :

$$A^{*2} = \begin{cases} A^2(1 + \frac{0.75}{n} - \frac{2.25}{n^2}) & \text{if u and } \sigma^2 \text{ is unknown} \\ A^2 & \text{otherwise} \end{cases}$$

If A^2 or A^{*2} exceed the corresponding critical values, the null hypothesis is rejected at the corresponding significance level. The critical values in Table 1 are used for the A^2 statistic while A^{*2} is rejected whenA^{*2} is greater than 0.631, 0.752 and 1.035 for the 10% , 5%, and 1% level of significance respectively [20]. The inference used in the latter statistic, A^{*2}, only holds for a sample size of $n \geq 8$ based on simulations studies conducted by Lewis [12]. Lewis tabulated the results of the simulation conducted at the IBM Research Center and was able to conclude that the aforementioned critical values are good approximations to use for inference when working with the Anderson-Darling test. See tables by Lewis (1961), [12], for more details.

The Anderson-Daring is considered to have excellent theoretical properties, however the test performs poorly when applied to some data sets. The data may contain ties (repeating sample data observations) and this adversely affects the Anderson-Darling test due to poor precision[13]. Hence, as the number of ties increase in sample data, the Anderson-Darling test will yield incorrect results that indicate that the data sample is not normal regardless of how good the sample data mimics normally distributed data. When the Anderson-Darling test is used for normality testing, results have shown that it is a good statistical tool for detecting most departures from normality [20]. Using a weight function that equals one in Equation (6) reduces the the Anderson-Darling statistic to a statistic that is commonly referred to as the Cramér-von Mises statistic that is discussed in the next section.

3.3 The Cramér-von Mises test

The Cramér-von Mises statistic also belongs to the quadratic class of EDF statistics and as a result the statistic is also in the form of

$$Q = n \int_{-\infty}^{\infty} w(G(x))\{\hat{F}_n(x) - G(x)\}^2 dG(x)$$

Given that the weight function

$$w(G(x)) = 1$$

the Anderson-Darling statistic described in the previous simplifies to the statistic called the Cramér-von Mises statistic. The Cramér-von Mises statistic is defined as:

$$W^2 = n \int_{-\infty}^{\infty} \{\hat{F}_n(x) - G(x)\}^2 dG(x) \tag{8}$$

which originated from the works of [23] and [5]. Since the Cramér-von Mises statistic and the Anderson-Darling statistic have different weight functions, the statistics exhibit different properties. For instance, unlike the Cramér-von Mises statistic, the Anderson-Darling statistic places more emphasis on sample data points in the tails of the distribution [21].

The calculation for the Cramér-von Mises statistic, as with the Anderson-Darling statistic, has been simplified to make computations more elementary. Cramér (1928), [5], and von Mises (1931), [22], developed the equation that utilizes order statistics. Given sample data of size$n$ (i.e. $x_1, x_2, ..., x_n$) ordered in ascending order such that $x_1 < x_2 < ... < x_n$, an alternative formula for the Cramér-von Mises statistic is given by:

$$W^2 = \sum_{i=1}^{n} \left(G(x_i) - \frac{2i-1}{n} \right)^2 + \frac{1}{12n} \tag{9}$$

where n is the sample size, i represents the i^{th} sample of the order statistics and $G(x_i)$ is the hypothesised distribution value of the i^{th} sample of the order statistics. The formula given by Equation (9) is known as the Cramér-von Mises statistic for one sample. The hypotheses for the test are

$$H_0: \text{ The sample data comes from a normal population}$$

and

$$H_1: \text{ The sample data does not come from a normal population}$$

The critical values for the Cramér-von Mises test are tabulated in the same tables that tabulate the critical values for the Anderson-Darling test [2]. Stephens (1974), [20], used the same simulation study to compute the critical values with the one distinction being the weight function for the Cramér-von Mises statistic. If the value computed using Equation (9) is larger than the tabulated critical value, then the null hypothesis that the data comes from the distribution is rejected.

3.4 The Shapiro-Wilk test

Analysis of variance (ANOVA) is a parametric statistical tool typically used to test to which extent two or more groups differ in an experiment. Like many other parametric methods, ANOVA requires extensive testing of the underlying assumptions such as independence and normality [3]. Shapiro and Wilk (1965),[15], describes a test that was proposed in 1965 called the Shapiro-Wilk test which is primarily

used for testing for normality in the analysis of variance. The Shapiro-Wilk test is a hypothesis test that tests whether a sample comes from a normal population or not. The Shapiro-Wilk statistic is defined as:

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

where

- $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean
- $x_{(i)}$ is the i^{th} order statistic
- a_i are constants defined as follows:

$$(a_1, a_2, ..., a_n) = \frac{m^T V^{-1}}{(m^T V^{-1} V^{-1} m)^{1/2}}$$

where

- $m = (m_1, m_2, ..., m_n)^T$.
- $m_1, m_2, ..., m_n$ are the expected values of the $x'_{(i)}s$.
- V is the covariance matrix of the $x'_{(i)}s$[15].

The Shapiro-Wilk test is computationally demanding and therefore is typically utilized when the sample size is relatively small. For instance, in SAS the Shapiro-Wilk test is only performed in the UNIVARIATE procedure for $n \leq 2000$. Note however that Shapiro and Wilk (1965), [15], showed that the test is likely to yield statistically significant results if a larger sample size is used.

3.5 The Jarque-Bera test

The Jarque-Bera test examines the skewness and kurtosis of sample data and compares the two properties to the corresponding properties of the normal distribution to determine whether the sample data is normal or not [8]. As mentioned in Section 2.2, the skewness of a distribution indicates to what extent the observations of the data set are symmetric about the mean and the kurtosis of the distribution measures the thickness in the tails or the flatness/steepness of the probability density function. The kurtosis and skewness of the distribution are related to the moments of the distribution thus moments are an integral component of the Jarque-Bera test [8].

Jarque & Bera (1978), [8], considered using a Lagrange multiplier method to test the assumption of normality for sample data. The test statistic proposed for the test is:

$$LM = N[(\sqrt{b_1})^2/6 + (b_2 - 3)^2/24] \tag{10}$$

where

$$\sqrt{b_1} = \hat{u}_3/\hat{u}_2^{3/2}$$

and

$$b_2 = \hat{u}_4/\hat{u}_2^2$$

Note that b_1 in equation Equation (10) is the Pearson coefficient of skewness while b_2 represents the Pearson coefficient of kurtosis. $\hat{u}_2$, $\hat{u}_3$ and $\hat{u}_4$ are estimates of the 2^{nd}, 3^{rd} and 4^{th} central moments respectively. Note from Lemma 2 the second central moment is σ^2.

The Jarque-Bera test uses the test statistic in Equation (10) to test the hypotheses:

$$H_0 : \text{The observations of the sample data are normal}$$

$$H_1 : \text{The observations of the sample data are not normal}$$

Since excess kurtosis was defined as

$$\text{excess kurtosis} = 3 - \text{Pearson coeffecient of kurtosis}$$

from Section 2.2, an alternative formulation of the hypotheses of the Jarque-Bera test are given as:

$$H_0 : \text{The skewness and excess kurtosis of the sample data are both 0}$$

and

$$H_1 : \text{Either the skewness or the excess kurtosis is not 0}$$

Under H_0, the test statistic in Equation (10) is asymptotically $\chi^2_{(2)}$ distributed [8]. Hence, the chi-square tables are used for inference for the Jarque-Bera test.

4 Simulation Study

4.1 The distributions utilised for the simulations

The simulation study will focus on 8 different distributions that are obtained by carefully selecting a combination of parameters for the van Staden-Loots distribution. The parameter values are calculated by specifying the L-location L_1, L-scale L_2 , L-skewness ratio τ_3 and the L-kurtosis ratio τ_4. L_1 and L_2 will always be set to 0 and 1 respectively for all the distributions while τ_3 and τ_4 will vary between the distributions. The 8 distributions being considered are given in the Table 5.

Table 5: The 8 distributions with their respective L-moment ratios

Distribution	L-moment ratios
1. Uniform, symmetric	$\tau_3 = 0$ and $\tau_4 = 0$
2. Platykurtic, symmetric	$\tau_3 = 0$ and $\tau_4 = \frac{1}{12}$
3. Platykurtic, asymmetric	$\tau_3 = \frac{1}{6}$ and $\tau_4 = \frac{1}{12}$
4. Normal, mesokurtic, symmetric	$\tau_3 = 0$ and $\tau_4 = \frac{30}{\Pi} arctan\left(\sqrt{2}\right) - 9$
5. Mesokurtic, asymmetric	$\tau_3 = \frac{1}{6}$ and $\tau_4 = \frac{30}{\Pi} arctan\left(\sqrt{2}\right) - 9$
6. Logistic, leptokurtic, symmetric	$\tau_3 = 0$ and $\tau_4 = \frac{1}{6}$
7. Skew-logistic, leptokurtic, asymmetric	$\tau_3 = \frac{1}{6}$ and $\tau_4 = \frac{1}{6}$
8. Exponential, leptokurtic, asymmetric	$\tau_3 = \frac{1}{3}$ and $\tau_4 = \frac{1}{6}$

A graphical illustration of the density curves of the 8 distributions is given on the next page in Figure 6.

Mathematica code:

Clear["Global`*"];

L_1= 0;

L_2=1;

τ_3=...;

τ_4=...; (**τ_3 and τ_4 values change for each different distribution**)

$\lambda = If\left(\tau_4 == N\left[\frac{1}{6}\right], 0, \frac{3+7\tau_4-\sqrt{\tau_4+98\tau_4-1}}{2(1-\tau_4)}\right)$;

$\delta = If\left(\lambda == 1, 0.5, 0.5\left(1 - \frac{\tau_3(\lambda+3)}{\lambda-1}\right)\right)$;

$\beta = L_2(\lambda+1)(\lambda+2)$;

$\alpha = L_1 + \frac{\beta(1-2\delta)}{\lambda+1}$;

$ParametricPlot\left[If\left[\lambda == 0, \left\{\alpha + \beta\left((1-\delta)\,log\left[p\right] - \delta log\left[1-p\right]\right), \frac{p(1-p)}{\beta(\delta p+(1-\delta)(1-p))}\right\}\right]\right]$

$\alpha + \beta\left((1-\delta)\left(\frac{p^\lambda-1}{\lambda}\right) - \delta\left(\frac{(1-p)^\lambda-1}{\lambda}\right)\right), \frac{1}{\beta\left((1-\delta)p^{\lambda-1}+\delta(1-p)^{\lambda-1}\right)}, p, 0.0000001, 0.9999999$

$AspectRatio->0.75, PlotRange->MaxRecursion->15, Frame->True,$

$PlotStyle->Black, Thickness[0.005]]$;

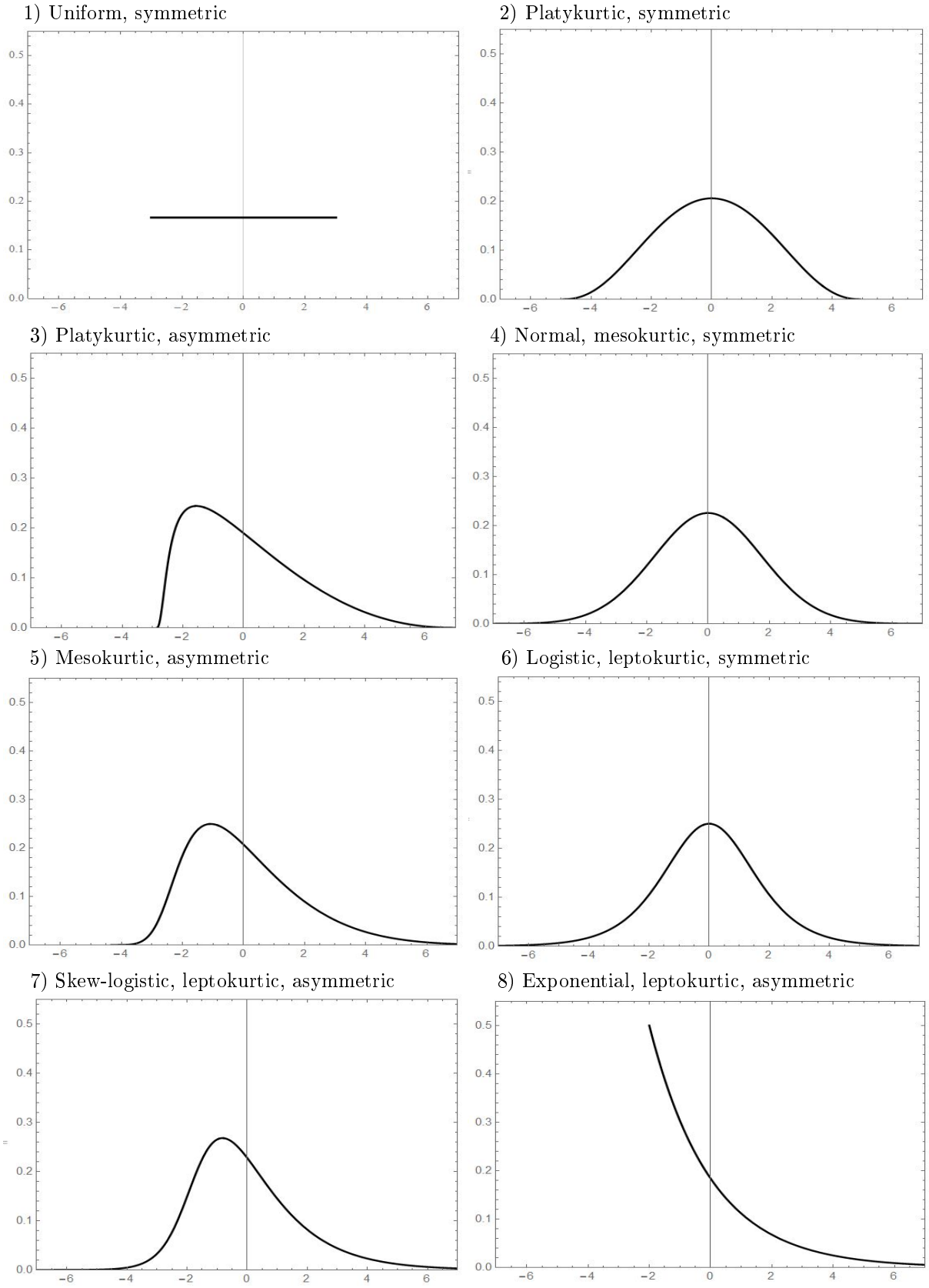


Figure 6: Graphical representations of the 8 distributions used for the simulation

4.2 The empirical $\alpha's$

To calculate the empirical $\alpha's$, the GLD_{vSL} was used to get the normal approximation (Distribution 4). Then 10 000 Monte Carlo simulated samples of different sizes n ($n = 10\,(small)$, $n = 50\,(medium)\,and\,n = 100\,(large)$) were simulated from the normal approximation for the simulation study. Applying the Kolmogorov-Smirnov test, the Cramér-von Mises test, the Anderson-Darling test, the Shapiro-Wilk test and the Jarque-Bera test to the simulated samples, different empirical α values for $\alpha = 0.01$, $\alpha = 0.05$ and $\alpha = 0.1$ were calculated and tabulated. Table 6 tabulates the calculated α values for the different tests for $n = 20$, 50 and 100.

Table 6: Empirical alpha of tests for normality

Test	n	0.01	0.05	0.1
Kolmogorov Smirnov	20	0.0090	0.0450	0.0953
	50	0.0095	0.0483	0.0983
	100	0.0099	0.0517	0.1042
Anderson-Darling	20	0.0086	0.0473	0.0934
	50	0.0079	0.0476	0.0974
	100	0.0093	0.0478	0.0980
Cramér-von Mises	20	0.0083	0.0474	0.0968
	50	0.0084	0.0494	0.0984
	100	0.0100	0.0481	0.1008
Shapiro-Wilk	20	0.0085	0.0441	0.0949
	50	0.0078	0.0449	0.0946
	100	0.007	0.0402	0.0876
Jarque-Bera	20	0.0079	0.0469	0.0992
	50	0.0058	0.0421	0.0925
	100	0.0045	0.0376	0.0845

Since Table 6 tabulates empirical α values using the normal approximation by the GLD_{vSL}, the actual significance level used for hypothesis testing using the tests is not based on theory rather on the observations or the simulated Monte Carlo samples. For instance, a Kolmogorov-Smirnov test applied to a 10 000 samples of size 20, is actually conducted at a significance level of 0.009 instead of the theoretical significance level of 0.01. Also, a Cramér-von Mises test applied to a 10 000 samples of size 50, is actually conducted at a significance level of 0.0084 instead of the theoretical significance level of 0.01. And so on. Hence, the actual significance level will differ from the empirical significance level. The Jarque-Bera test consistently has the bigger difference between the actual significance level and the empirical significance level among the 5 tests. The Kolmogorov Smirnov, Anderson-Darling, Cramér-von Mises and Shapiro-Wilk tests generally give empirical powers similar to the theoretical powers however it can be noted that the Shapiro-Wilk is adversely affected by a larger sample size. This is validated by the empirical powers that are way smaller than theoretical powers when the sample size is $n = 100$.

4.3 Mathematica simulation

In Mathematica, data from the 8 distributions was simulated using the quantile function from the definition of the van Staden-Loots distribution in Section 2.4. Then the goodness-of-fit tests described in Section 3 were applied to the data to find the empirical power of the different goodness-of-fit tests.

Table 7, 8 and 9 gives the summary of the empirical powers of the different tests at the different significance levels. These empirical powers are calculated based on the 10 000 simulated Monte Carlo samples of size $n = 20$, $n = 50$ and $n = 100$ for data that comes from all the 8 distributions detailed in Section 4.1. Table 7 tabulates the empirical powers of the different tests based on the 10 000 simulated Monte Carlo samples at $\alpha = 0.01$. Tables 8 and 9 tabulate the same information at $\alpha = 0.05$ and $\alpha = 0.1$ respectively. Note that the empirical powers for the normal approximation (Distribution 4) will be the same as the empirical powers calculated in Section 4.2. The distribution was included in the tables for comparison purposes.

Table 7: Power of the tests for the different distributions at $\alpha = 0.01$

Distribution	n	KS	AD	CvM	SW	JB
1. Uniform, symmetric	20	0.0237	0.0502	0.0397	0.0308	0.0001
	50	0.0827	0.3073	0.2025	0.3602	0
	100	0.2843	0.8129	0.6193	0.9459	0
2. Platykurtic, symmetric	20	0.0079	0.0080	0.0081	0.0039	0.0011
	50	0.0107	0.0123	0.0109	0.0070	0
	100	0.0162	0.0257	0.0213	0.0166	0
3. Platykurtic, asymmetric	20	0.0620	0.1004	0.0898	0.1062	0.0299
	50	0.2382	0.4639	0.3784	0.5464	0.0589
	100	0.5733	0.8975	0.8011	0.9688	0.2005
4. Normal, mesokurtic, symmetric	20	0.0090	0.0086	0.0083	0.0085	0.0079
	50	0.0095	0.0079	0.0084	0.0078	0.0058
	100	0.0093	0.0093	0.0100	0.0070	0.0045
5. Mesokurtic, asymmetric	20	0.0612	0.1001	0.0880	0.1209	0.0690
	50	0.2116	0.3851	0.3283	0.4674	0.1939
	100	0.5030	0.7849	0.7010	0.8747	0.4793
6. Logistic, leptokurtic, symmetric	20	0.0169	0.0270	0.0244	0.0408	0.0564
	50	0.0303	0.0555	0.0444	0.0986	0.1228
	100	0.0460	0.0968	0.0768	0.1732	0.2170
7. Skew-logistic, leptokurtic, asymmetric	20	0.0805	0.1348	0.1202	0.1658	0.1344
	50	0.2490	0.4105	0.3656	0.4897	0.3631
	100	0.5390	0.7572	0.7039	0.8168	0.6760
8. Exponential, leptokurtic, asymmetric	20	0.3156	0.5593	0.5028	0.6261	0.3148
	50	0.8527	0.9834	0.9607	0.9948	0.7579
	100	0.9987	1	0.9998	1	0.9893

Table 8: Power of the tests for the different distributions at $\alpha = 0.05$

Distribution	n	KS	AD	CvM	SW	JB
1. Uniform, symmetric	20	0.1045	0.191819	0.1620	0.1998	0.0019
	50	0.2928	0.6172	0.4812	0.7520	0.0001
	100	0.6260	0.9598	0.8625	0.9967	0.5570
2. Platykurtic, symmetric	20	0.0468	0.0469	0.0504	0.0400	0.0094
	50	0.0589	0.0698	0.0681	0.0551	0.0011
	100	0.0802	0.1186	0.1070	0.1127	0.0014
3. Platykurtic, asymmetric	20	0.1913	0.2732	0.2435	0.3217	0.1317
	50	0.4910	0.7211	0.6383	0.8233	0.3279
	100	0.8237	0.9797	0.9367	0.9967	0.8474
4. Normal, mesokurtic, symmetric	20	0.0450	0.0473	0.0474	0.0441	0.0469
	50	0.0483	0.0476	0.0494	0.0449	0.0421
	100	0.0517	0.0478	0.0481	0.0402	0.0376
5. Mesokurtic, asymmetric	20	0.1837	0.2513	0.2271	0.2989	0.2050
	50	0.4422	0.6258	0.5641	0.7141	0.4893
	100	0.7508	0.9198	0.8750	0.9624	0.8638
6. Logistic, leptokurtic, symmetric	20	0.0731	0.0970	0.0864	0.1178	0.1511
	50	0.1057	0.1475	0.1332	0.1969	0.2677
	100	0.1500	0.2265	0.1974	0.3033	0.4053
7. Skew-logistic, leptokurtic, asymmetric	20	0.2080	0.2732	0.2518	0.3196	0.2868
	50	0.4644	0.6081	0.5685	0.6658	0.6048
	100	0.7493	0.8835	0.8551	0.9125	0.8808
8. Exponential, leptokurtic, asymmetric	20	0.5670	0.7691	0.7193	0.8307	0.5641
	50	0.9608	0.9965	0.9913	0.9998	0.9583
	100	1	1	1	1	1

Table 9: Power of the tests for the different distributions at $\alpha = 0.10$

Distribution	n	KS	AD	CvM	SW	JB
1. Uniform, symmetric	20	0.2031	0.3178	0.2808	0.3571	0.0101
	50	0.4542	0.7641	0.6395	0.8813	0.2992
	100	0.7886	0.9841	0.9352	0.9997	0.9647
2. Platykurtic, symmetric	20	0.0934	0.0975	0.1005	0.0864	0.0281
	50	0.1192	0.1409	0.1336	0.1253	0.0127
	100	0.1606	0.2106	0.1922	0.2158	0.0563
3. Platykurtic, asymmetric	20	0.3086	0.4029	0.3644	0.4660	0.2462
	50	0.6326	0.8276	0.7567	0.9107	0.6549
	100	0.9082	0.9921	0.9742	0.9996	0.9842
4. Normal, mesokurtic, symmetric	20	0.0953	0.0934	0.0968	0.0949	0.0992
	50	0.0983	0.0974	0.0984	0.0946	0.0925
	100	0.1042	0.0980	0.1008	0.0876	0.0845
5. Mesokurtic, asymmetric	20	0.2894	0.3643	0.3348	0.4154	0.3254
	50	0.5795	0.7366	0.6828	0.8092	0.6931
	100	0.8490	0.9578	0.9289	0.9835	0.9612
6. Logistic, leptokurtic, symmetric	20	0.1383	0.160632	0.1513	0.1841	0.2349
	50	0.1809	0.2284	0.2073	0.2757	0.3689
	100	0.2435	0.3269	0.2952	0.3947	0.5061
7. Skew-logistic, leptokurtic, asymmetric	20	0.3066	0.3745	0.3496	0.4205	0.4031
	50	0.5880	0.7035	0.6727	0.7491	0.7292
	100	0.8389	0.9247	0.9058	0.9409	0.9371
8. Exponential, leptokurtic, asymmetric	20	0.6964	0.8538	0.8118	0.9043	0.7198
	50	0.9841	0.9991	0.9969	1	0.9941
	100	1	1	1	1	1

4.4 SAS simulation

In SAS, the the Kolmogorov-Smirnov test, the Cramér-von Mises test, the Anderson-Darling test, the Shapiro-Wilk test and the Jarque-Bera test were computed using the same methodology used for the simulation in Mathematica as described in Section 4.3. To perform the Shapiro-Wilk test power study, a comparison of the test between the two softwares was investigated. Hence, after using the same methodology to perform the Shapiro-Wilk test in SAS as was done in Mathematica, the data was exported from SAS and imported into Mathematica for comparisons to be made using graphical tools such as p-value graphs. The SAS code used for the simulation is included in the appendix. In Table 10, the power of the Shapiro-Wilk test is tabulated for the different distributions for the simulation conducted in SAS.

Table 10: Power of the Shapiro-Wilk test

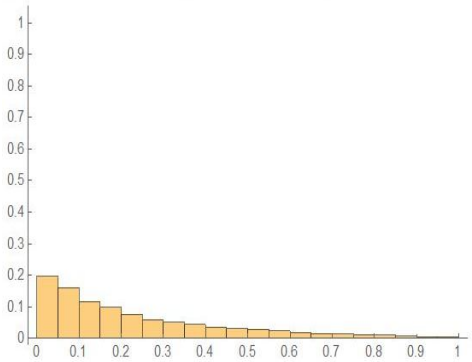
Distribution	n	0.01	0.05	0.10
1. Uniform, symmetric	20	0.0303	0.1966	0.3561
	50	0.0303	0.1966	0.3561
	100	0.0303	0.1966	0.3561
2. Platykurtic, symmetric	20	0.0043	0.0395	0.0909
	50	0.0050	0.0516	0.1214
	100	0.0157	0.1055	0.2149
3. Platykurtic, asymmetric	20	0.0583	0.2117	0.3403
	50	0.5399	0.8243	0.9144
	100	0.9622	0.9956	0.9988
4. Normal, mesokurtic, symmetric	20	0.00831	0.0438	0.09492
	50	0.0083	0.0439	0.0940
	100	0.0097	0.0401	0.0869
5. Mesokurtic, asymmetric	20	0.0867	0.2978	0.4149
	50	0.3256	0.7132	0.8087
	100	0.7007	0.9619	0.9829
6. Logistic, leptokurtic, symmetric	20	0.0476	0.1221	0.1877
	50	0.0953	0.1962	0.2693
	100	0.1673	0.3096	0.3964
7. Skew-logistic, leptokurtic, asymmetric	20	0.1648	0.3174	0.4122
	50	0.4905	0.6682	0.7463
	100	0.8160	0.9109	0.9431
8. Exponential, leptokurtic, asymmetric	20	0.6261	0.8315	0.9033
	50	0.9949	0.9993	1
	100	1	1	1

4.5 SAS and Mathematica comparison of the Shapiro-Wilk test

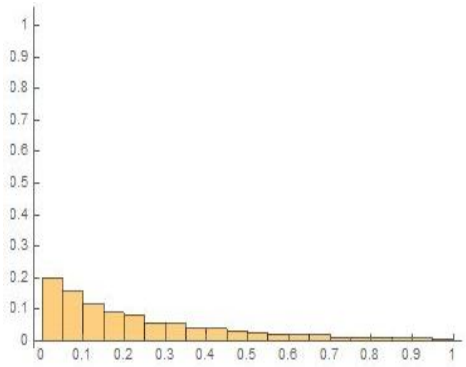
The Shapiro-Wilk test has more power for simulations conducted in Mathematica than simulations conducted in SAS. Although the Mathematica simulation study indicates more power, the difference between the power of both softwares is very small. A graphical representation of the p-value histograms for the uniform symmetric and skew-logistic leptokurtic asymmetric distributions between the two softwares is given in Figure 7. To further validate this, Table 11 tabulates and compares the power (for $n = 100$ and $\alpha = 0.05$) of both softwares for four distributions.

SAS

Uniform symmetric distribution

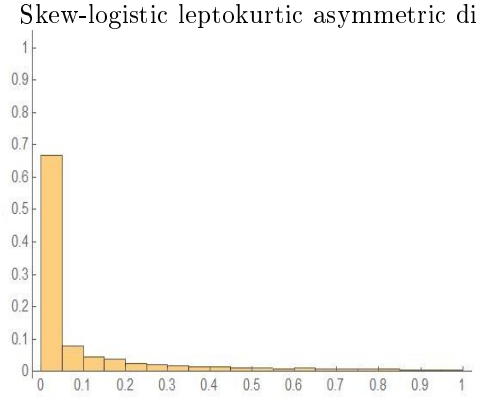


Mathematica



SAS

Skew-logistic leptokurtic asymmetric distribution



Mathematica

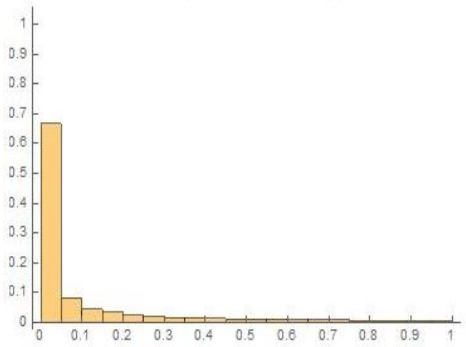


Figure 7: Graphical comparison between the SAS and Mathematica histograms of the Shapiro-Wilk test

Table 11: Shapiro-Wilk test power comparison between SAS and Mathematica

Distribution	SAS	Mathematica
Uniform symmetric	0.1966	0.1998
Platykurtic asymmetric distributions	0.9956	0.9967
Logistic leptokurtic symmetric	0.9109	0.9125
Exponential leptokurtic asymmetric	1	1

Although both softwares produced different empirical powers, the difference is insignificant. This is expected as both softwares use similar processes to conduct the Shapiro-Wilk hypothesis test.

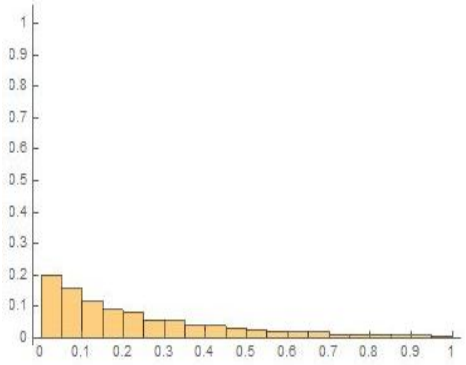
4.6 Power of the Shapiro-Wilk test

The Shapiro-Wilk test has very good power when it is compared to the other four tests. It performs as well as the Anderson-Darling test which is the best EDF test. The Shapiro-Wilk test performs particularly well for skewed alternative distribution. Another noticeable element of the Shapiro-Wilk test is that it is adversely affected by the sample size n. This is indicated by the drastic increase in the power of the test for $n = 100$. Figure 8 graphically illustrates how the power of the Shapiro-Wilk test is affected by the skewness by considering how the test performs for symmetric and asymmetric distributions. Then in

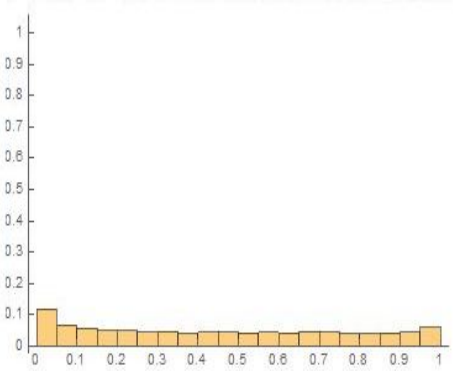
Table 12, the effect of the sample size on the Shapiro-Wilk test is shown by considering 4 distributions at $\alpha = 0.05$.

Symmetric Alternatives

Platykurtic symmetric

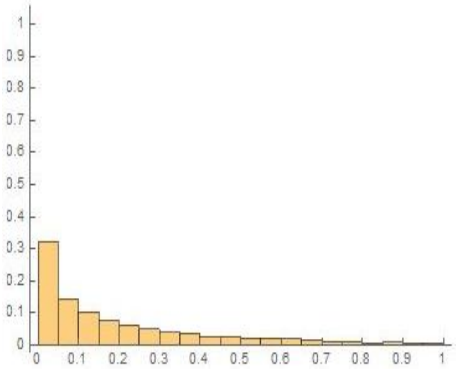


Logistic leptokurtic symmetric



Asymmetric Alternatives

Platykurtic, asymmetric



Skew-logistic, leptokurtic, asymmetric

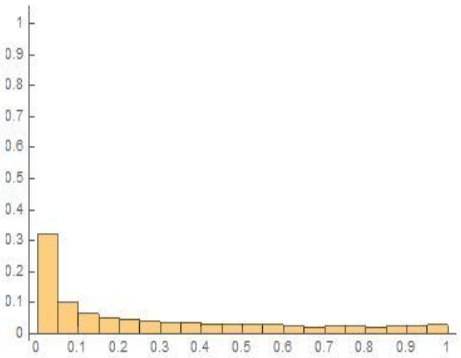


Figure 8: Shapiro-Wilk test power comparison between symmetric and asymmetric alternatives

Table 12: Effect of the sample size on the Shapiro-Wilk test

Distribution	$n = 20$	$n = 50$	$n = 100$
Uniform, symmetric	0.1998	0.752	0.9967
Mesokurtic, asymmetric	0.2989	0.7141	0.9624
Skew-logistic, leptokurtic, asymmetric	0.3196	0.6658	0.9125
Platykurtic, asymmetric	0.3217	0.8233	0.9967

4.7 Power of the Jarque-Bera test

Table 7, 8 and 9 clearly show that the Jarque-Bera test has the lowest power among all 5 tests for symmetric alternatives. Using p-value graphs for the uniform symmetric distribution highlights this property very well in Figure 9. This is attributed to the fact that the Jarque-Bera test statistic is based on the skewness and kurtosis of a distribution. Hence, when applied to a uniform symmetric distribution, only the skewness component of the distribution is similar to the normal distribution whereas the tails on the uniform symmetric distribution differs from the tails the normal distribution.

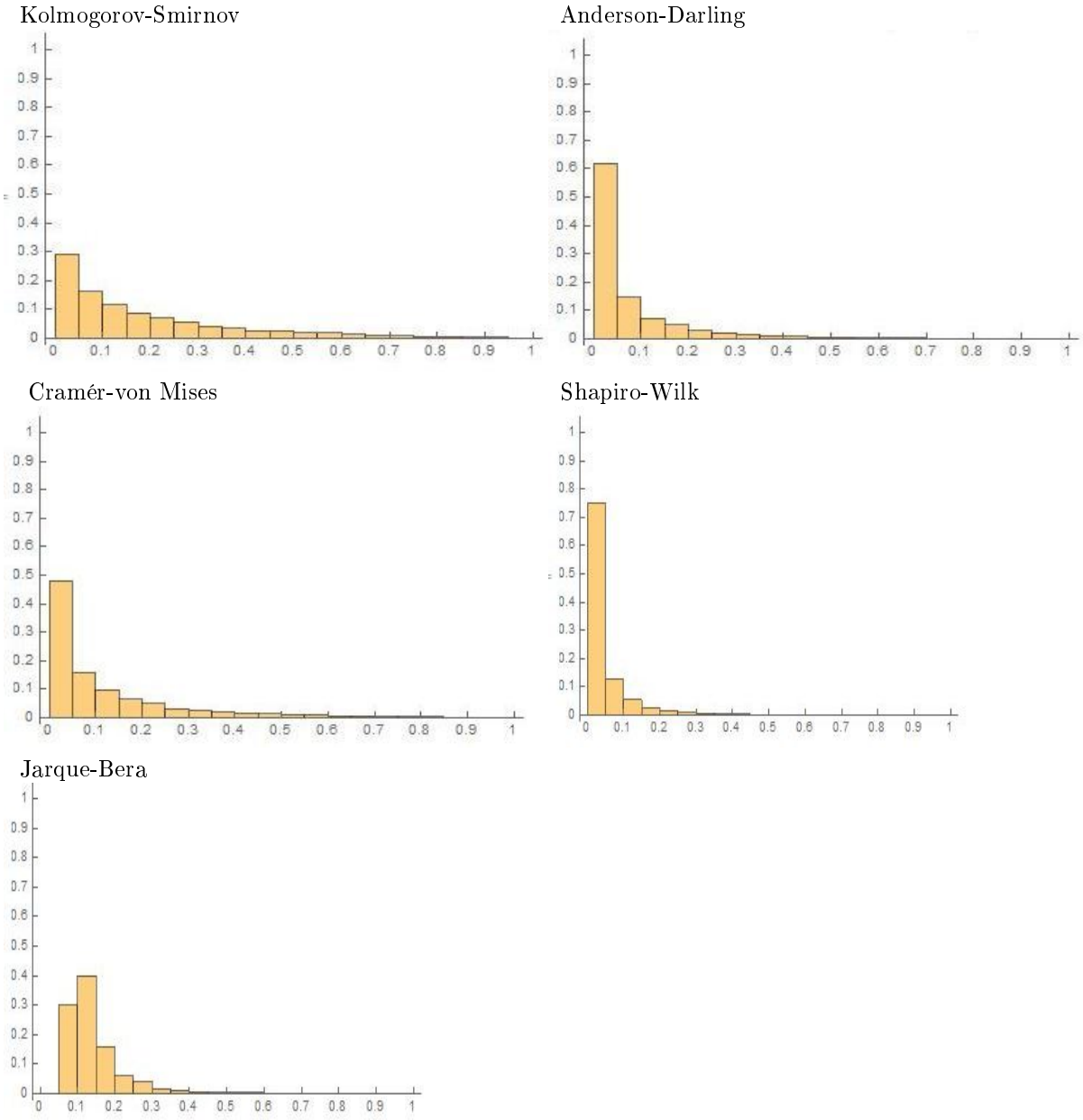


Figure 9: Histograms comparing power of the tests for all the goodness-of-fit tests against the uniform symmetric alternative at $\alpha = 0.05$

Figure 9 illustrates how the Jarque-Bera test performs when the alternative distribution is symmetric, hence when the skewness of the alternative distribution matches that of the normal distribution. The Jarque-Bera test performs better when the alternative distribution is asymmetric leptokurtic as compared to symmetric platykurtic distribution. And in comparison to the normal mesokurtic distribution which is symmetrical, the Jarque-Bera test has more power for asymmetric leptokurtic than normal mesokurtic distribution while the test has more power for the normal mesokurtic distribution than the symmetric platykurtic distribution. In Figure 10, for samples of size $n = 50$ and $\alpha = 0.05$, the Jarque-Bera test was applied to the platykurtic symmetric, normal mesokurtic symmetric and the skew-logistic leptokurtic

asymmetric distributions for comparisons of the p-value graphs.

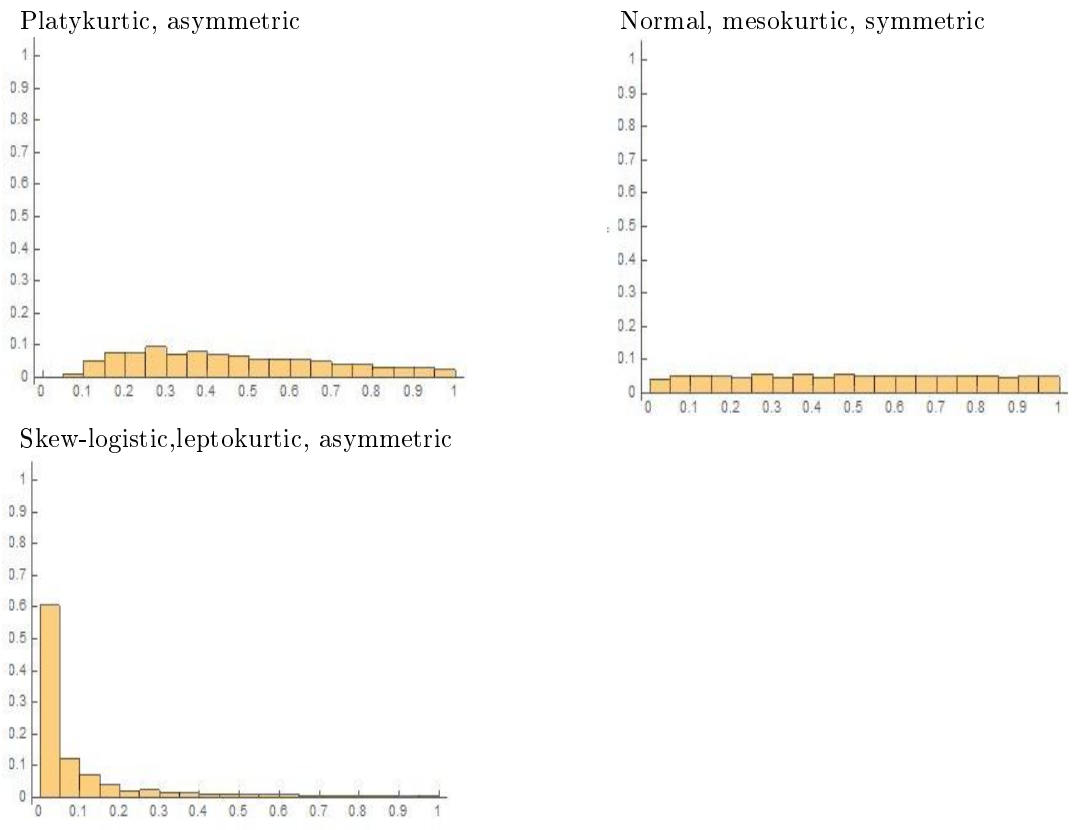


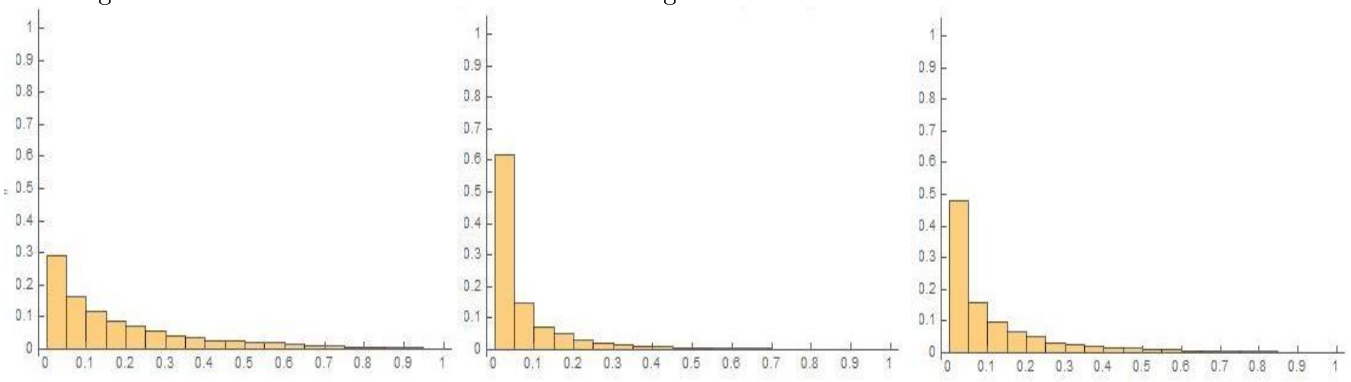
Figure 10: Histograms comparing power of the Jarque-Bera test applied to different distributions at$\alpha = 0.05$

4.8 Power of the EDF tests

For the Kolmogorov-Smirnov, Anderson-Darling and Cramér-von Mises tests, the Anderson-Darling test consistently has the higher power between the three EDF tests. However, comparison between the three tests is a bit flawed because of distribution assumptions required for the hypothesis testing. In Section 3.2, it was shown that 4 different cases for the Anderson-Darling test exist. And using the assumption of normality under the null hypothesis, the report assumes the mean and variance of the underlying null hypothesis is known. This is rarely the case in theory and an estimation method for the normal parameters such as ordinary least squares is typically used to get estimates for the hypothesised normal distribution. Another discrepancy that exists in the comparison of the EDF tests is that the Kolmogorov-Smirnov test statistic requires a sample size of $n \geq 2000$ and the simulation study used sample sizes $n = 20$(small), $n = 50$(moderate) and $n = 100$(large). However, a comparison of the three EDF tests between the four distributions given in Table 8 at a large sample size and $\alpha = 0.05$ indicates that the Anderson-Darling test generally has the better power between the tests.

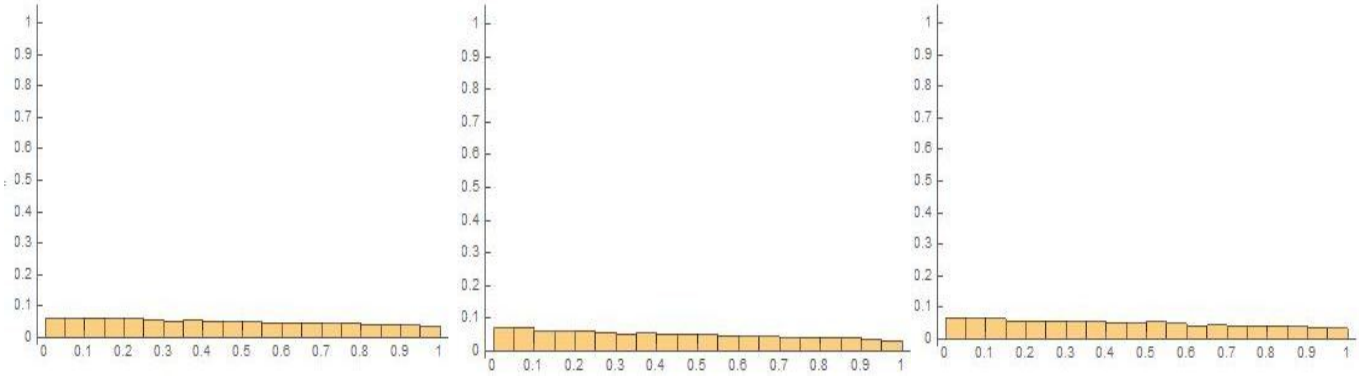
Uniform symmetric distribution

Kolmogorov-Smirnov | Anderson-Darling | Cramér-von Mises



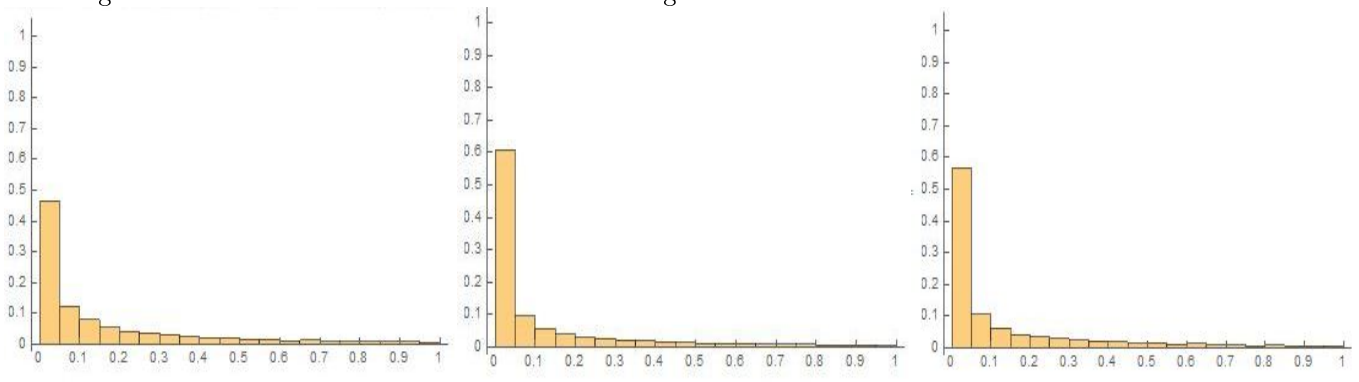
Platykurtic symmetric distribution

Kolmogorov-Smirnov | Anderson-Darling | Cramér-von Mises



Skew-logistic leptokurtic asymmetric distribution

Kolmogorov-Smirnov | Anderson-Darling | Cramér-von Mises



Exponential leptokurtic asymmetric distribution

Kolmogorov-Smirnov | Anderson-Darling | Cramér-von Mises

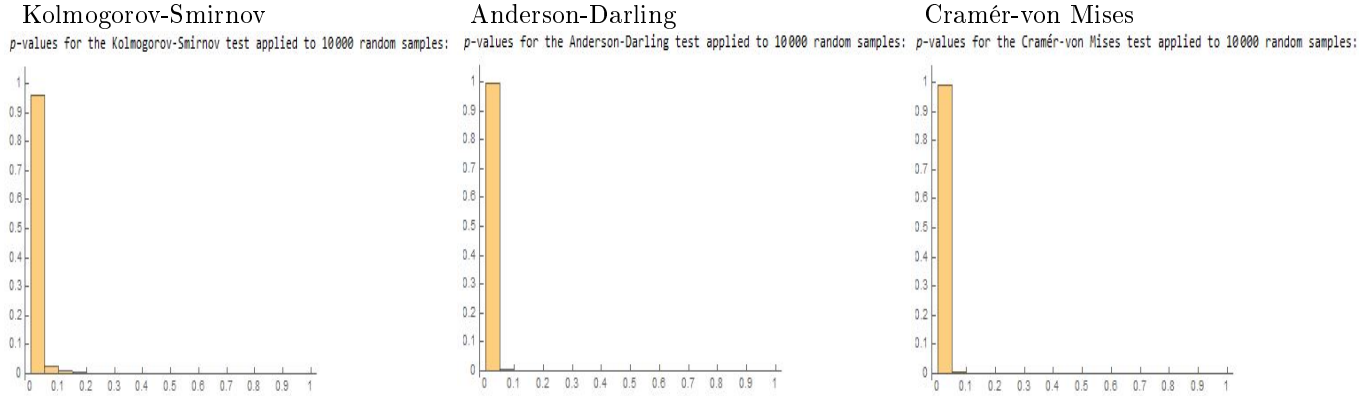


Figure 11: Histograms of the powers for the Kolomogorov-Smirnov, Anderson-Darling and Cramér-von Mises tests at$\alpha = 0.05$

5 Conclusion

The report investigated how the Kolmogorov-Smirnov test, the Cramér-von Mises test, the Anderson-Darling test, the Shapiro-Wilk test and the Jarque-Bera test perform when testing for normality. By using a simulation study, the five tests were compared and contrasted to see how they perform against different alternative distributions. The simulation study mainly consisted of using Monte Carlo samples and applying the different tests to get the power of the tests. The power of these tests was used to critic and analyse the performance of the five different goodness-of-fit tests. The Anderson-Darling test was generally the best performing EDF test while the Jarque-Bera test generally performed the worst. The Shapiro-Wilk test generally performed the best from all the five goodness-of-fit tests. Tables 7, 8 and 9 were used to compare and rank the tests. However, beyond just the power of the tests, other aspects such as the sample size, assumptions of the underlying hypothesised distribution and the complexity of the test statistic calculation should be considered when comparing goodness-of-fit tests.

Shortfalls of the report include software limitations that restricted the comparison of the tests between different statistical software packages. In SAS, the Jarque-Bera test is not incorporated into the system however the test forms part of the hypothesis tests provided in Mathematica. Comprehensive power comparisons on goodness-of-fit tests have appeared in literature. However none of the studies considered symmetric mesokurtic distributions which are non-normal and that could be a recommendation for any future works that are done on goodness-of-fit tests.

References

[1] T. W. Anderson and D. A. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The Annals of Mathematical Statistics*, 23(2):193–212, 1952.

[2] T. W. Anderson and D. A. Darling. A test of goodness of fit. *Journal of the American Statistical Association*, 49(268):765–769, 1954.

[3] L.J. Bain and M. Engelhardt. *Introduction to Probability and Mathematical Statistics.* Classic Series. Brooks/Cole, 2nd edition, 2000.

[4] N. Balakrishnan and V. B Nevzorov. *A Primer on Statistical Distributions.* Wiley., 2003.

[5] H. Cramér. On the composition of elementary errors. *Scandinavian Actuarial Journal*, 11:13–74, 1928.

[6] P Driscoll and F Lecky. An introduction to hypothesis testing. parametric comparison of two groups. *Emergency Medicine Journal*, 18(2):124–130, 2001.

[7] J. R. M. Hosking. L-moments: Analysis and estimation of distributions using linear combinations of order statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 52(1):105–124, 1990.

[8] C.M. Jarque and A. K. Bera. A test for normality of observations and regression residuals. *International Statistical Review/Revue Internationale de Statistique*, 55(2):163–172, 1987.

[9] S. Johnson, N. L. Kotz and N Balakrishnan. *Continuous Univariate Distributions.* Wiley, 2nd edition, 1994.

[10] MC Jones. On some expressions for variance, covariance, skewness and l-moments. *Journal of Statistical Planning and Inference*, 126(1):97–106, 2004.

[11] A.N. Kolmogorov. Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, 4:83–91, 1933.

[12] Peter AW Lewis. Distribution of the anderson-darling statistic. *The Annals of Mathematical Statistics*, pages 1118–1124, 1961.

[13] George Marsaglia and John Marsaglia. Evaluating the anderson-darling distribution. *Journal of Statistical Software*, 9(2):1–5, 2004.

[14] N Unnikrishnan Nair, PG Sankaran, and N Balakrishnan. *Quantile-based reliability analysis.* Springer, 2013.

[15] S.S. Shapiro and B. M. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, 1965.

[16] G. P. Sillito. Derivation of approximants to the inverse distribution function of a continuous univariate population from the order statistics of a sample. *Biometrika*, 56(3):641–650, 1969.

[17] G. P. Sillitto. Interrelations between certain linear systematic statistics of samples from any continuous population. *Biometrika*, 38(3/4):377–382, 1951.

[18] G. P. Sillitto. Some relations between expectations of order statistics in samples of different sizes. *Biometrika*, 51(1/2):259–262, 1964.

[19] N. Smirnov. Sur les escarts de la courbe empirique. *Recueil Mathématique*, 6(1):3–26, 1939.

[20] M.A. Stephens. Edf statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(10):730–737, 1974.

[21] O. Thas. *Comparing Distributions*. Springer, 2010.

[22] R. v. Mises. *Vorlesungen aus dem Gebiete der angewandten Mathematik. 1. Wahrscheinlichkeitsrechnung und ihre Anwendung in der Statistik und Theoretischen Physik*. F. Deuticke, 1931.

[23] R. v. Mises. On the asymptotic distribution of differentiable statistical functions. *The Annals of Mathematical Statistics*, 18(3):309–348, 1947.

[24] Paul J van Staden and MT Theodor Loots. Method of l-moment estimation for the generalized lambda distribution. In *Proceedings of the Third Annual ASEARC Conference*, 2009.

[25] Paul Jacobus Van Staden et al. *Modeling of generalized families of probability distribution in the quantile statistical universe*. PhD thesis, University of Pretoria, 2014.

Appendix

Main Mathematica code

```
Clear["Global`*"];
Print[Style["For all distributions the L − location and the L − scale will be set to 0 and 1", Bold]]
L1 = 0;
L2 = 1;
Print[Style["L − skewness and L − kurtosis ratios with corresponding
distributions to be considered :", Bold]]
Print[Style["Distribution 1 (uniform, symmetric) :", Bold]]
τ3 = 0;
τ4 = 0;
Print["L − location L1 =", L1];
Print["L − scale L2 =", L2];
Print["L − skewness ratio τ3 =", ];
Print["L − kurtosis ratio τ4 =", τ4];
λ = If (τ4 == N [1/6], 0, (3+7τ4−√(τ4+98τ4−1))/(2(1−τ4)));
δ = If (λ == 1, 0.5, 0.5 (1 − (τ3(λ+3))/(λ−1)));
β = L2(λ + 1)(λ + 2);
α = L1 + (β(1−2δ))/(λ+1);
Print["Location parameter =", α];
Print["Scale parameter =", β];
Print["Shape parameter =", δ];
Print["Shape parameter =", λ];
** The code above is replicated for all 8 distributions while changing τ3 and τ4 **i
p = Table[RandomVariate[UniformDistribution[{0, 1}], n], {i, 1, m}]]
Data [If [λ == 0, {α + β ((1 − δ) log [p] − δlog [1 − p]), (p(1−p))/(β(δp+(1−δ)(1−p)))}]]
α + β ((1 − δ) ((p^λ−1)/λ) − δ (((1−p)^λ−1)/λ))
**Kolmogorov-Smirnov Test**
Print["Test statistic values for the Kolmogorov − Smirnov test applied to", m, "random samples :"]
KStestvalue = Table[KolmogorovSmirnovTest[data(i), NormalDistribution[0, 1],
"TestStatistic"], {i, 1, m}];
```

$Histogram[KStestvalue]$

$Print[\text{"}p-values\ for\ the\ Kolmogorov-Smirnov\ test\ applied\ to\text{"}, m, \text{"}randomsamples:\text{"}]$

$KSpvalue = Table[KolmogorovSmirnovTest[data_{(i)}], \{i, 1, m\}];$

$Histogram[KSpvalue]$

Anderson-Darling Test

$Print[\text{"}Test\ statistic\ values\ for\ the\ Anderson-Darling\ test\ applied\ to\text{"}, m, \text{"}random\ samples:\text{"}]$

$ADtestvalue = Table[AndersonDarlingTest[data_{(i)}, NormalDistribution[0, 1],$

$\text{"}TestStatistic\text{"}], \{i, 1, m\}];$

$Histogram[ADtestvalue]$

$Print[\text{"}p-values\ for\ the\ Anderson-Darlingv\ test\ applied\ to\text{"}, m, \text{"}randomsamples:\text{"}]$

$ADpvalue = Table[AndersonDarlingTest[data_{(i)}], \{i, 1, m\}];$

$Histogram[ADpvalue]$

Cramer von-Mises Test

$Print[\text{"}Test\ statistic\ values\ for\ the\ Cramer-von\ Mises\ test\ applied\ to\text{"}, m, \text{"}random\ samples:\text{"}]$

$CvMtestvalue = Table[AndersonDarlingTest[data_{(i)}, NormalDistribution[0, 1],$

$\text{"}TestStatistic\text{"}], \{i, 1, m\}];$

$Histogram[CvMtestvalue]$

$Print[\text{"}p-values\ for\ the\ Cramer-von\ Mises\ test\ applied\ to\text{"}, m, \text{"}randomsamples:\text{"}]$

$CvMpvalue = Table[CramerVonMisesTest[data_{(i)}], \{i, 1, m\}];$

$Histogram[CvMpvalue]$

Shapiro-Wilk Test

$Print[\text{"}Test\ statistic\ values\ for\ the\ Shapiro-Wilk\ test\ applied\ to\text{"}, m, \text{"}random\ samples:\text{"}]$

$SWtestvalue = Table[ShapiroWilkTest[data_{(i)}, \text{"}TestStatistic\text{"}], \{i, 1, m\}];$

$Histogram[SWtestvalue]$

$Print[\text{"}p-values\ for\ the\ Shapiro-Wilk\ test\ applied\ to\text{"}, m, \text{"}randomsamples:\text{"}]$

$SWpvalue = Table[ShapiroWilkTest[data_{(i)}], \{i, 1, m\}];$

$Histogram[SWpvalue]$

Jarque-Bera Test

$Print[\text{"}Test\ statistic\ values\ for\ the\ Jarque-Bera\ test\ applied\ to\text{"}, m, \text{"}random\ samples:\text{"}]$

$JBtestvalue = Table[JarqueBeraTest[data_{(i)}, \text{"}TestStatistic\text{"}], \{i, 1, m\}];$

$Histogram[JBtestvalue]$

$Print[\text{"}p-values\ for\ the\ Jarque-Bera\ test\ applied\ to\text{"}, m, \text{"}randomsamples:\text{"}]$

$JBpvalue = Table[JarqueBeraTest[data_{(i)}], \{i, 1, m\}];$

$Histogram[JBpvalue]$

Main SAS code

```
options ps = 5000 no date page no = 1;
proc iml;
n = 50;
m = 10000;
seed = 13;
L1 = 0;
L2 = 1;
tau3 = 0;
tau4 = 0;
if tau4 = 0 then lamda = 1/6;
if tau4 ^ = 0 then lamda = (3 + 7#tau4 - sqrt(tau##2 + 98#tau4 + 1))/(2#(1 - tau4));
if lambda = 1 then delta = 0.5;
if lambda^=1 then delta = 0.#(1 - tau3#(lambda + 3)/(lambda - 1));
beta = l2#(lambda + 1)#(lambda + 2);
alpha = l1 + beta#(1 - 2#delta)/(lambda + 1);
p = ranuni(j(m, n, seed));
if lambda = 0 then x = alpha + beta(1 - delta)#log(p) - delta#log(1 - p));
if lambda^=0 then x = alpha+beta#((1−delta)#(p##lambda−1)/lambda−delta#((1−p)##lambda−
1)/lambda);
x = shape(x, n#m, 1);
id = j(m, n, .);
do i = 1 to m;
do j = 1 to n;
id[i, j] = i;
end;
end;
id = shape(id, n#m, 1);
xid = x||id;
varlist = 'x','id';
create simulateddata from xid[colname = varlist];
append from xid;
quit;
```

```
proc univariate data = simulateddata normal noprint;
var x;
by id;
output out = shapirowilk normaltest = swtestvalue probn = swpvalue;
run;
proc print data = shapirowilk noobs;
var id swpvalue;
run;
```

Robust Bayesian linear mixed effects regression models in tuberculosis research

Shadrick Simumba 14341230

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor(s): Dr Divan Aristo Burger

Department of Statistics, University of Pretoria

30 October 2017 (Final)

Abstract

Conventional normal linear mixed effects regression models have widely been used for the analysis of clinical trial endpoints without taking into account outliers seen in the data. The objective of this report is to fit a robust Bayesian linear mixed effects regression model for colony forming unit (CFU) count of early bactericidal activity (EBA) tuberculosis (TB) trials. Statistical regression models of log(CFU) count over time usually assume normally distributed random effects and residuals for the analysis of EBA of TB drugs. These regression models therefore do not necessarily accommodate outliers seen in the data. Outliers are occasionally present in CFU count due to erroneous sputum sampling. Such outliers can influence estimates of the rate of change in CFU count.

A Bayesian linear mixed effects regression model was introduced to offer a robust approach to accommodate outliers in the data. The proposed regression model fits the Student t distribution to residuals, and the normal distribution to random coefficients. A Bayesian framework is adopted for estimation of and inferences on model parameters. A Bayesian approach is an alternative to classical methods which is relatively easy to implement.

Declaration

I, Shadrick Simumba, declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____

Shadrick Simumba

_____

Divan Aristo Burger

_____

Date

Acknowledgements

I would like to acknowledge the financial support from the Center for AI Research, Meraka Institute, CSIR in the form of a post graduate bursary. I would also like to thank my supervisor Dr Divan Aristo Burger for pointing me in the right direction for my research. Lastly I would like to thank my family and friends for all the support during the whole research period.

Contents

List of Figures

List of Tables

1 Introduction

Mixed effects regression models are those that include both fixed and random effects, and are flexible in the modeling of longitudinal data and other correlated data. In clinical trials, random effects generally represent patient-specific effects, whereas fixed effects represent the population-level effects of the model. With the study of longitudinal data of clinical trials, study participants are evaluated over a period of time and, for each individual, data are collected at multiple time points. Mixed effects models may be viewed as extensions of classical regression models for multi-level (or hierarchical) data by introducing random effects in the model. The feature that distinguishes mixed models from fixed effects models is that mixed models can model data for which the observations are not independent. In other words, mixed models can model the covariance structure of the data [3]. Treating longitudinal measurements, collected from the same patient, as uncorrelated data is inappropriate when making inferences on clinical endpoints of interest. Mixed effects regression models take into account correlation among measurements made on the same patient, either by incorporating random effects or, (random coefficients) by the specification of relevant covariance patterns. Mixed effects regression models allow one to fit longitudinal data of all patients of a clinical trial in a single model, such that the model parameters can vary between patients (hence, the applicable regression can be tailored for each patient) [18]. These models are also appealing in the sense that random effects estimates are generally shrunken towards their corresponding fixed effects counterparts, and therefore may improve the precision of parameter estimates that are of interest [5]. Furthermore, mixed effects regression models appropriately accommodate missing (when missing at random) and unbalanced data which are regularly encountered in longitudinal applications.

Most linear mixed effects regression models used in practice assume that both the residuals and random effects are normally distributed. However, in the presence of outliers (heavy tails), such models lack robustness to deviations from certain model assumptions which can lead to invalid inference and unreasonable parameter estimates [17]. Robust regression models are appealing alternatives to conventional regression models in the sense that they accommodate outliers in the data, and are less sensitive to other departures from the applicable model assumptions [15]. Outliers are occasionally present in CFU count due to erroneous sputum sampling, and hence, the reasonability to adopt robust methods to accommodate such outliers [6].

The focus of this report is to fit a robust Bayesian linear mixed effects regression model to CFU count of tuberculosis (TB) treatments of bactericidal activity (EBA) trials. The adopted regression model fitted to the data assumes that random effects follow normal distributions, and incorporates Student t distributed residuals (hence, the model is potentially robust to outliers present in the data). The Student t distribution with small degrees of freedom has heavier tails than the corresponding normal distribution. A robust regression model should maintain the validity of inferences made with a minimal effect of outliers

on statistical inference.

A Bayesian framework is adopted for estimation and inferences of model parameters using Gibbs sampling available in OpenBUGS. A Bayesian approach is an alternative to classical methods which is in many cases relatively easy to implement. Bayesian inferences are also advantageous in the sense that they do not depend on asymptotic approximations as classical inference methods do for complex models [1].

2 Background Theory

2.1 Need for robust models in tuberculosis research

In EBA trials, the primary goal is to obtain accurate measurements for CFU count collected from sputum samples. However, outliers are occasionally present in the data due to erroneous sputum sampling or reporting of data. Factors such as dilution of samples, contamination, slow culture growth, temperature, and other conditions attribute to erroneous sputum sampling, which result in the presence of outliers in the data. These outliers can influence estimates of the rate of change in CFU count [6].

In most EBA trials, only a small number of patients are allocated per treatment group [8]. The presence of outliers in such a small number of dataset has an influence on the statistical analysis of the associated findings [9]. Previous research suggested that, implausible data points (such as outliers) should be exclude when fitting a regression model to CFU count [13]. However, the criteria used for identifying and excluding outliers are not only difficult to implement, but also in some cases impossible to carry out. The primary efficacy analysis of TB trials should rather include all observations in the study (instead of excluding them). Therefore, given the aspects listed above, it seems reasonable to fit robust regression methods to accommodate outliers.

2.2 Early bactericidal activity

The EBA of TB drugs is measured as the rate of decrease in CFU count in sputum of patients with microscopy-positive pulmonary TB that are collected during the first days to weeks of treatment [9]. The EBA of TB drugs assesses the relative potency in the early stages of treatment.

The EBA$(t_1 - t_2)$ is defined as the rate of change in log(CFU) count over a given time interval, say Day t_1 and Day t_2 [16], and is expressed as follows:

$$\text{EBA}\,(t_1 - t_2) = -\frac{\hat{f}\,(t_2) - \hat{f}\,(t_1)}{t_2 - t_1} \tag{1}$$

where $f\,(t)$ is the appropriate regression function for the CFU count against time, and $\hat{f}\,(t_1)$ and $\hat{f}\,(t_2)$ are the fitted log(CFU) counts at Day t_1 and Day t_2, respectively [16]. The calculation of the EBA

from Equation (1) is thus model based (that is, not being based on the observed data). The regression functions $f(t_1)$ and $f(t_2)$ can therefore be estimated using all of observed CFU counts available over time. From Equation (1), it can be seen that as the rate of change over a given time interval, (i.e. $\text{EBA}(t_1 - t_2)$) increases, the effectiveness of a given drug against TB bacteria also increases. In other words, the potency of a given drug against TB bacteria is proportional to the EBA of a TB drug.

2.3 Linear regression function

In this section the linear regression model of CFU count over time is described by the rate of decrease in CFU count (that is held constant over time). Assuming that the rate of change (decrease) in the expected CFU count at time t, $\mu(t)$, is proportional to the expected value at time t, then $\mu(t)$, can be expressed as follows:

$$\frac{d\mu(t)}{dt} = -\lambda\mu(t) \tag{2}$$

where $\lambda > 0$ is a constant that describes the rate of change over time.

From Equation (2) it follows that:

$$\frac{d\mu(t)}{\mu(t)} = -\lambda dt \tag{3}$$

Integrating both sides of Equation (3) gives:

$$\int \frac{1}{\mu(t)} d\mu(t) = \int -\lambda dt \tag{4}$$

The solution to Equation (4) is given as:

$$\log(\mu[t]) = C - \lambda t \tag{5}$$

where C is the constant term.

Let α characterise a single intercept parameter (incorporated in the regression function in the Equation (5)), assume the initial condition to be $\log(\mu[t]) = \alpha$, then C can be solved as:

$$C = \alpha \tag{6}$$

Replacing C in Equation (5) with Equation (6) results in the following regression function:

$$\log(\mu[t]) = \alpha - \lambda t \tag{7}$$

Figure 1 shows an example of the plot of the expected CFU count and corresponding time from a linear regression function, i.e. the plot of Equation (7) for a given patient from a conventional 14-day EBA study [4].

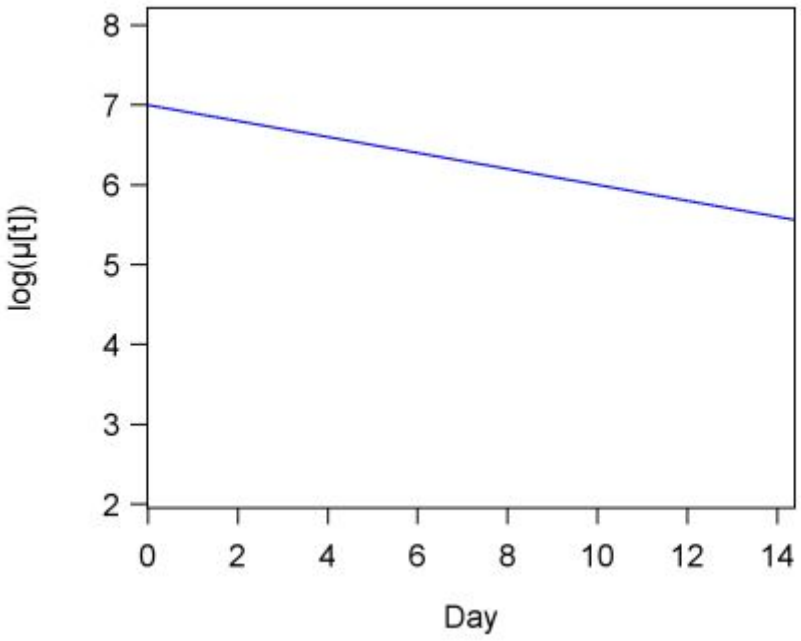


Figure 1: Plot of the expected CFU count (log([t])) over time

2.4 Bayesian linear mixed effects regression model

Longitudinal data, specifically in this instance, CFU count over time can be modelled by a linear mixed effects regression model that best explains how the outcome measurements in CFU count are related to time. The "time" effect is incorporated in the regression model as the covariate, resulting in the so called random coefficients model. The adopted Bayesian linear mixed effects regression model for CFU count versus time is expressed as follows:

$$y_{ijk} = \alpha_{ij} - \lambda_{ij} t_{ijk} + \varepsilon_{ijk} \tag{8}$$

where y_{ijk} is the CFU count for patient $i = 1, ..., N_j$ in treatment group $j = 1, ..., J$ at time point $k = 1, ..., K_{ij}$, and t_{ijk} is time measurement, and ε_{ijk} are the residuals terms. In this case, N_j represents the number of patients assigned to treatment group j, and T_j represents the total number of time points across all patients allocated to treatment group j. Let $\sum_{j=1}^{J} N_j = N$ represents the total number of patients in a given trial.

The model includes the following random coefficients: (α_{ij}) are the intercepts, and (λ_{ij}) are the slopes characterising the rate of change over time. The terms α_{ij} and λ_{ij} are the sums of fixed effects and associated random coefficients, namely:

$$\mu_{ij} = \begin{bmatrix} \alpha_{ij} \\ \lambda_{ij} \end{bmatrix} = \begin{bmatrix} \alpha_j \\ \lambda_j \end{bmatrix} + \begin{bmatrix} \mu_{0ij} \\ \mu_{1ij} \end{bmatrix} \tag{9}$$

$$\Omega_{\boldsymbol{\mu j}} = \begin{bmatrix} \sigma^2_{\alpha j} & \sigma_{\alpha j \lambda j} \\ \sigma_{\alpha j \lambda j} & \sigma^2_{\lambda j} \end{bmatrix} \tag{10}$$

where $\mu_{ij} = (\alpha_{ij}, \lambda_{ij})'$ and $\mu_j = (\alpha_j, \lambda_j)$ (or $[\mu_{0ij}, \mu_{1ij}]'$) are respectively the vectors of random and mean intercepts and slopes. $\Omega_{\boldsymbol{\mu j}}$ are the covariance matrices of the random intercepts and slopes. Here, the fixed effects represent the average effect for each treatment group similarly, the fixed effects represent the average effect of each treatment group. Random effects are specified so that separate regression lines can be fitted for each patient (here, model parameters are allowed to vary between patients). The specification of a random coefficients model generally shrinks the estimates of the random effects (hence, regression estimates per patient) towards the average estimates (or estimates of the fixed effects), thus avoiding outlier estimates of the random effects which might arise from incomplete data [3]. The distributions of ε_{ijk} are assumed to be independent of the distributions of μ_{ij}.

2.5 Model specifications for Bayesian mixed effects regression models

From Equation (8) , the residuals are assumed to follow *i.i.d.* Student t distributions and random coefficients are assumed to be normally distributed as follows:

$$\varepsilon_{ijk} | \sigma^2_{\varepsilon j}, \nu_j \sim T\left(0, \sigma^2_{\varepsilon j}, \nu_j\right) \tag{11}$$

where $\sigma^2_{\varepsilon j}$ are the scale parameters, and ν_j are degrees of freedom of the corresponding Student t distribution. The covariance matrices are assumed to follow Wishart distributions.

The density function of $\varepsilon_{ijk} | \sigma^2_{\varepsilon j}, \nu_j$ can be written as:

$$P\left(\varepsilon_{ijk} | \sigma^2_{\varepsilon j}, \nu_j\right) \propto \frac{1}{\sqrt{v_j} \cdot \sigma_{\varepsilon j}} \frac{\Gamma\left(\frac{\nu_j+1}{2}\right)}{\Gamma\left(\frac{\nu_j}{2}\right)} \times \left(1 + \frac{1}{\nu_j}\left(\frac{\varepsilon_{ijk}}{\sigma_{\varepsilon j}}\right)^2\right)^{-\frac{\nu_j+1}{2}} \tag{12}$$

where $\Gamma(.)$ denotes the gamma function. The scale parameters follow gamma prior distributions, namely $\sigma^{-2}_{\varepsilon j} \sim G\left(10^{-4}, 10^{-4}\right)$, and the degrees of freedom follow uniform prior distributions, namely:

$$\nu_j \sim U\left(2, 100\right) \tag{13}$$

The density function of ν_j is written as:

$$P(\nu_j) \propto I(2 \leq \nu_j \leq 100) \tag{14}$$

where $I(x)$ is an indicator function taking the value 1 if x is true, and 0 otherwise.

The specification of the Student t distribution offers a robust approach which accommodates outliers and heavily tailed residuals (depending on its degrees of freedom ν_j) in the CFU count [13].

Theorem 1. *The Student t distribution can be specified as a mixture of a normal distribution with mean and unknown variance, and an inverse gamma distribution assumed for the unknown variance.*

$$y_{ijk}|\alpha_{ij}, \lambda_{ij}, \sigma^2_{\xi j}, \sigma^2_{\varepsilon j} \sim N\left(\alpha_{ij} - \lambda_{ij} \cdot t_{ijk}, \sigma^2_{\xi j} \cdot \sigma^2_{\varepsilon j}\right). \tag{15}$$

$$\sigma^{-2}_{\xi j}|\nu_j \sim G\left(\frac{\nu_j}{2}, \frac{\nu_j}{2}\right). \tag{16}$$

Proof. Let $y_{ijk}|\alpha_{ij} - \lambda_{ij}, \sigma^2_{\xi j}, \sigma^2_{\varepsilon j} \sim N\left(\alpha_{ij} - \lambda_{ij} \cdot t_{ijk}, \sigma^2_{\xi j} \cdot \sigma^2_{\varepsilon j}\right)$. □

The probability density function of the normal distribution is given by:
$P\left(y_{ijk}|\sigma^2_{\xi j}, \sigma^2_{\varepsilon j}\right) = \frac{1}{\sqrt{2\pi}\sigma_{\xi j}\sigma_{\varepsilon j}} \exp\left[-\frac{1}{2}\frac{(y_{ijk}-(\alpha_{ij}-\lambda_{ij}t_{ijk}))^2}{\sigma^2_{\xi j}\sigma^2_{\varepsilon j}}\right]$
Let $\sigma^{-2}_{\xi j}|\nu_j \sim G\left(\frac{\nu_j}{2}, \frac{\nu_j}{2}\right)$.
The probability density function of the gamma distribution is given by:
$P\left(\sigma^{-2}_{\xi j}|\nu_j\right) = \frac{\left(\frac{\nu_j}{2}\right)^{\frac{\nu_j}{2}}}{\Gamma\left(\frac{\nu_j}{2}\right)}\left(\sigma^{-2}_{\xi j}\right)^{\frac{\nu_j}{2}-1}\exp\left(-\frac{\nu_j}{2\sigma^2_{\xi j}}\right)$
The marginal probability of $y_{ijk}|\nu_j$ is given by:

$$P(y_{ijk}|\alpha_{ij} - \lambda_{ij} \cdot t_{ijk}, \nu_j) = \int_0^\infty P\left(y_{ijk}|\alpha_{ij} - \lambda_{ij} \cdot t_{ijk}, \sigma^2_{\xi j}, \sigma^2_{\varepsilon j}\right) P\left(\sigma^{-2}_{\xi j}|\nu_j\right) d\sigma^{-2}_{\xi j}$$

$$= \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_{\xi j}\sigma_{\varepsilon j}} \exp\left[-\frac{1}{2}\frac{(y_{ijk} - (\alpha_{ij} - \lambda_{ij} \cdot t_{ijk}))^2}{\sigma^2_{\xi j} \cdot \sigma^2_{\varepsilon j}}\right] \frac{\left(\frac{\nu_j}{2}\right)^{\frac{\nu_j}{2}}}{\Gamma\left(\frac{\nu_j}{2}\right)} \left(\sigma^{-2}_{\xi j}\right)^{\frac{\nu_j}{2}-1} \exp\left(-\frac{\nu_j}{2\sigma^2_{\xi j}}\right) d\sigma^{-2}_{\xi j}$$

$$= \frac{1}{\sqrt{2\pi}\sigma_{\varepsilon j}} \frac{\left(\frac{\nu_j}{2}\right)^{\frac{\nu_j}{2}}}{\Gamma\left(\frac{\nu_j}{2}\right)} \int_0^\infty \left(\sigma^{-2}_{\xi j}\right)^{\frac{\nu_j}{2}+\frac{1}{2}-1} \exp\left\{-\left[\frac{(y_{ijk} - (\alpha_{ij} - \lambda_{ij}t_{ijk}))^2}{2\sigma^2_{\varepsilon j}} + \frac{\nu_j}{2}\right]\sigma^{-2}_{\xi j}\right\} d\sigma^{-2}_{\xi j}$$

Since the density function of gamma integrates to 1, it follows that:

$$= \frac{1}{\sqrt{2\pi}\sigma_{\varepsilon j}} \frac{\left(\frac{\nu_j}{2}\right)^{\frac{\nu_j}{2}}}{\Gamma\left(\frac{\nu_j}{2}\right)} \times \Gamma\left(\frac{\nu_j + 1}{2}\right) \times \left(\frac{(y_{ijk} - (\alpha_{ij} - \lambda_{ij}t_{ijk}))^2}{2\sigma^2_{\varepsilon j}} + \frac{\nu_j}{2}\right)^{-\frac{\nu_j}{2}-\frac{1}{2}}$$

$$= \frac{1}{\sqrt{2\pi}\sigma_{\varepsilon j}} \frac{\left(\frac{\nu_j}{2}\right)^{\frac{\nu_j}{2}}}{\Gamma\left(\frac{\nu_j}{2}\right)} \times \Gamma\left(\frac{\nu_j+1}{2}\right) \times \left(\frac{\nu_j}{2}\right)^{-\frac{\nu_j}{2}-\frac{1}{2}} \times \left(1 + \frac{\left(y_{ijk} - (\alpha_{ij} - \lambda_{ij} t_{ijk})\right)^2}{\nu_j \sigma^2_{\varepsilon j}}\right)^{-\frac{\nu_j}{2}-\frac{1}{2}}$$

$$= \frac{1}{\sqrt{\pi v_j}\sigma_{\varepsilon j}} \frac{\Gamma\left(\frac{\nu_j+1}{2}\right)}{\Gamma\left(\frac{\nu_j}{2}\right)} \times \left(1 + \frac{\left(y_{ijk} - (\alpha_{ij} - \lambda_{ij} t_{ijk})\right)^2}{\nu_j \sigma^2_{\varepsilon j}}\right)^{-\frac{\nu_j}{2}-\frac{1}{2}}$$

$$= \frac{1}{\sqrt{\pi v_j}\sigma_{\varepsilon j}} \frac{\Gamma\left(\frac{\nu_j+1}{2}\right)}{\Gamma\left(\frac{\nu_j}{2}\right)} \times \left(1 + \frac{1}{\nu_j}\left(\frac{y_{ijk} - (\alpha_{ij} - \lambda_{ij} t_{ijk})}{\sigma_{\varepsilon j}}\right)^2\right)^{-\frac{\nu_j+1}{2}}$$

The above density function is that of a Student t distribution with degrees of freedom ν_j and scale parameter $\sigma^2_{\varepsilon j}$.

Accordingly, the unknown variance of the specified mixture distribution integrated out results in the Student t distribution [21]. In that sense, the Gibbs sampling algorithm with Student t distributed residuals can be implemented in a straightforward manner as its conjugacy of model parameters (conditional posterior distributions versus prior distributions) is similar to the model with residuals that are normally distributed. The specification of the Student t distribution as a mixture of random variables is implemented automatically in OpenBUGS [22].

2.6 Bayesian estimation and inference

A Bayesian method for the estimation of and inferences on model parameters is based on the use of information contained in the joint posterior distribution of the model parameters. The Bayesian approach is an alternative to classical methods. The difference between Bayesian and classical maximum likelihood (ML) estimation, is that in Bayesian estimation the unknown parameter is treated as a random variable. In Bayesian estimation parameter values are fully evaluated in the posterior distribution, whereas with classical ML estimation parameter values, and their corresponding standard errors are reported which maximise the corresponding likelihood [10]. Bayesian inferences are also advantageous in the sense that they do not depend on asymptotic approximations as classical inference methods do for complex models [1].

Let $\boldsymbol{y}$ be the data assumed to follow a parameter distribution with a probability density function $L(\boldsymbol{y}, \boldsymbol{\Theta})$ (which is the likelihood of the model parameters), where $\boldsymbol{\Theta} = (\Theta_1, \Theta_2, ..., \Theta_p)'$ is set of p unknown model parameters with probability density function $P(\boldsymbol{\Theta})$, called the prior distribution which expresses information available before any data collection. The Bayesian inference on the unknown model parameters $\boldsymbol{\Theta}$ involves the derivation of the posterior distribution $P(\boldsymbol{\Theta} \mid \boldsymbol{y})$ of the model parameters, given the data $\boldsymbol{y}$, based on the specified prior distribution.

The posterior distribution $P(\boldsymbol{\Theta} \mid \boldsymbol{y})$ of the model parameters in the Equation (8) can be obtained via the Bayes theorem, as follows:

$$P(\boldsymbol{\Theta} \mid \boldsymbol{y}) = \frac{L(\boldsymbol{y} \mid \boldsymbol{\Theta}) \cdot P(\boldsymbol{\Theta})}{\int L(\boldsymbol{y} \mid \boldsymbol{\Theta}) \cdot P(\boldsymbol{\Theta})\, d\boldsymbol{\Theta}} \tag{17}$$

In Bayesian inference, computation of posterior distributions is often a main challenge due to high dimension intractable integrals. The Bayesian methods are often implemented by the Markov Chain Monte Carlo (MCMC) Gibbs sampling algorithm [11]. The Gibbs sampler is a method which draws samples from the full conditional posterior distribution of the full set of model parameters in question. The full conditional distribution for each of the parameters in Equation (17) should be derived to enable the implementation of the Gibbs sampler. The Gibbs sampling algorithm can be implemented in a straightforward manner when conjugacy of model parameters exist (that is conditional posterior distributions are similar to the corresponding prior distributions). In practice, various software such as OpenBUGS, which is based on the BUGS (Bayesian inference Using Gibbs Sampling) project, are used to carry out the Gibbs sampling procedure [19]. The main condition before implementation is that the full posterior densities should be tractable.

Prior distributions for $\boldsymbol{\mu_j}, \Omega_{\boldsymbol{\mu j}}, \sigma^2_{\varepsilon j}$ and $\sigma^2_{\xi j}$

Firstly, the prior distribution for $\boldsymbol{\mu_j}$ and $\Omega_{\boldsymbol{\mu j}}$ are specified to respectively follow bivariate normal and Wishart prior distributions as follows:

$$\boldsymbol{\mu_j} \sim N_2\left(\mathbf{0}, 10^4 \times I_2\right) \tag{18}$$

$$\Omega^{-1}_{\boldsymbol{\mu j}} \sim W\left(2, 2 \times R_j\right) \tag{19}$$

where $\mathbf{0} = (0, 0)$ and I_2 denotes a 2×2 identity matrix and R_j represent 2×2 inverse scale matrices from the corresponding Wishart distribution. The posterior distributions of $\boldsymbol{\mu_j}$ and $\Omega_{\boldsymbol{\mu j}}$ are provided in the appendix of this research report.

Lastly, the prior distributions for both the variances $\sigma^2_{\varepsilon j}$ and $\sigma^2_{\xi j}$ are specified to follow gamma distributions:

$$\sigma^2_{\varepsilon j} \sim G\left(10^{-4}, 10^{-4}\right) \tag{20}$$

$$\sigma^2_{\xi j} \sim G\left(10^{-4}, 10^{-4}\right) \tag{21}$$

The posterior distributions of $\sigma^2_{\varepsilon j}$ and $\sigma^2_{\xi j}$ are provided in the appendix of this research report.

Posterior distributions

The joint posterior distribution of the complete set of regression model parameters is obtained by multiplying the associated likelihood functions and prior distributions (see appendix). The conditional posterior distributions of the model parameters are derived from the joint posterior distribution by ignoring terms that do not include the relevant model parameters (see appendix).

3 Application

This section presents applications of the robust linear mixed effects models that was fitted to CFU count data of EBA TB trials. This study was based on data of three TB trials. The corresponding results of the datasets analysed are discussed here in detail. The CFU data were collected over a period of 14 days in each of the EBA trials [7].

3.1 NC001 trial

3.1.1 Objectives

This was a Phase II, partially double-blind, randomised clinical trial that assessed the 14-day EBA, safety, tolerability, and PK of various combinations of TMC207, pyrazinamide, moxifloxacin, and Rifafour, in a total of 85 previously untreated drug susceptible TB patients [8]. EBA was described by the evaluation of CFU count.

3.1.2 Study design

Patients were randomised to receive either monotherapy of TMC207, combination therapy of TMC207 and pyrazinamide, combination therapy of TMC207 and PA-824, combination therapy of PA-824 and pyrazinamide, combination therapy of PA-824, moxifloxacin and pyrazinamide, or Rifafour. Overnight sputum samples were collected daily from Day 0 up to Day 14. The randomisation/treatment and sputum sampling schedule of the NC001 study are outlined in Table 1 below.

Scheduled Sample Days	Treatment Group	N
Daily from Day 0 to 14	J	15
	J-Z	15
	J-Pa	15
	Pa-Z	15
	Pa-Z-M	15
	Rifafour	10
	Total	85

Table 1: Treatment and sputum sampling schedule of NC001 trial

Note: Treatment group: J = TMC207, J-Z = TMC207 + Pyrazinamide, J-Pa = TMC207 + PA-824, Pa-Z = PA-824 + Pyrazinamide, Pa-Z-M or M-PA-Z = PA-824 + Pyrazinamide + Moxifloxacin. N = Total number of randomised patients.

3.1.3 Results

Table 2 shows the results of posterior estimates of EBA(0-14) and corresponding 95% Bayesian credibility interval (BCIs) for NC001 trial [4].

			Normal		Student t	
Parameter	Treatment Group	N	Posterior Estimate	95% BCI	Posterior Estimate	95% BCI
EBA(0-14)	J	15	0.076	[0.016, 0.143]	0.074	[0.010, 0.145]
	J-Z	15	0.135	[0.067, 0.206]	0.133	[0.065, 0.204]
	J-Pa	15	0.101	[0.057, 0.146]	0.101	[0.056, 0.146]
	Pa-Z	15	0.152	[0.098, 0.204]	0.154	[0.100, 0.207]
	Pa-Z-M	15	0.248	[0.085, 0.428]	0.248	[0.087, 0.430]
	Rifafour	10	0.142	[0.047, 0.238]	0.146	[0.055, 0.238]

Table 2: Posterior estimates of EBA(0-14) and corresponding 95% BCIs for NC001 trial

Note: BCI: Bayesian credibility interval. N = Total number of randomised patients. Posterior estimate: Represents the mean of the associated posterior distribution.

Table 2 above shows the posterior estimates and corresponding 95% BCIs of the normal and Student t models by treatment groups. For example, in treatment group J the posterior estimates and corresponding 95% BCIs for normal and Student t models are given as 0.076 (95% BCI: [0.016, 0.143]) and 0.074 (95% BCI: [0.010, 0.145]) respectively. The posterior estimates and corresponding 95% BCIs of the Student t model seem to be similar to those of normal model. The differences in posterior estimates and corresponding 95% BCI between the two models is negligible.

Table 3 shows the results of posterior estimates of ν_j (degrees of freedom) and corresponding 95% BCIs for NC001 trial.

Parameter	Treatment Group	N	Posterior Estimate	95% BCI
ν_j	J	15	4.599	[2.115, 12.170]
	J-Z	15	3.607	[2.157, 6.437]
	J-Pa	15	44.480	[4.053, 96.890]
	Pa-Z	15	18.060	[3.133, 86.110]
	Pa-Z-M	15	47.630	[5.766, 97.180]
	Rifafour	10	10.210	[2.238, 61.090]

Table 3: Posterior estimates of ν_j and corresponding 95% BCIs for NC001 trial

Note: BCI: Bayesian credibility interval. N = Total number of randomised patients. Posterior estimate: Represents the mean of the associated posterior distribution.

Table 3 above shows results of posterior estimates and corresponding 95% BCIs for the degrees of freedom (of residuals) by treatment group. The estimates for ν_j (the degrees of freedom parameter) are below 30 in 4 out of 6 cases, thus providing some evidence that the distribution of residuals in CFU counts is heavy tailed.

For further illustration purposes, Figure 2 graphically illustrates the nested plots of the observed CFU counts by treatment group [4]. Outliers in CFU count seem to be present in some treatment groups.

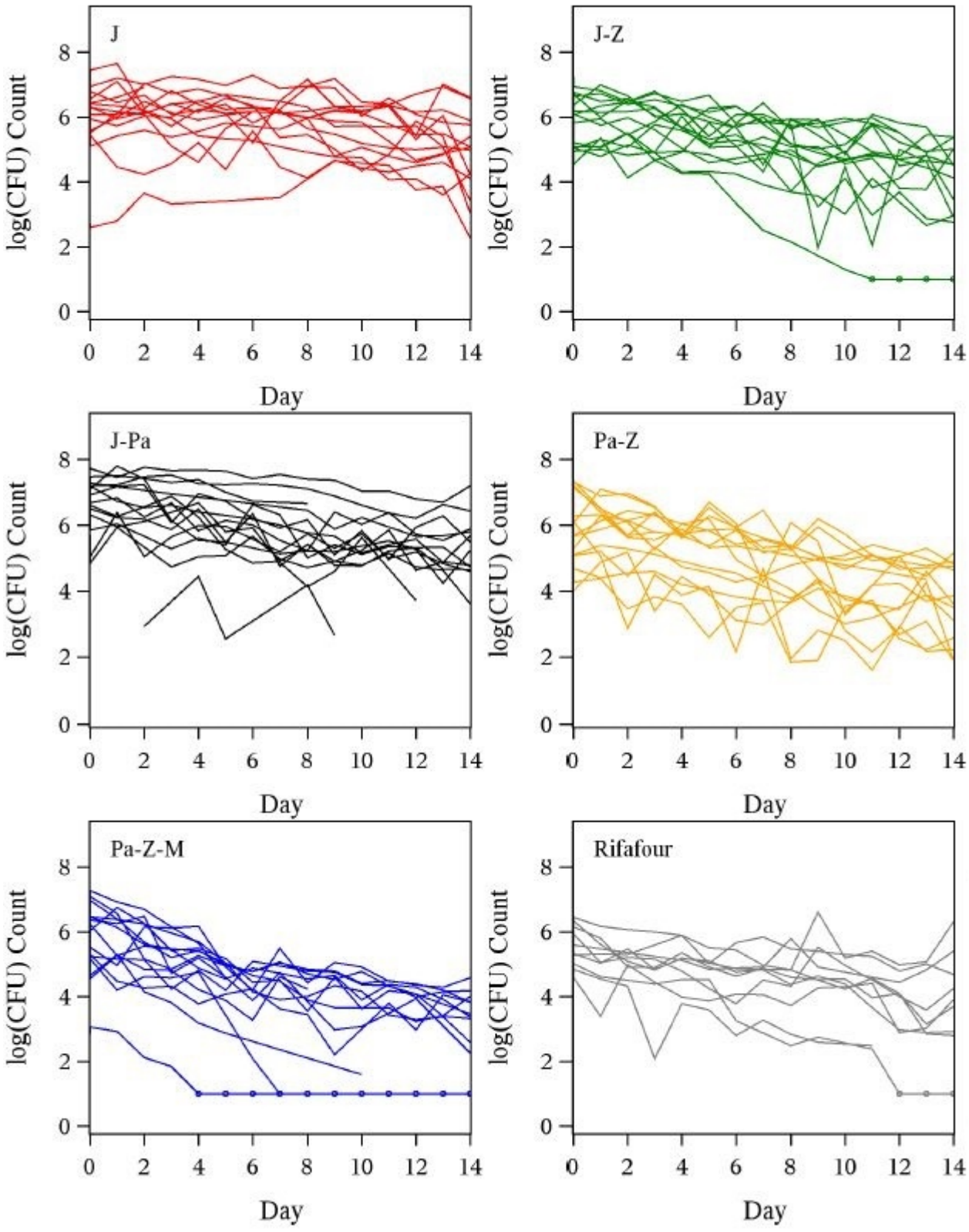


Figure 2: Observed log(CFU) counts over time of NC001 trial

3.2 NC003 trial

3.2.1 Objectives

This was a TB trial whose objectives included the evaluation of the safety, tolerability, PK and EBA of 14-day combination therapy of pyrazinamide, clofazimine, PA-824 and TMC207 in 105 previously untreated TB patients [8]. EBA was described by the evaluation of CFU count.

3.2.2 Study design

Patients were randomised to receive either daily doses of combination therapy of TMC207, PA-824, pyrazinamide and clofazimine, TMC207, PA-824 and pyrazinamide, TMC207, pyrazinamide and clofazimine, TMC207, pyrazinamide and clofazimine, monotherapy of clofazimine, and pyrazinamide, or Rifafour (control group) for 14 days. Overnight sputum samples were collected daily from Day 0 up to Day 14. The randomisation/treatment and sputum sampling of the NC003 study are outlined in Table 4 below.

Scheduled Sample Days	Treatment Group	N
Daily from Day 0 to 14	J-Pa-Z-C	14
	J-Pa-Z	14
	J-Z-C	15
	Z	14
	C	15
	Rifafour	15
	Total	87

Table 4: Treatment and sputum sampling schedule of NC003 trial

J-Pa-Z-C = TMC207 + PA-824 + Pyrazinamide + Clofazimine, J-Pa-Z = TMC207 + PA-824 + Pyrazinamide, J-Pa-C = TMC207 + PA-824 + Clofazimine, J-Z-C = TMC207 + Pyrazinamide + Clofazimine, Z = Pyrazinamide, C = Clofazimine.

3.2.3 Results

Table 5 shows the results of posterior estimates of EBA(0-14) and corresponding 95% BCIs for NC003 trial.

			Normal		Student t	
Parameter	Treatment Group	N	Posterior Estimate	95% BCI	Posterior Estimate	95% BCI
EBA(0-14)	J-Pa-Z-C	14	0.116	[0.050, 0.183]	0.115	[0.053, 0.178]
	J-Pa-Z	12	0.172	[0.063, 0.276]	0.164	[0.075, 0.272]
	J-Pa-C	15	0.083	[0.018, 0.149]	0.087	[0.024, 0.151]
	J-Z-C	14	0.101	[0.022, 0.183]	0.073	[0.004, 0.142]
	Z	15	0.036	[-0.019, 0.088]	0.038	[-0.012, 0.087]
	C	14	-0.022	[-0.077, 0.034]	-0.023	[-0.070, 0.023]
	Rifafour	15	0.152	[0.067, 0.241]	0.134	[0.066, 0.206]

Table 5: Posterior estimates of EBA(0-14) and corresponding 95% BCIs for NC003 trial

Note: BCI: Bayesian credibility interval. N = Total number of randomised patients. Posterior estimate: Represents the mean of the associated posterior distribution.

Table 5 above shows the posterior estimates and corresponding 95% BCIs of the normal and Student t models by treatment groups. For example, in treatment group J-Pa-Z posterior estimates and corresponding 95% BCIs of the normal and Student t models are given as 0.172 (95% BCI: [0.063, 0.276]) and 0.164 (95% BCI: [0.075, 0.272]) respectively. The posterior estimates of the Student t model are smaller with narrower 95% BCIs than those of the normal model.

Table 6 shows the results of posterior estimates of ν_j (degrees of freedom) and corresponding 95% BCIs for NC003 trial.

Parameter	Treatment Group	N	Posterior Estimate	95% BCI
ν_j	J-Pa-Z-C	14	6.123	[2.188, 22.220]
	J-Pa-Z	12	2.540	[2.019, 3.759]
	J-Pa-C	15	3.051	[2.064, 4.980]
	J-Z-C	14	2.408	[2.012, 3.414]
	Z	15	2.570	[2.023, 3.762]
	C	14	2.955	[2.063, 4.688]
	Rifafour	15	2.237	[2.007, 2.831]

Table 6: Posterior estimates of ν_j and corresponding 95% BCIs for NC003 trial

Note: BCI: Bayesian credibility interval. N = Total number of randomised patients. Posterior estimate: represents the mean of the associated posterior distribution. Table 6 above shows the posterior estimates and corresponding 95% BCIs for the degrees of freedom (of residuals) of the Student t model by treatment group. The estimates for ν_j (the degrees of freedom parameter) are below 30, providing evidence that the distribution of residuals in CFU counts is heavy tailed.

For further illustration purposes, Figure 3 graphically illustrates the nested plots of the observed CFU counts by treatment group [4].

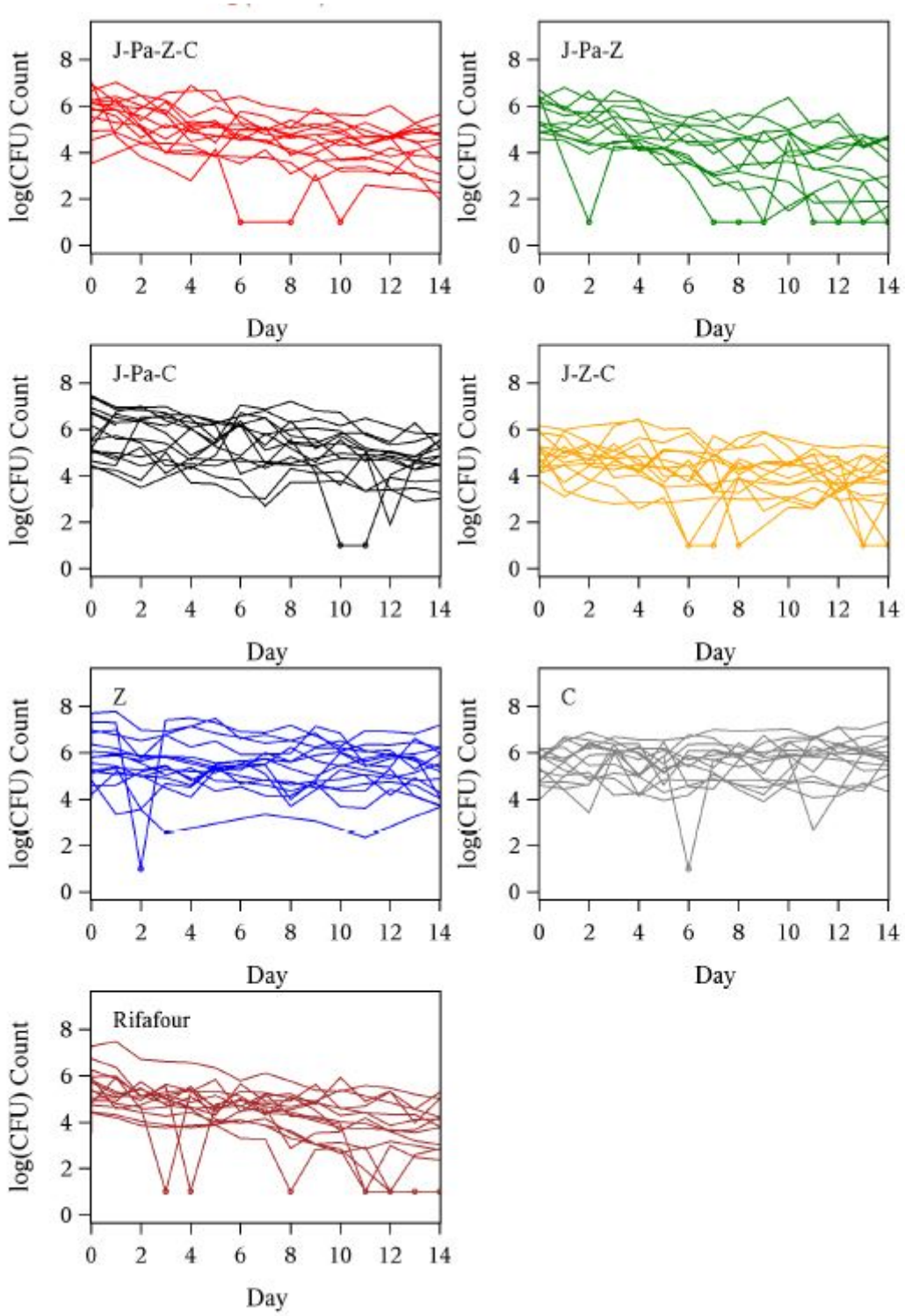


Figure 3: Observed log(CFU) counts over time of NC003 trial

Figure 3 above depict the observed CFU counts over time for different treatment groups. As can be seen for the case of treatment group J-Pa-Z-C that most of the CFU count at Day 6, 8, and 10 in individual profiles appear to be contaminated and implausible. The contamination and implausibility of CFU count indicating the presence of outliers in the data.

3.3 CL001 trial

3.3.1 Objectives

The CL001 study was based on a 14 day dose finding clinical trials for which the objectives are evaluation of the safety, tolerability, pharmacokinetics (PK) and EBA of various doses of TMC207 in 68 previously untreated TB patients [9]. The EBA was described by the evaluation of CFU count.

3.3.2 Study design

Patients were randomised to receive either daily doses of 100 mg TMC207, 200 mg TMC207, 300 mg TMC207, 400 mg TMC207 or Rifafour (control group) for 14 days. Overnight sputum samples were collected daily from Day 0 up to Day 8, and every second day from Day 10 up to Day 14. The randomisation/ treatment and sputum sampling of the CL001 study are outlined in Table 7 below.

Scheduled Sample Days	Treatment Group	N
Daily from Day 2 to 8;	TMC207 100 mg	15
Day 10, Day 12, Day 14	TMC207 200 mg	15
	TMC207 200 mg	15
	TMC207 400 mg	15
	Rifafour	8
	Total	68

Table 7: Treatment and sputum sampling schedule of CL001 trial

3.3.3 Results

Table 8 shows the results of posterior estimates of EBA(0-14) and corresponding 95% BCIs for CL001 trial.

			Normal		Student t	
Parameter	Treatment Group	N	Posterior Estimate	95% BCI	Posterior Estimate	95% BCI
EBA(0-14)	TMC207 100 mg	15	0.042	[0.001, 0.083]	0.041	[0.001, 0.082]
	TMC207 200 mg	15	0.059	[0.023, 0.097]	0.057	[0.022, 0.092]
	TMC207 200 mg	15	0.077	[0.028, 0.126]	0.078	[0.027, 0.128]
	TMC207 400 mg	15	0.098	[0.049, 0.146]	0.101	[0.052, 0.149]
	Rifafour	15	0.117	[0.042, 0.196]	0.111	[0.066, 0.206]

Table 8: Posterior estimates of EBA(0-14) and corresponding 95% BCIs for CL001 trial

Note: BCI: Bayesian credibility interval. N = Total number of randomised patients. Posterior estimate: Represents the mean of the associated posterior distribution.

Table 8 above shows the posterior estimates and corresponding 95% BCIs of the normal and Student t models by treatment groups. For example, in treatment group TMC207 100 mg the posterior estimates and corresponding 95% BCIs of the normal and Student t models are given as 0.042 (95% BCI: [0.001, 0.083]) and 0.041 (95% BCI: [0.001, 0.082]) respectively. Posterior estimates and corresponding 95% BCIs of the Student t model seem to be similar to those of the normal model. In other words, the difference in posterior estimates and corresponding 95% BCIs between the two models is negligible.

Table 9 shows results of posterior estimates of ν_j (degrees of freedom) and corresponding 95% BCIs of CL001 trial by treatment group.

Parameter	Treatment Group	N	Posterior Estimate	95% BCI
ν_j	TMC207 100 mg	15	42.39	[6.16, 96.32]
	TMC207 200 mg	15	6.58	[2.47, 21.48]
	TMC207 200 mg	15	38.84	[3.94, 96.01]
	TMC207 400 mg	15	3.15	[2.07, 5.52]
	Rifafour	15	14.44	[2.12, 86.77]

Table 9: Posterior estimates of ν_j and corresponding 95% BCIs for NC003 trial

Note: BCI: Bayesian credibility interval. N = Total number of randomised patients. Posterior estimate: represents the mean of the associated posterior distribution.

The posterior estimates for ν_j (the degrees of freedom parameter) are below 30 in 3 out of 5 cases, providing some evidence that the distribution of residuals in CFU counts is heavy tailed.

For further illustration purposes, Figure 4 graphically illustrates the nested plots of the observed CFU counts by treatment group [4]. Outliers in CFU count seem to be present in some treatment groups.

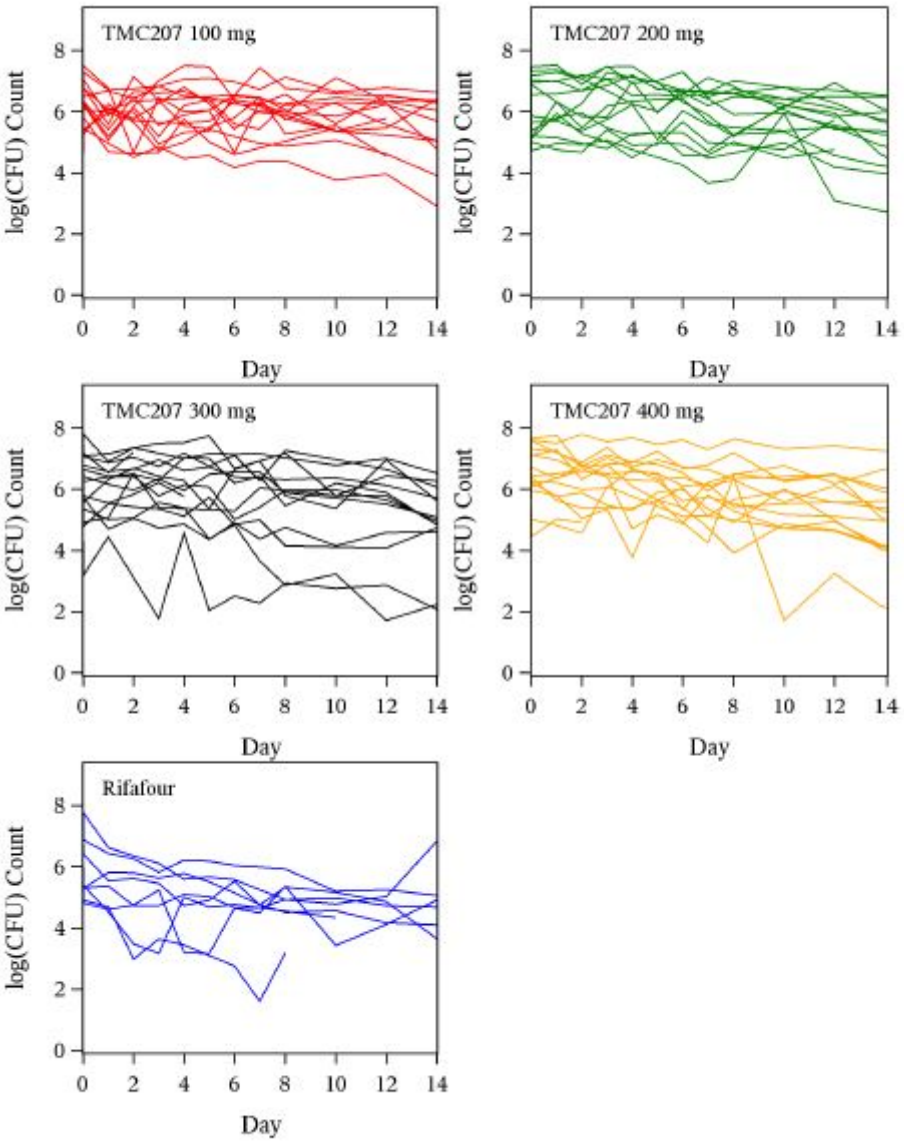


Figure 4: Observed log(CFU) counts over time of CL001 trial

3.4 Robust regression modeling

The EBA estimates computed from the joint Bayesian linear mixed effects regression models analysis are shrunken towards their corresponding mean estimates. Therefore, mixed effects regression modeling is preferred to analyse CFU count instead of regressing CFU count on a by-patient basis. However, extreme outliers in CFU count have previously shown to have a significant impact on the estimation of and inferences on EBA, despite the shrinkage effect. The heavy tailed Student t distribution is specified

for the residuals, and the associated estimates of degrees of freedom provide strong evidence that outliers in CFU count are present in the data. The specification of heavy tailed distribution clearly provides an even greater shrinkage effect compared to normal mixed effects regression modeling.

4 Conclusion

The EBA of TB drugs is conventionally assessed using statistical regression modeling of CFU count over time. It has been previously assumed that the residuals of the regression model are normally distributed which do not accommodate outliers in the distribution of data. In this report the normality assumption is relaxed by fitting a robust regression model which specifies the Student t distribution for the residuals. Outliers are occasionally present in CFU count due to erroneous sputum sampling. Such outliers can influence estimates of the rate of change in CFU count. A robust linear mixed effects regression model was fitted to CFU count of EBA TB trials to offer a robust approach that accommodates outliers.

A Bayesian framework was adopted for estimation of and inferences on model parameters using the Gibbs sampler in OpenBUGS. A Bayesian approach is an alternative to classical methods which in many cases is relatively easy to implement.

According to the study results of the three EBA TB trials, the adopted model provides accurate results of parameter estimates compared to the normal model when outliers are present in the data. On the other hand, the Student t model provides results that are similar to the normal model when few or no outliers are present in the data.

The research findings support the recommendation of the fit of robust linear mixed effects regression models to accommodate outliers in CFU count.

References

[1] James H Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

[2] George EP Box and George C Tiao. *Bayesian Inference in Statistical Analysis*. John Wiley & Sons, 1973.

[3] Helen Brown and Robin Prescott. Repeated measures data. *Applied Mixed Models in Medicine*, 2006.

[4] Divan A Burger. *Bayesian non-linear models for the bactericidal activity of tuberculosis drugs*. PhD thesis, University of the Free State, 2015.

[5] Divan A Burger and Robert Schall. A Bayesian nonlinear mixed-effects regression model for the characterization of early bactericidal activity of tuberculosis drugs. *Journal of Biopharmaceutical Statistics*, 25(6):1247–1271, 2015.

[6] Divan A Burger and Robert Schall. Robust fit of Bayesian mixed effects regression models with application to colony forming unit count in tuberculosis research. *Statistics in Medicine*, in press.

[7] Andreas H Diacon, Rodney Dawson, Florian Von Groote-Bidlingmaier, Gregory Symons, Amour Venter, Peter R Donald, Almari Conradie, Ngozi Erondu, Ann M Ginsberg, and Erica Egizi. Randomized dose-ranging study of the 14-day early bactericidal activity of bedaquiline (TMC207) in patients with sputum microscopy smear-positive pulmonary tuberculosis. *Antimicrobial Agents and Chemotherapy*, 57(5):2199–2203, 2013.

[8] Andreas H Diacon, Rodney Dawson, Florian von Groote-Bidlingmaier, Gregory Symons, Amour Venter, Peter R Donald, Christo van Niekerk, Daniel Everitt, Jane Hutchings, and Divan A Burger. Bactericidal activity of pyrazinamide and clofazimine alone and in combinations with pretomanid and bedaquiline. *American Journal of Respiratory and Critical Care Medicine*, 191(8):943–953, 2015.

[9] Andreas H Diacon, Rodney Dawson, Florian von Groote-Bidlingmaier, Gregory Symons, Amour Venter, Peter R Donald, Christo van Niekerk, Daniel Everitt, Helen Winter, and Piet Becker. 14-Day bactericidal activity of PA-824, bedaquiline, pyrazinamide, and moxifloxacin combinations: A randomised trial. *The Lancet*, 380(9846):986–993, 2012.

[10] Simon Farrell and Casimir JH Ludwig. Bayesian and maximum likelihood estimation of hierarchical response time models. *Psychonomic Bulletin & Review*, 15(6):1209–1217, 2008.

[11] Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990.

[12] Walter R Gilks, Sylvia Richardson, and David J Spiegelhalter. Introducing Markov Chain Monte Carlo. *Markov Chain Monte Carlo in Practice*, 1:19, 1996.

[13] Stephen H Gillespie, Roland D Gosling, and Bambos M Charalambous. A reiterative method for calculating the early bactericidal activity of antituberculosis drugs. *American Journal of Respiratory and Critical Care Medicine*, 166(1):31–35, 2002.

[14] Roly D Gosling, Leonid Heifets, and Stephen H Gillespie. A multicentre comparison of a novel surrogate marker for determining the specific potency of anti-tuberculosis drugs. *Journal of Antimicrobial Chemotherapy*, 52(3):473–476, 2003.

[15] MaryAnn Hill and WJ Dixon. Robustness in real life: A study of clinical laboratory data. *Biometrics*, 46:377–396, 1982.

[16] Amina Jindani, Caroline J Doré, and Denis A Mitchison. Bactericidal and sterilizing activities of antituberculosis drugs during the first 14 days. *American Journal of Respiratory and Critical Care Medicine*, 167(10):1348–1354, 2003.

[17] Hongjing Liao, Yanju Li, and Gordon Brooks. Outlier impact and accommodation methods: Multiple comparisons of type I error rates. *Journal of Modern Applied Statistical Methods*, 15(1):23, 2016.

[18] Mary J Lindstrom and Douglas M Bates. Nonlinear mixed effects models for repeated measures data. *Biometrics*, 46:673–687, 1990.

[19] David Lunn, David Spiegelhalter, Andrew Thomas, and Nicky Best. The BUGS project: Evolution, critique and future directions. *Statistics in Medicine*, 28(25):3049–3067, 2009.

[20] Ioannis Ntzoufras. *Bayesian Modeling Using WinBUGS*. John Wiley & Sons, 2011.

[21] Simon JD Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, 2012.

[22] Sujit K Sahu, Dipak K Dey, and Márcia D Branco. A new class of multivariate skew distributions with applications to bayesian regression models. *Canadian Journal of Statistics*, 31(2):129–150, 2003.

[23] Richard N van Zyl-Smit, Anke Binder, Richard Meldau, Hridesh Mishra, Patricia L Semple, Grant Theron, Jonathan Peter, Andrew Whitelaw, Suren K Sharma, and Robin Warren. Comparison of quantitative techniques including Xpert MTB/RIF to evaluate mycobacterial burden. *PLoS ONE*, 6(12):e28815, 2011.

Appendix

In this section the conditional posterior distributions of the model parameters are derived from the joint posterior distribution by ignoring terms that do not include the relevant model parameter.

$$\boldsymbol{\mu_{ij}} = (\alpha_{ij}, \lambda_{ij})'$$

$$\boldsymbol{\mu_j} = (\alpha_j, \lambda_j)'$$

$$\Omega_{\boldsymbol{\mu j}} = \begin{bmatrix} \sigma^2_{\alpha j} & \sigma_{\alpha j \lambda j} \\ \sigma_{\alpha j \lambda j} & \sigma^2_{\lambda j} \end{bmatrix}$$

Full likelihood

$$L\left(\boldsymbol{\mu_{ij}}, \boldsymbol{\mu_j}, \Omega_{\boldsymbol{\mu j}}, \sigma^2_{\varepsilon j}, \sigma^2_{\xi j}, i = 1, ..., N, j = 1, ..., J, k = 1, ..., K_{ij} | \boldsymbol{y}\right) \tag{22}$$

$$= \left(\prod_{i=1}^{N} \prod_{\substack{j=1 \\ i \in \{j\}}}^{J} L\left(\boldsymbol{\mu_{ij}}, \sigma^2_{\varepsilon j}, \sigma^2_{\xi j}, k = 1, ..., K_{ij} | \boldsymbol{y_{ij}}\right) \right) \cdot \prod_{i=1}^{N} \prod_{\substack{j=1 \\ i \in \{j\}}}^{J} P\left(\boldsymbol{\mu_{ij}} | \boldsymbol{\mu_j}, \Omega_{\boldsymbol{\mu j}}\right)$$

where:

$$L\left(\boldsymbol{\mu_{ij}}, \sigma^2_{\varepsilon j}, \sigma^2_{\xi j}, k = 1, ..., K_{ij} | \boldsymbol{y_{ij}}\right)$$

$$\propto \prod_{k=1}^{K_{ij}} \frac{1}{\sqrt{2\pi}\sigma_{\xi j}\sigma_{\varepsilon j}} \exp\left[-\frac{1}{2} \frac{\left(y_{ijk} - (\alpha_{ij} - \lambda_{ij} \cdot t_{ijk})\right)^2}{\sigma^2_{\xi j} \cdot \sigma^2_{\varepsilon j}} \right] \cdot \frac{\left(\frac{\nu_j}{2}\right)^{\frac{\nu_j}{2}}}{\Gamma\left(\frac{\nu_j}{2}\right)} \left(\sigma^{-2}_{\xi j}\right)^{\frac{\nu_j}{2}-1} \exp\left(-\frac{\nu_j}{2\sigma^2_{\xi j}}\right)$$

Therefore:

$$L\left(\boldsymbol{\mu_{ij}}, \boldsymbol{\mu_j}, \Omega_{\boldsymbol{\mu j}}, \sigma^2_{\varepsilon j}, \sigma^2_{\xi j}, i = 1, ..., N, j = 1, ..., J, k = 1, ..., K_{ij} | \boldsymbol{y}\right)$$

$$\propto \left(\prod_{j=1}^{J} \left(\sigma^2_{\xi j} \cdot \sigma^2_{\varepsilon j}\right)^{-\frac{1}{2}T_j} \cdot \left(\frac{\left(\frac{\nu_j}{2}\right)^{\frac{\nu_j}{2}}}{\Gamma\left(\frac{\nu_j}{2}\right)} \left(\sigma^{-2}_{\xi j}\right)^{\frac{\nu_j}{2}-1} \right)^{T_j} \right) \cdot \exp\left(-\frac{1}{2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \in \{j\}}}^{J} \sum_{k=1}^{K_{ij}} \left\{ \left[\frac{\left(y_{ijk} - (\alpha_{ij} - \lambda_{ij} \cdot t_{ijk})\right)^2}{\sigma^2_{\xi j} \cdot \sigma^2_{\varepsilon j}} \right] \right\} \right) \cdot$$

$$\prod_{i=1}^{N} \prod_{\substack{j=1 \\ i \in \{j\}}}^{J} \left(\mid \Omega_{\boldsymbol{\mu j}} \mid^{-\frac{1}{2}} \exp\left[-\frac{1}{2} \left(\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_j}\right)' \cdot \Omega^{-1}_{\boldsymbol{\mu j}} \cdot \left(\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_j}\right) \right] \right)$$

where T_j is the total number of time points across all patients allocated to treatment group j.

Joint prior distribution

$$P\left(\boldsymbol{\mu_j}, \Omega_{\boldsymbol{\mu j}}, \sigma^2_{\varepsilon j}, \sigma^2_{\xi j}, \nu_j, j = 1, ..., J\right) \tag{23}$$

$$\prod_{i=1}^{J} \left(P\left[\boldsymbol{\mu_j}\right] \cdot P\left[\Omega^{-1}_{\boldsymbol{\mu j}}\right] \cdot P\left[\sigma^2_{\varepsilon j}\right] .P\left[\sigma^2_{\xi j}\right] \cdot P\left[\nu_j\right] \right)$$

$$\propto \prod_{j=1}^{J} \left(\exp\left[-\frac{1}{2} \cdot \boldsymbol{\mu_j'} \cdot \frac{1}{10} \cdot \boldsymbol{\mu_j} \right] \cdot \mid \Omega^{-1}_{\boldsymbol{\mu j}} \mid \cdot etr\left[R_j \cdot \Omega^{-1}_{\boldsymbol{\mu j}} \right] \cdot \left(\sigma^{-2}_{\varepsilon j}\right)^{\left(10^{-4}-1\right)} \cdot \exp\left(-10^{-4} \cdot \sigma^{-2}_{\varepsilon j}\right) \right) \cdot$$

$$\left(\sigma^{-2}_{\xi j}\right)^{\left(10^{-4}-1\right)} \cdot \exp\left(-10^{-4} \cdot \sigma^{-2}_{\xi j}\right) \cdot I\left(2 \leq \nu_j \leq 100\right)$$

Joint posterior distribution

$$P\left(\boldsymbol{\mu_{ij}}, \boldsymbol{\mu_j}, \Omega_{\boldsymbol{\mu j}}, \sigma^2_{\varepsilon j}, \sigma^2_{\xi j}, \nu_j, i = 1, ..., N, j = 1, ..., J, k = 1, ..., K_{ij} | \boldsymbol{y}\right) \tag{24}$$

$$\propto L\left(\boldsymbol{\mu_{ij}}, \boldsymbol{\mu_j}, \Omega_{\boldsymbol{\mu j}}, \sigma^2_{\varepsilon j}, \sigma^2_{\xi j}, i = 1, ..., N, j = 1, ..., J, k = 1, ..., K_{ij} | \boldsymbol{y}\right) \cdot P\left(\boldsymbol{\mu_j}, \Omega_{\boldsymbol{\mu j}}, \sigma^2_{\varepsilon j}, \sigma^2_{\xi j}, \nu_j, j = 1, ..., J\right)$$

$$\propto \left(\prod_{j=1}^{J} \left(\sigma^2_{\xi j} \cdot \sigma^2_{\varepsilon j}\right)^{-\frac{1}{2}T_j} \cdot \left(\frac{\left(\frac{\nu_j}{2}\right)^{\frac{\nu_j}{2}}}{\Gamma\left(\frac{\nu_j}{2}\right)} \left(\sigma^{-2}_{\xi j}\right)^{\frac{\nu_j}{2}-1} \right)^{T_j} \right) \cdot \exp\left(-\frac{1}{2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \in \{j\}}}^{J} \sum_{k=1}^{K_{ij}} \left[\frac{\left(y_{ijk} - \left(\alpha_{ij} - \lambda_{ij} \cdot t_{ijk}\right)\right)^2}{\sigma^2_{\xi j} \cdot \sigma^2_{\varepsilon j}} \right] \right) \cdot$$

$$\left(\prod_{j=1}^{J} \mid \Omega_{\boldsymbol{\mu j}} \mid^{-\frac{1}{2}N_j} \right) \cdot \exp\left[-\frac{1}{2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \in \{j\}}}^{J} \left(\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_j}\right)' \cdot \Omega^{-1}_{\boldsymbol{\mu j}} \cdot \left(\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_j}\right) \right] \cdot \left(\sigma^{-2}_{\xi j}\right)^{\left(10^{-4}-1\right)} \cdot \exp\left(-10^{-4} \cdot \sigma^{-2}_{\xi j}\right) \cdot$$

$$\exp\left(-\frac{1}{2} \sum_{j=1}^{J} \left[\boldsymbol{\mu_j'} \cdot \frac{1}{10} \cdot \boldsymbol{\mu_j} \right] \right) \cdot \left(\prod_{j=1}^{J} \mid \Omega^{-1}_{\boldsymbol{\mu j}} \mid^{-\frac{1}{2}} \right) \cdot etr\left(\sum_{j=1}^{J} R_j \cdot \Omega^{-1}_{\boldsymbol{\mu j}} \right) \cdot$$

$$\left(\sigma^{-2}_{\varepsilon j}\right)^{\left(10^{-4}-1\right)} \cdot \exp\left(-10^{-4} \cdot \sigma^{-2}_{\varepsilon j}\right) \cdot I\left(2 \leq \nu_j \leq 100\right)$$

Conditional postrior distribution

$$P\left(\boldsymbol{\mu_{ij}}|\boldsymbol{\mu_j}, \Omega_{\boldsymbol{\mu j}}, \sigma^2_{\varepsilon j}, \sigma^2_{\xi j}, \boldsymbol{y}\right) \tag{25}$$

$$\propto \exp\left(-\frac{1}{2}\sum_{k=1}^{K_{ij}}\left[\frac{\left(y_{ijk}-\left(\alpha_{ij}-\lambda_{ij}\cdot t_{ijk}\right)\right)^2}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\right]\right)\cdot\exp\left(-\frac{1}{2}\left(\boldsymbol{\mu_{ij}}-\boldsymbol{\mu_j}\right)^{'}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\left(\boldsymbol{\mu_{ij}}-\boldsymbol{\mu_j}\right)\right)$$

$$L\left(\boldsymbol{\mu_{ij}}, \sigma^2_{\varepsilon j}, \sigma^2_{\xi j}, k=1,...,K_{ij}|\boldsymbol{y_{ij}}\right) \tag{26}$$

$$\propto \exp\left(-\frac{1}{2}\sum_{k=1}^{K_{ij}}\left[\frac{\left(y_{ijk}-\left(\alpha_{ij}-\lambda_{ij}\cdot t_{ijk}\right)\right)^2}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\right]\right)$$

$$\propto \frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\left(\boldsymbol{y_{ij}}-X_{ij}\cdot\boldsymbol{\mu_{ij}}\right)^{'}\left(\boldsymbol{y_{ij}}-X_{ij}\cdot\boldsymbol{\mu_{ij}}\right)$$

where X_{ij} is a $K_{ij}\times 2$ matrix as follows:

$$X_{ij}=\begin{bmatrix}1 & -t_{ij1}\\ . & .\\ . & .\\ . & .\\ 1 & -t_{ijk}\\ . & .\\ . & .\\ . & .\\ 1 & -t_{ijK_{ij}}\end{bmatrix}$$

Define $\boldsymbol{B_{ij}}$ and $\boldsymbol{s_{ij}}$ as follows:

$$\boldsymbol{B_{ij}}=\left(X^{'}_{ij}\cdot X_{ij}\right)^{-1}\cdot X^{'}_{ij}\cdot\boldsymbol{y_{ij}} \tag{27}$$

$$\boldsymbol{s_{ij}}=\left(\boldsymbol{y_{ij}}-X_{ij}\cdot\boldsymbol{B_{ij}}\right)^{'}\cdot\left(\boldsymbol{y_{ij}}-X_{ij}\cdot\boldsymbol{B_{ij}}\right) \tag{28}$$

Making use of the following algebraic identity for Equation (26) :

$$\left(\boldsymbol{y_{ij}}-X_{ij}\cdot\boldsymbol{\mu_{ij}}\right)^{'}\cdot\left(\boldsymbol{y_{ij}}-X_{ij}\cdot\boldsymbol{\mu_{ij}}\right) \tag{29}$$

$$= (\boldsymbol{y_{ij}} - X_{ij} \cdot \boldsymbol{B_{ij}} - X_{ij} \cdot [\boldsymbol{\mu_{ij}} - \boldsymbol{B_{ij}}])^{'} \cdot (\boldsymbol{y_{ij}} - X_{ij} \cdot \boldsymbol{B_{ij}} - X_{ij} \cdot [\boldsymbol{\mu_{ij}} - \boldsymbol{B_{ij}}])$$

$$= (\boldsymbol{y_{ij}} - X_{ij} \cdot \boldsymbol{B_{ij}})^{'} \cdot (\boldsymbol{y_{ij}} - X_{ij} \cdot \boldsymbol{B_{ij}}) + (\boldsymbol{\mu_{ij}} - \boldsymbol{B_{ij}})^{'} \cdot X_{ij}^{'} \cdot X_{ij} \cdot (\boldsymbol{\mu_{ij}} - \boldsymbol{B_{ij}})$$

$$= \boldsymbol{s_{ij}} + (\boldsymbol{\mu_{ij}} - \boldsymbol{B_{ij}})^{'} \cdot X_{ij}^{'} \cdot X_{ij} \cdot (\boldsymbol{\mu_{ij}} - \boldsymbol{B_{ij}})$$

since the corresponding terms equals:

$$= (\boldsymbol{\mu_{ij}} - \boldsymbol{B_{ij}})^{'} \cdot X_{ij}^{'} \cdot (\boldsymbol{y_{ij}} - X_{ij} \cdot \boldsymbol{B_{ij}}) \tag{30}$$

$$= (\boldsymbol{\mu_{ij}} - \boldsymbol{B_{ij}})^{'} \cdot \left(X_{ij}^{'} \cdot \boldsymbol{y_{ij}} - X_{ij}^{'} \cdot X_{ij} \cdot \boldsymbol{B_{ij}}\right)$$

$$= (\boldsymbol{\mu_{ij}} - \boldsymbol{B_{ij}})^{'} \cdot \left(X_{ij}^{'} \cdot \boldsymbol{y_{ij}} - X_{ij}^{'} \cdot X_{ij} \cdot \left[X_{ij}^{'} \cdot X_{ij}\right]^{-1} \cdot X_{ij}^{'} \cdot \boldsymbol{y_{ij}}\right)$$

$$= (\boldsymbol{\mu_{ij}} - \boldsymbol{B_{ij}})^{'} \cdot \left(X_{ij}^{'} \cdot \boldsymbol{y_{ij}} - X_{ij}^{'} \cdot \boldsymbol{y_{ij}}\right)$$

$$= \mathbf{0}$$

Finally, from Equation (29) , Equation (26) can be written as follows:

$$L\left(\boldsymbol{\mu_{ij}}, \sigma_{\varepsilon j}^2, \sigma_{\xi j}^2, k = 1, ..., K_{ij} | \boldsymbol{y_{ij}}\right) \tag{31}$$

$$\propto \exp\left(-\frac{1}{2 \cdot \sigma_{\xi j}^2 \cdot \sigma_{\varepsilon j}^2}\left[\boldsymbol{s_{ij}} + (\boldsymbol{\mu_{ij}} - \boldsymbol{B_{ij}})^{'} \cdot X_{ij}^{'} \cdot X_{ij} \cdot (\boldsymbol{\mu_{ij}} - \boldsymbol{B_{ij}})\right]\right)$$

$$P\left(\boldsymbol{\mu_{ij}} | \boldsymbol{\mu_j}, \Omega_{\boldsymbol{\mu j}}, \boldsymbol{y}\right) \tag{32}$$

$$\propto L\left(\boldsymbol{\mu_{ij}}, \sigma_{\varepsilon j}^2, \sigma_{\xi j}^2, k = 1, ...K_{ij} | \boldsymbol{y_{ij}}\right) . \exp\left(-\frac{1}{2}\left(\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_j}\right)^{'} \cdot \Omega_{\boldsymbol{\mu j}}^{-1} \cdot \left(\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_j}\right)\right)$$

$$\propto \exp\left(-\frac{1}{2\cdot\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\left[\boldsymbol{s_{ij}}+(\boldsymbol{\mu_{ij}}-\boldsymbol{B_{ij}})^{'}\cdot X^{'}_{ij}\cdot X_{ij}\cdot(\boldsymbol{\mu_{ij}}-\boldsymbol{B_{ij}})\right]\right).\exp\left(-\frac{1}{2}(\boldsymbol{\mu_{ij}}-\boldsymbol{\mu_j})^{'}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot(\boldsymbol{\mu_{ij}}-\boldsymbol{\mu_j})\right)$$

Completing the square inside $\exp\left(-\frac{1}{2}[\cdot]\right)$ of Equation (32) :

$$\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\left[(\boldsymbol{\mu_{ij}}-\boldsymbol{B_{ij}})^{'}\cdot X^{'}_{ij}\cdot X_{ij}\cdot(\boldsymbol{\mu_{ij}}-\boldsymbol{B_{ij}})\right]+(\boldsymbol{\mu_{ij}}-\boldsymbol{\mu_j})^{'}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot(\boldsymbol{\mu_{ij}}-\boldsymbol{\mu_j})$$

$$=\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\left(\boldsymbol{\mu^{'}_{ij}}\cdot X^{'}_{ij}\cdot X_{ij}\cdot\boldsymbol{\mu_{ij}}-\boldsymbol{\mu^{'}_{ij}}\cdot X^{'}_{ij}\cdot X_{ij}\boldsymbol{B_{ij}}-\boldsymbol{B^{'}_{ij}}\cdot X^{'}_{ij}\cdot X_{ij}\cdot\boldsymbol{\mu_{ij}}+\boldsymbol{B^{'}_{ij}}\cdot X^{'}_{ij}\cdot X_{ij}\cdot\boldsymbol{B_{ij}}\right)+$$

$$\boldsymbol{\mu^{'}_{ij}}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_{ij}}-\boldsymbol{\mu^{'}_{ij}}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j}-\boldsymbol{\mu^{'}_j}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_{ij}}+\boldsymbol{\mu^{'}_j}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j}$$

$$=\boldsymbol{\mu^{'}_{ij}}\left(\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\cdot X^{'}_{ij}\cdot X_{ij}+\Omega^{-1}_{\boldsymbol{\mu j}}\right)\cdot\boldsymbol{\mu_{ij}}-\boldsymbol{\mu^{'}_{ij}}\left(\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}X^{'}_{ij}\cdot X_{ij}\cdot\boldsymbol{B_{ij}}+\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j}\right)-$$

$$\left(\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\cdot\boldsymbol{B^{'}_{ij}}\cdot X^{'}_{ij}\cdot X_{ij}+\boldsymbol{\mu^{'}_j}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\right)\cdot\boldsymbol{\mu_{ij}}+\left(\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\boldsymbol{B^{'}_{ij}}\cdot X^{'}_{ij}\cdot X_{ij}\cdot\boldsymbol{B_{ij}}+\boldsymbol{\mu^{'}_j}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j}\right)$$

$$=\boldsymbol{\mu^{'}_{ij}}\cdot V^{-1}_{ij}\cdot\boldsymbol{\mu_{ij}}-\boldsymbol{\mu^{'}_{ij}}\cdot V^{-1}_{ij}\cdot\boldsymbol{O_{ij}}-\boldsymbol{O^{'}_{ij}}\cdot V^{-1}_{ij}\cdot\boldsymbol{\mu_{ij}}+\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\cdot\boldsymbol{B^{'}_{ij}}\cdot X^{'}_{ij}\cdot X_{ij}\cdot\boldsymbol{B_{ij}}+\boldsymbol{\mu^{'}_j}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j}$$

$$=\boldsymbol{\mu^{'}_{ij}}\cdot V^{-1}_{ij}\cdot\boldsymbol{\mu_{ij}}-\boldsymbol{\mu^{'}_{ij}}\cdot V^{-1}_{ij}\cdot\boldsymbol{O_{ij}}-\boldsymbol{O^{'}_{ij}}\cdot V^{-1}_{ij}\cdot\boldsymbol{\mu_{ij}}+\boldsymbol{O^{'}_{ij}}\cdot V^{-1}_{ij}\cdot\boldsymbol{O_{ij}}-\boldsymbol{O^{'}_{ij}}\cdot V^{-1}_{ij}\cdot\boldsymbol{O_{ij}}+$$

$$\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\cdot\boldsymbol{B^{'}_{ij}}\cdot X^{'}_{ij}\cdot X_{ij}\cdot\boldsymbol{B_{ij}}+\boldsymbol{\mu^{'}_j}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j}$$

$$=(\boldsymbol{\mu_{ij}}-\boldsymbol{O_{ij}})^{'}\cdot V^{-1}_{ij}\cdot(\boldsymbol{\mu_{ij}}-\boldsymbol{O_{ij}})-\boldsymbol{O^{'}_{ij}}\cdot V^{-1}_{ij}\cdot\boldsymbol{O_{ij}}+$$

$$\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\cdot\boldsymbol{B^{'}_{ij}}\cdot X^{'}_{ij}\cdot X_{ij}\cdot\boldsymbol{B_{ij}}+\boldsymbol{\mu^{'}_j}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j}$$

where $V_{ij} = \left(\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\cdot X_{ij}^{'}\cdot X_{ij} + \Omega^{-1}_{\boldsymbol{\mu j}}\right)^{-1}$ and $\boldsymbol{O_{ij}} = V_{ij}\cdot\left(\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\cdot X_{ij}^{'}\cdot X_{ij}\cdot\boldsymbol{B_{ij}} + \Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j}\right) = V_{ij}\cdot\left(\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\cdot X_{ij}^{'}\cdot\boldsymbol{y_{ij}} + \Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j}\right)$

Thus, the conditional posterior distribution of $\boldsymbol{\mu_{ij}}|\boldsymbol{\mu_j}, \Omega^{-1}_{\boldsymbol{\mu j}}, \boldsymbol{y}$ is given by:

$$P\left(\boldsymbol{\mu_{ij}}|\boldsymbol{\mu_j}, \Omega^{-1}_{\boldsymbol{\mu j}}, \boldsymbol{y}\right) \propto \exp\left(-\frac{1}{2}\left(\boldsymbol{\mu_{ij}} - \boldsymbol{O_{ij}}\right)^{'}\cdot V_{ij}^{-1}\cdot\left(\boldsymbol{\mu_{ij}} - \boldsymbol{O_{ij}}\right)\right) \tag{33}$$

From Equation (33), the posterior distribution of $\boldsymbol{\mu_{ij}}|\boldsymbol{\mu_j}, \Omega^{-1}_{\boldsymbol{\mu j}}, \sigma^2_{\xi j}, \sigma^2_{\varepsilon j}, \boldsymbol{y}$ is therefore as follows:

$$\boldsymbol{\mu_{ij}}|\boldsymbol{\mu_j}, \Omega^{-1}_{\boldsymbol{\mu j}}, \sigma^2_{\xi j}, \sigma^2_{\varepsilon j}, \boldsymbol{y} \sim N\left(\boldsymbol{O_{ij}}, V_{ij}\right) \tag{34}$$

$$N\left(\left[\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\cdot X_{ij}^{'}\cdot X_{ij} + \Omega^{-1}_{\boldsymbol{\mu j}}\right]^{-1}\cdot\left[\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\cdot X_{ij}^{'}\cdot\boldsymbol{y_{ij}} + \Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j}\right], \left[\frac{1}{\sigma^2_{\xi j}\cdot\sigma^2_{\varepsilon j}}\cdot X_{ij}^{'}\cdot X_{ij} + \Omega^{-1}_{\boldsymbol{\mu j}}\right]^{-1}\right)$$

$$P\left(\boldsymbol{\mu_j}|\boldsymbol{\mu_{ij}}, \Omega^{-1}_{\boldsymbol{\mu j}}, \boldsymbol{y}, i\left[i \in \{j\}\right] = 1, ..., N\right) \tag{35}$$

$$\propto \exp\left(-\frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\left(\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_j}\right)^{'}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\left(\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_j}\right)\right)\cdot\exp\left(-\frac{1}{2}\sum_{j=1}^{J}\left[\boldsymbol{\mu_j^{'}}\cdot\frac{1}{10^4}\cdot\boldsymbol{\mu_j}\right]\right)$$

Completing the square inside $\exp\left(-\frac{1}{2}\left[\cdot\right]\right)$ of Equation (35) :

$$\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\left(\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_j}\right)^{'}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\left(\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_j}\right) + \sum_{i=1}^{J}\left(\boldsymbol{\mu_j^{'}}\cdot\frac{1}{10^4}\cdot\boldsymbol{\mu_j}\right)$$

$$=\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\left(\boldsymbol{\mu_{ij}^{'}}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_{ij}^{'}}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j} - \boldsymbol{\mu_{ij}^{'}}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_{ij}} + \boldsymbol{\mu_j^{'}}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j}\right) + \sum_{j=1}^{J}\left(\boldsymbol{\mu_j^{'}}\cdot\frac{1}{10^4}\cdot\boldsymbol{\mu_j}\right)$$

$$=\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\boldsymbol{\mu_{ij}^{'}}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_{ij}} - 2\sum_{j=1}^{J}\boldsymbol{\mu_j^{'}}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\boldsymbol{\mu_{ij}} + \sum_{j=1}^{J}N_j\cdot\boldsymbol{\mu_j^{'}}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_j} + \sum_{j=1}^{J}\boldsymbol{\mu_j^{'}}\cdot\frac{1}{10^4}\cdot\boldsymbol{\mu_j}$$

$$=\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\boldsymbol{\mu_{ij}^{'}}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\cdot\boldsymbol{\mu_{ij}} + \sum_{j=1}^{J}\boldsymbol{\mu_j^{'}}\cdot\left(N_j\cdot\Omega^{-1}_{\boldsymbol{\mu j}} + \frac{1}{10^4}\right)\cdot\boldsymbol{\mu_j} - 2\sum_{j=1}^{J}\boldsymbol{\mu_j^{'}}\cdot\Omega^{-1}_{\boldsymbol{\mu j}}\sum_{j=1}^{J}\boldsymbol{\mu_{ij}}$$

$$=\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\boldsymbol{\mu_{ij}'}\cdot\Omega_{\boldsymbol{\mu j}}^{-1}\cdot\boldsymbol{\mu_{ij}}+\sum_{j=1}^{J}\boldsymbol{\mu_j'}\cdot D_j^{-1}\cdot\boldsymbol{\mu_j}-2\sum_{j=1}^{J}\boldsymbol{\mu_j'}\cdot D_j^{-1}\cdot\boldsymbol{E_j}$$

$$=\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\boldsymbol{\mu_{ij}'}\cdot\Omega_{\boldsymbol{\mu j}}^{-1}\cdot\boldsymbol{\mu_{ij}}+\sum_{j=1}^{J}\boldsymbol{\mu_j'}\cdot D_j^{-1}\cdot\boldsymbol{\mu_j}-2\cdot\sum_{j=1}^{J}\boldsymbol{\mu_j'}\cdot D_j^{-1}\cdot\boldsymbol{E_j}+\sum_{j=1}^{J}\boldsymbol{E_j'}\cdot D_j^{-1}\cdot\boldsymbol{E_j}-\sum_{j=1}^{J}\boldsymbol{E_j'}\cdot D_j^{-1}\cdot\boldsymbol{E_j}$$

$$=\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\boldsymbol{\mu_{ij}'}\cdot\Omega_{\boldsymbol{\mu j}}^{-1}\cdot\boldsymbol{\mu_{ij}}+\sum_{j=1}^{J}\left(\boldsymbol{\mu_j}-\boldsymbol{E_j}\right)'\cdot D_j^{-1}\cdot\left(\boldsymbol{\mu_j}-\boldsymbol{E_j}\right)-\sum_{j=1}^{J}\boldsymbol{E_j'}\cdot D_j^{-1}\cdot\boldsymbol{E_j}$$

where $D_j=\left(N_j\cdot\Omega_{\boldsymbol{\mu j}}^{-1}+\frac{1}{10^4}\right)^{-1}$ and $\boldsymbol{E_j}=D_j\cdot\Omega_{\boldsymbol{\mu j}}^{-1}\sum\limits_{\substack{j=1\\ i\in\{j\}}}^{J}\boldsymbol{\mu_{ij}}$.

Thus, the conditional posterior disstribution of $\boldsymbol{\mu_j}|\boldsymbol{\mu_{ij}},\Omega_{\boldsymbol{\mu j}},\boldsymbol{y},i\left[i\in\{j\}\right]=1,...,N$ is given by:

$$P\left(\boldsymbol{\mu_j}|\boldsymbol{\mu_{ij}},\Omega_{\boldsymbol{\mu j}},\boldsymbol{y},i\left[i\in\{j\}\right]=1,...,N\right)\propto\exp\left(-\frac{1}{2}\left(\boldsymbol{\mu_j}-\boldsymbol{E_j}\right)'\cdot D_j^{-1}\cdot\left(\boldsymbol{\mu_j}-\boldsymbol{E_j}\right)\right)\tag{36}$$

From Equation (36), the posterior distribution of $\boldsymbol{\mu_j}|\boldsymbol{\mu_{ij}},\Omega_{\boldsymbol{\mu j}},\boldsymbol{y},i\left[i\in\{j\}\right]=1,...,N$ is therefore as follows:

$$\boldsymbol{\mu_j}|\boldsymbol{\mu_{ij}},\Omega_{\boldsymbol{\mu j}}^{-1},\boldsymbol{y},i\left[i\in\{j\}\right]=1,...,N\sim N\left(\boldsymbol{E_j},D_j\right)\tag{37}$$

$$N\left(\left(N_j\cdot\Omega_{\boldsymbol{\mu j}}^{-1}+\frac{1}{10^4}\right)^{-1}\cdot\left[\Omega_{\boldsymbol{\mu j}}^{-1}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\boldsymbol{\mu_{ij}}\right],\left(N_j\cdot\Omega_{\boldsymbol{\mu j}}^{-1}+\frac{1}{10^4}\right)^{-1}\right)$$

$$P\left(\Omega_{\boldsymbol{\mu j}}^{-1}|\boldsymbol{\mu_{ij}},\boldsymbol{\mu_j},\boldsymbol{y},i\left[i\in\{j\}\right]=1,...,N\right)\tag{38}$$

$$\propto\left(\prod_{j=1}^{J}|\ \Omega_{\boldsymbol{\mu j}}\ |^{-\frac{1}{2}N_j}\right)\cdot\exp\left[-\frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\left(\boldsymbol{\mu_{ij}}-\boldsymbol{\mu_j}\right)'\cdot\Omega_{\boldsymbol{\mu j}}^{-1}\cdot\left(\boldsymbol{\mu_{ij}}-\boldsymbol{\mu_j}\right)\right]\cdot$$

$$\left(\prod_{j=1}^{J}|\ \Omega_{\boldsymbol{\mu j}}^{-1}\ |^{-\frac{1}{2}}\right)\cdot etr\left(-\sum_{j=1}^{J}R_j\cdot\Omega_{\boldsymbol{\mu j}}^{-1}\right)$$

$$\propto\left(\prod_{j=1}^{J}|\ \Omega_{\boldsymbol{\mu j}}\ |^{-\frac{1}{2}(N_j-1)}\right)\cdot etr\left[-\frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\left(\boldsymbol{\mu_{ij}}-\boldsymbol{\mu_j}\right)\cdot\left(\boldsymbol{\mu_{ij}}-\boldsymbol{\mu_j}\right)'\cdot\Omega_{\boldsymbol{\mu j}}^{-1}+2\sum_{j=1}^{J}R_j\cdot\Omega_{\boldsymbol{\mu j}}^{-1}\right]$$

From Equation (38), the posterior distribution of $\Omega_{\boldsymbol{\mu j}}^{-1}|\boldsymbol{\mu_{ij}}, \boldsymbol{y}, i\left[i \in \{j\}\right] = 1, ..., N$ is therefore as follows:

$$\Omega_{\boldsymbol{\mu j}}^{-1}|\boldsymbol{\mu_{ij}}, \boldsymbol{\mu_j}, \boldsymbol{y}, i\left[i \in \{j\}\right] = 1, ..., N \sim W_2\left(N_j + 2, \left[\sum_{\substack{j=1 \\ i \in \{j\}}}^{J} \left(\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_j}\right)\left(\boldsymbol{\mu_{ij}} - \boldsymbol{\mu_j}\right)' + 2 \cdot R_j\right]^{-1}\right) \tag{39}$$

$$P\left(\sigma_{\varepsilon j}^{-2}|\boldsymbol{\mu_{ij}}, \sigma_{\xi j}^{-2}, \boldsymbol{y}, i\left[i \in \{j\}\right] = 1, ..., N\right) \tag{40}$$

$$\propto \left(\prod_{j=1}^{J}\left(\sigma_{\xi j}^2 \cdot \sigma_{\varepsilon j}^2\right)^{-\frac{1}{2}T_j}\right) \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1 \\ i \in \{j\}}}^{J}\sum_{k=1}^{K_{ij}}\left[\frac{\left(y_{ijk} - \left(\alpha_{ij} - \lambda_{ij} \cdot t_{ijk}\right)\right)^2}{\sigma_{\xi j}^2 \cdot \sigma_{\varepsilon j}^2}\right]\right) \cdot$$

$$\left(\prod_{j=1}^{J}\left(\sigma_{\varepsilon j}^{-2} \cdot \sigma_{\xi j}^{-2}\right)^{\left(10^{-4}-1\right)}\right) \cdot \exp\left(-10^{-4}\sum_{j=1}^{J}\sigma_{\varepsilon j}^{-2} \cdot \sigma_{\xi j}^{-2}\right)$$

$$\propto \left(\prod_{j=1}^{J}\left(\sigma_{\xi j}^2 \cdot \sigma_{\varepsilon j}^2\right)^{-\frac{1}{2}T_j}\right) \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1 \\ i \in \{j\}}}^{J}\frac{1}{\sigma_{\xi j}^2 \cdot \sigma_{\varepsilon j}^2}\left(\boldsymbol{y_{ij}} - X_{ij} \cdot \boldsymbol{\mu_{ij}}\right)'\left(\boldsymbol{y_{ij}} - X_{ij} \cdot \boldsymbol{\mu_{ij}}\right)\right) \cdot$$

$$\left(\prod_{j=1}^{J}\left(\sigma_{\varepsilon j}^{-2} \cdot \sigma_{\xi j}^{-2}\right)^{\left(10^{-4}-1\right)}\right) \cdot \exp\left(-10^{-4}\sum_{j=1}^{J}\sigma_{\varepsilon j}^{-2} \cdot \sigma_{\xi j}^{-2}\right)$$

$$\propto \left(\prod_{j=1}^{J}\left(\sigma_{\xi j}^2 \cdot \sigma_{\varepsilon j}^2\right)^{-\frac{1}{2}T_j + 10^{-4} - 1}\right) \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^{N}\sigma_{\xi j}^{-2} \cdot \sigma_{\varepsilon j}^{-2}\left[\sum_{\substack{j=1 \\ i \in \{j\}}}^{J}\left(\boldsymbol{y_{ij}} - X_{ij} \cdot \boldsymbol{\mu_{ij}}\right)'\left(\boldsymbol{y_{ij}} - X_{ij} \cdot \boldsymbol{\mu_{ij}}\right) + 10^{-4}\right]\right)$$

From Equation (40), the posterior distribution of $\sigma_{\varepsilon j}^{-2}|\boldsymbol{\mu_{ij}}, \sigma_{\xi j}^{-2}, \boldsymbol{y}, i\left[i \in \{j\}\right] = 1, ..., N$ is therefore as follows:

$$\sigma_{\varepsilon j}^{-2}|\boldsymbol{\mu_{ij}}, \sigma_{\xi j}^{-2}, \boldsymbol{y}, i\left[i \in \{j\}\right] = 1, ..., N$$

$$\sim G\left(\frac{1}{2} \cdot T_j + 10^{-4}, \frac{1}{2}\sum_{\substack{j=1 \\ i \in \{j\}}}^{J}\sigma_{\xi j}^{-2}\left(\boldsymbol{y_{ij}} - X_{ij} \cdot \boldsymbol{\mu_{ij}}\right)'\left(\boldsymbol{y_{ij}} - X_{ij} \cdot \boldsymbol{\mu_{ij}}\right) + 10^{-4}\right) \tag{41}$$

$$P\left(\sigma_{\xi j}^{-2}|\boldsymbol{\mu_{ij}},\sigma_{\varepsilon j}^{-2},\boldsymbol{y},i\left[i\in\{j\}\right]=1,...,N\right) \tag{42}$$

$$\propto\left(\prod_{j=1}^{J}\left(\sigma_{\xi j}^{2}\cdot\sigma_{\varepsilon j}^{2}\right)^{-\frac{1}{2}T_j}\right)\cdot\exp\left(-\frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\sum_{k=1}^{K_{ij}}\left[\frac{\left(y_{ijk}-\left(\alpha_{ij}-\lambda_{ij}\cdot t_{ijk}\right)\right)^{2}}{\sigma_{\xi j}^{2}\cdot\sigma_{\varepsilon j}^{2}}\right]\right)\cdot$$

$$\left(\prod_{j=1}^{J}\left(\sigma_{\varepsilon j}^{-2}\cdot\sigma_{\xi j}^{-2}\right)^{\left(10^{-4}-1\right)}\right)\cdot\exp\left(-10^{-4}\sum_{j=1}^{J}\sigma_{\varepsilon j}^{-2}\cdot\sigma_{\xi j}^{-2}\right)$$

$$\propto\left(\prod_{j=1}^{J}\left(\sigma_{\xi j}^{2}\cdot\sigma_{\varepsilon j}^{2}\right)^{-\frac{1}{2}T_j}\right)\cdot$$

$$\exp\left(-\frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\frac{1}{\sigma_{\xi j}^{2}\cdot\sigma_{\varepsilon j}^{2}}\left(\boldsymbol{y_{ij}}-X_{ij}\cdot\boldsymbol{\mu_{ij}}\right)'\left(\boldsymbol{y_{ij}}-X_{ij}\cdot\boldsymbol{\mu_{ij}}\right)\right)$$

$$\left(\prod_{j=1}^{J}\left(\sigma_{\varepsilon j}^{-2}\cdot\sigma_{\xi j}^{-2}\right)^{\left(10^{-4}-1\right)}\right)\cdot\exp\left(-10^{-4}\sum_{j=1}^{J}\sigma_{\varepsilon j}^{-2}\cdot\sigma_{\xi j}^{-2}\right)$$

$$\propto\left(\prod_{j=1}^{J}\left(\sigma_{\xi j}^{2}\cdot\sigma_{\varepsilon j}^{2}\right)^{-\frac{1}{2}T_j+10^{-4}-1}\right)\cdot\exp\left(-\frac{1}{2}\sum_{i=1}^{N}\sigma_{\xi j}^{-2}\cdot\sigma_{\varepsilon j}^{-2}\left[\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\left(\boldsymbol{y_{ij}}-X_{ij}\cdot\boldsymbol{\mu_{ij}}\right)'\left(\boldsymbol{y_{ij}}-X_{ij}\cdot\boldsymbol{\mu_{ij}}\right)+10^{-4}\right]\right)$$

From Equation (42), the posterior distribution of $\sigma_{\xi j}^{-2}|\boldsymbol{\mu_{ij}},\sigma_{\varepsilon j}^{-2},\boldsymbol{y},i\left[i\in\{j\}\right]=1,...,N$ is therefore as follows:

$$\sigma_{\xi j}^{-2}|\boldsymbol{\mu_{ij}},\sigma_{\varepsilon j}^{-2},\boldsymbol{y},i\left[i\in\{j\}\right]=1,...,N$$

$$\sim G\left(\frac{1}{2}\cdot T_j+10^{-4},\frac{1}{2}\sum_{\substack{j=1\\ i\in\{j\}}}^{J}\sigma_{\varepsilon j}^{-2}\left(\boldsymbol{y_{ij}}-X_{ij}\cdot\boldsymbol{\mu_{ij}}\right)'\left(\boldsymbol{y_{ij}}-X_{ij}\cdot\boldsymbol{\mu_{ij}}\right)+10^{-4}\right) \tag{43}$$

An analysis of multilevel models in understanding the science performance of South African learners using the TIMSS 2015 data

Sphiwe Skhosana 14308747

STK795 Research Report

Submitted in partial fulfillment of the degree BCom (Hons) Statistics

Supervisor: Dr. G. Crafford

Department of Statistics, University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

October 30, 2017

Abstract

In South Africa, students have been found to differ in their knowledge of science [21]. Using the TIMSS (Trends In International Mathematics and Science study) 2015 data, in this paper a multilevel model is fitted for the two-level case involving students nested in schools in the investigation of factors associated with the science performance of South African 9th-graders. Multilevel analysis shows that school factors contribute more to the differences. About 54% of the variation in science achievement occurs between schools.

Declaration

I, *Sphiwe Bonakele Skhosana,* declare that this essay, submitted in partial fulfillment of the degree *BCom (Hons) Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____

Sphiwe Bonakele Skhosana

_____

Gretel Crafford

_____

Date

Acknowledgements

The completion of this essay was impossible without the support of the following entities, they are therefore acknowledged:

- I am grateful to God for providing me with life and the spirit of patience and steadfastness.
- I thank my supervisor Dr. Gretel Crafford for her valuable input towards the completion of this essay.
- I would like to extend my humblest gratitude to the Center for AI Research, Meraka Institute, CSIR and STATOMET for providing financial support in order for me to further my studies.
- Lots of love to my family, for providing moral support in difficult times.
- Thanks to the Statistics Department, University of Pretoria for giving me an opportunity to acquire further knowledge is statistics.

Contents

List of Figures

List of Tables

1 Introduction

According to the human science research council (HSRC), South Africa is among the bottom five countries in their performance in science [21]. Science achievement is a key indicator in assessing a country's schooling system [22]. A country's economy is strongly associated with scientific literacy [9], this might be important for developing countries such as South Africa. For this reason, the quality of the teaching and learning of of science has to be monitored. Towards this end, factors associated with science performance must be identified. This is achieved by making use of the 2015 data from the Trends in international mathematics and science study (TIMSS) at the 9^{th} grade level.

Because students are taught in schools they might share the same class, be taught by the same teacher and even come from the same neighborhood as such, their academic achievements are not independent. To account for these dependencies a two-level multilevel analysis is conducted with students as level-1 units and schools as level-2 units [19].

Nationally and internationally, two-level multilevel studies have been conducted with the aim of identifying factors associated with science achievement. Nationally: at the student-level, Cho et al. [1], using the TIMSS 2003 data at the secondary school level, student's socio-economic status (SES) and attitude towards science were among the most significant factors associated with science achievement in South Africa. At the school-level, Frempong et al. [3] proved that, in South Africa, the school's SES is over and above that of the individual student. Internationally: at the student-level, using the TIMSS 2007 data, Mohammadpour[15] identified the student's value of science and the time spent working at home as being strongly related to science achievement in Malaysia. At the school-level, Mohammadpour [15] found that school location and teacher emphasis on educational success highly influenced science achievement in Malaysia. The analysis can be extended to more than two levels, in a three-level analysis, Mohammadpour [16], using the TIMSS 2007 data, investigated the factors associated with science achievement as a function of student-, classroom-, and school-level factors in Singapore. In this extended analysis, teaching limitations was found to be highly associated with science achievement at the classroom-level.

This paper will make use of Raudenbush and Bryk [18] for the theoretical aspects and Singer [23] for the practical specification of multilevel modeling in the investigation of factors affecting science performance in South Africa.

2 Background Theory

Hierarchical data structures, in which level-1 units are nested within level-2 units, are a commonly encountered phenomenon [17]. For example, in educational research students' performance is assessed at the classroom, school, district and provincial level. Similarly in developmental studies, a hierarchical

structure occurs when multiple data is collected repeatedly on an individual person because the repeated measures are nested within an individual [20, 12]. These data structures cannot be analyzed using ordinary regression techniques because these techniques pool or aggregate the data for estimation purposes and thus ignores the group effects which might lead to wrong conclusions. The growing popularity of hierarchical data structures in the social science saw an evolution in the development of methods that take into account the structure of the data. The first attempt occurred in the early 1980s with the development of an algorithm for covariance component estimation [27]. This development led to what is now, in the statistical literature, known as covariance components modeling [20] or simply multilevel modeling. A multilevel modeling technique is a complex form of ordinary least squares regression [27], instead of fitting a single-level model as with ordinary regression, multiple models at different levels within the structure are fitted. The fitted models express the relationships within a given level and specify how an occurrence at one level influence the variation occurring at other levels [20]. These methods are effective in modeling data with a natural hierarchy (clustered or nested units of a given lower level in another higher level's units). The following subsequent sections provide the building blocks of multilevel modeling by describing the theoretical consideration of multilevel models focusing on three important modeling aspects: model specification, estimation and diagnostics. To explain the fundamental statistical features the discussion here is constrained to the basic two-level model. The standard form of presenting statistical models is employed to introduce key concepts and then the paper reverts to the official notation using matrices.

2.1 Theory of model specification

2.1.1 The basic two-level model

Suppose that data is collected from N level-2 units each of the units containing n_j (for $j = 1, 2, ..., N$) level-1 units, the data is thus hierarchical because the level-1 units are clustered or nested within the level-2 units.

Let

y_{ij} = the response variable of the i^{th} level-1 unit from the j^{th} level-2 unit.

x_{ij} =the predictor (explanatory) variable of the i^{th} level-1 unit from the j^{th} level-2 unit.

z_j =the predictor of the j^{th} level-2 unit.

The resulting Level-1 model can be written as follows

$$y_{ij} = b_{0j} + b_{1j}x_{ij} + r_{ij}, \quad r_{ij} \sim N(0, \sigma^2) \tag{1}$$

$$\text{for j} = 1, 2, ..., N \text{ and i} = 1, 2, ..., n_j$$

The random error term, r_{ij}, associated with the i^{th} level-1 unit from the j^{th} level-2 unit, is normally

distributed with mean zero and constant variance, σ^2. The subscripts on the intercept, b_{0j}, and slope, b_{1j}, implies that each level-2 unit will have its own unique set of coefficients. This implies that the level-2 model's response variables (level-1 coefficients) will be a linear combination of the means, γ_{00} or γ_{01} (see Equation 2), and the behavior of the level-2 predictor, z_j, and the random error term for the effect of the j^{th} level-2 unit. The level-2 model is thus specified as follows

$$\begin{aligned} b_{0j} =& \gamma_{00} + \gamma_{01} z_j + u_{0j} \\ b_{1j} =& \gamma_{10} + \gamma_{11} z_j + u_{1j} \\ & \text{for j} = 1, 2, ..., N \\ E \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ and } var \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = \begin{bmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{bmatrix} \end{aligned} \tag{2}$$

Assuming that u_{0j} and u_{1j} have a bivariate normal distribution with mean zero and variance τ_{00} and τ_{11}, respectively, with covariance τ_{01}. These variance-covariance components measure the variability in b_{0j} and b_{1j} not explained by z_j. To get the multilevel model, (2) is substituted into (1) leading to

$$\begin{aligned} y_{ij} =& [\gamma_{00} + \gamma_{01} z_j + \gamma_{10} x_{ij} + \gamma_{11} z_j x_{ij}] + [u_{0j} + u_{1j} x_{ij} + r_{ij}] \\ \text{for j} =& 1, 2, ..., N \text{ and i} = 1, 2, ..., n_j \end{aligned} \tag{3}$$

Equation (3) has two components: the fixed part (the first bracket) and the random component (the second term bracket). Since the model coefficients in (3) are the effects from both levels of the hierarchy, they might jointly be referred to as multilevel coefficients. Because the error structure, the random part, includes both the within- and between-group error terms, the estimation of the fixed effects will require iterative procedures. In later sections it will be demonstrated how to estimate the multilevel parameters and the random effects. In general, models in (1) and (2) can be expressed as follows

Matrix notation

Level-1 Model

$$\mathbf{Y}_j = \mathbf{X}_j \boldsymbol{\beta}_j + \mathbf{r}_j, \quad \mathbf{r}_j \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_{n_j}) \tag{4}$$

$$= \begin{pmatrix} \mathbf{1} & \mathbf{x}_{1j} & \cdots & \mathbf{x}_{Qj} \end{pmatrix} \begin{pmatrix} \beta_{0j} \\ \beta_{1j} \\ \vdots \\ \beta_{Qj} \end{pmatrix} + \mathbf{r}_j \tag{5}$$

Where Q is the number of level-1 predictors.$\mathbf{Y}_j : \ (n_j \times 1)$ is a vector of outcomes for the j$^{\text{th}}$ level-2 unit, $\mathbf{X}_j : \ n_j \times (Q+1)$ is a matrix of predictors or explanatory variables including the intercept, $\boldsymbol{\beta}_j : (Q \times 1)$ is a vector of level-1 regression coefficients and $\mathbf{r}_j : (n_j \times 1)$ vector of random terms for the j$^{\text{th}}$ level-2 unit . The error-vector $\mathbf{r}_j$ is normally distributed with mean $\mathbf{0}$ and a variance-covariance matrix $\sigma^2\mathbf{I}_{n_j}$, with constant variability for each level-1 unit from the j$^{\text{th}}$ level-2 unit, where $\mathbf{I}_{n_j} : (n_j \times n_j)$ is an identity matrix. The variance σ^2 is also known as the within-group variance.

Level-2 Model

$$\boldsymbol{\beta}_j = \begin{pmatrix} \beta_{0j} \\ \beta_{1j} \\ \vdots \\ \beta_{Qj} \end{pmatrix} = \begin{pmatrix} \gamma_{00} + \gamma_{01}z_{1j} + \cdots + \gamma_{0P}z_{Pj} + u_{0j} \\ \gamma_{10} + \gamma_{11}z_{1j} + \cdots + \gamma_{1P}z_{Pj} + u_{1j} \\ \ddots \\ \gamma_{Q0} + \gamma_{Q1}z_{1j} + \cdots + \gamma_{QP}z_{Pj} + u_{Qj} \end{pmatrix} \tag{6}$$

$$= \mathbf{I}_{(Q+1)\times(Q+1)} \otimes \begin{pmatrix} 1 & z_{1j} & \cdots & z_{Pj} \end{pmatrix} \boldsymbol{\gamma} + \begin{pmatrix} u_{0j} \\ u_{1j} \\ \vdots \\ u_{Qj} \end{pmatrix} \tag{7}$$

$$= \mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{u}_j, \qquad \mathbf{u}_j \sim N(\mathbf{0}, \boldsymbol{\Phi}) \tag{8}$$

$$\text{where } \boldsymbol{\Phi} = \begin{pmatrix} \tau_{00} & \tau_{01} & \cdots & \tau_{0P} \\ \tau_{10} & \tau_{11} & \cdots & \tau_{1P} \\ & \vdots & \ddots & \vdots \\ \tau_{Q0} & \tau_{Q1} & \cdots & \tau_{QP} \end{pmatrix} \text{ and } \mathbf{Z}_j = \mathbf{I}_{(Q+1)\times(Q+1)} \otimes \begin{pmatrix} 1 & z_{1j} & \cdots & z_{Pj} \end{pmatrix}$$

Where P is the number of level-2 predictors. From (8), $\mathbf{Z}_j : (Q+1) \times (P+1)$ is a block diagonal matrix of the stacked $Q+1$ row vectors of the level-2 predictors including the overall effect represented by 1 (see (7)), $\boldsymbol{\gamma} : (QP \times 1)$ is a vector of fixed effects and$\mathbf{u}_j : (Q+1) \times 1$ is a vector of random effects. $\mathbf{u}_j$ is a multivariate normally distributed variable with mean $\mathbf{0}$ and variance-covariance matrix $\boldsymbol{\Phi}$. $\boldsymbol{\Phi}$is a square matrix. The elements on the diagonal τ_{00}, $\tau_{11} \cdots \tau_{QP}$ measure the variability between level-2

intercept coefficients, β_{0j}, and the partial slopes, β_{1j},β_{2j}, $\cdots$,β_{Qj} ,respectively. The elements off-diagonal elements are covariance measures between the regression coefficients in the vector $\boldsymbol{\beta}_j$.

Combined or multilevel model

Substituting (8) into (4) will yield the combined model

$$\begin{aligned}\mathbf{Y}_j =&\mathbf{X}_j(\mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{u}_j) + \mathbf{r}_j \\ =&\mathbf{X}_j\mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{X}_j\mathbf{u}_j + \mathbf{r}_j \\ =&\mathbf{X}_j^*\boldsymbol{\gamma} + \mathbf{X}_j\mathbf{u}_j + \mathbf{r}_j \end{aligned} \tag{9}$$

In (9), $\mathbf{X}_j^* : n_j \times (P+1)$ is a matrix of predictors from both levels including their interactions and the intercept, it can be referred to as a matrix of multilevel variables.. The equation can be split into two parts, the fixed part $\mathbf{X}_j^*\boldsymbol{\gamma}$, and the random part $\mathbf{X}_j\mathbf{u}_j + \mathbf{r}_j$.

2.1.2 Simpler sub-models

The discussion that follows focuses on how to specify a variety of models with a hierarchical data structure. These models are derived from the basic two-level model and they can be used to answer a variety of questions concerning occurrences at various levels within the hierarchy.

One way ANOVA with random effects

The first and most important hierarchical linear model in that it gives preliminary auxiliary information as to how the outcome variability is apportioned between the two levels.

By setting the variables $\mathbf{x}_{1j}$, $\mathbf{x}_{2j}$, $\cdots$, $\mathbf{x}_{Qj}$ and the coefficients β_{1j}, β_{2j}, $\cdots$, β_{Qj} in (5) to zero, the level-1 model is specified as follows

$$\mathbf{Y}_j =\beta_{0j}\mathbf{1} + \mathbf{r}_j, \qquad \mathbf{r}_j \sim N(\mathbf{0}, \sigma^2\mathbf{I}_{n_j}) \tag{10}$$

The level-1 random-effect, $\mathbf{r}_j$, is the effect of the randomness occurring within the j^{th} level-2 unit. It is assumed to be normally distributed with mean $\mathbf{0}$ and variance-covariance matrix $\sigma^2\mathbf{I}_{n_j}$ with constant variance σ^2 on the diagonal and $cov(\mathbf{r}_j, \mathbf{r}_k) = 0$, for $k \neq j$ as the off diagonal elements.

With β_{1j}, β_{2j} and z_j in (6) set to zero, the level-2 model for the one-way ANOVA with random effects is given as

$$\beta_{0j} = \gamma_{00} + u_{0j}, \qquad u_{0j} \sim N(0, \tau_{00}) \tag{11}$$

In (11), u_{0j} is the effect of the randomness (random effect) between the level-2 units and it is assumed to be normally distributed with mean 0 and variance τ_{00}.

Substitution of (11) into (10) yields the combined model

$$\begin{aligned} \mathbf{Y}_j &= (\gamma_{00} + u_{0j})\mathbf{1} + \mathbf{r}_j \\ &= \gamma_{00}\mathbf{1} + u_{0j}\mathbf{1} + \mathbf{r}_j \end{aligned} \tag{12}$$

The model in (12) is the model that is estimated under the one way ANOVA with random effects. In the following section it is shown how a multilevel model is estimated. For now a variety of specifications are considered and their purpose is explained. For example, for the one way ANOVA with random effects it was mentioned above that the model gives a preliminary auxiliary information. It is determined by taking the ratio of the outcome variability between level-2 units, τ_{00}, and the total outcome variability, $\tau_{00} + \sigma^2$, given in (13). This statistic is referred to as the intra-class correlation (ICC) denoted by ρ, and it gives the proportion of the variability in the outcome variable that occurs between level-2 units.

$$\rho = \frac{\tau_{00}}{\tau_{00} + \sigma^2} \tag{13}$$

One-way ANCOVA model with random effects

This model is considered an alternative to the multilevel model whenever there is only one level-1 predictor. From (5), by setting the variables $\mathbf{x}_{2j}, \mathbf{x}_{3j}, \cdots, \mathbf{x}_{Qj}$ and the coefficients $\beta_{2j}, \beta_{3j}, \cdots, \beta_{Qj}$ equal to zero, the level-1 model is obtained. The level-2 model is specified as given by

$$\boldsymbol{\beta}_j = \begin{pmatrix} \beta_{0j} \\ \beta_{1j} \end{pmatrix} = \begin{pmatrix} \gamma_{00} + u_{0j} \\ \gamma_{10} + 0 \end{pmatrix} = \boldsymbol{\gamma} + \mathbf{u}_j \tag{14}$$

The combined model is given in (15), it is obtained by substituting the level-2 model into the level-1 model.

$$\begin{aligned}\mathbf{Y}_j =& \begin{pmatrix} \mathbf{1} & \mathbf{x}_j \end{pmatrix} \boldsymbol{\gamma} + \begin{pmatrix} \mathbf{1} & \mathbf{x}_j \end{pmatrix} u_{0j} + \mathbf{r}_j \\ =& \mathbf{X}_j \boldsymbol{\gamma} + u_{0j}\mathbf{1} + \mathbf{r}_j \end{aligned} \tag{15}$$

The ANCOVA model can be extended to include level-2 predictors and even more than one level-1 predictor in which case it will be a multiple-way ANCOVA model (or simply MANCOVA).

Means as outcomes regression

A frequently encountered analysis problem is to predict the outcome of level-2 units given the factors at that level. This model answers the question of how influential the group factors are on the outcome.

The level-1 model is the same as the one in (10). The level-2 model is obtained by setting the coefficients $\beta_{1j}, \beta_{2j}, \cdots, \beta_{Qj}$ in (6) equal to zero:

$$\begin{aligned}\beta_{0j} =& \gamma_{00} + \gamma_{01} z_j + u_{0j} \\ =& \begin{pmatrix} 1 & z_j \end{pmatrix} \begin{pmatrix} \gamma_{00} \\ \gamma_{01} \end{pmatrix} + u_{0j} \\ =& \mathbf{Z}_j \boldsymbol{\gamma} + u_{0j} \\ \text{where } u_{0j} \sim& N(0, \tau_{00}) \end{aligned} \tag{16}$$

Substituting (16) into (10) yields the combined model:

$$\begin{aligned}\mathbf{Y}_j =& (\mathbf{Z}_j \boldsymbol{\gamma} + u_{0j})\mathbf{1} + \mathbf{r}_j \\ =& \mathbf{Z}_j \boldsymbol{\gamma} + u_{0j}\mathbf{1} + \mathbf{r}_j \end{aligned} \tag{17}$$

The random effects in (17) have the same distributional implications as those in (12).

Random coefficients model

Until now, the models considered had varying intercepts that were allowed to vary across level-2 units, in this model both regression coefficients are allowed to vary across level-2 units. This specification helps to aid in an answer to the question of whether strong relationships between level-1 factors and the outcome have a positive or negative effect on the outcome of level-2 units. The level-1 model is specified by setting

variables $\mathbf{x}_{2j}$, $\mathbf{x}_{3j}$, $\cdots$, $\mathbf{x}_{Qj}$ and the coefficients β_{2j}, β_{3j}, $\cdots$, β_{Qj} in (5) equal to zero:

$$\begin{aligned}\mathbf{Y}_j =& \mathbf{X}_j\boldsymbol{\beta}_j + \mathbf{r}_j, \quad \mathbf{r}_j \sim N(\mathbf{0}, \sigma^2\mathbf{I}_{n_j}) \\ =& \begin{pmatrix} \mathbf{1} & \mathbf{x}_{1j} \end{pmatrix} \begin{pmatrix} \beta_{0j} \\ \beta_{1j} \end{pmatrix} + \mathbf{r}_j \end{aligned} \tag{18}$$

The random effect $\mathbf{r}_j$ in (18) has the same distributional implications as the one in (4).

The level-2 model is obtained by setting β_{2j} in (6) equal to zero and excluding the level-2 predictor z_j:

$$\begin{aligned}\boldsymbol{\beta}_j =& \begin{pmatrix} \beta_{0j} \\ \beta_{1j} \end{pmatrix} \\ =& \begin{pmatrix} \gamma_{00} + u_{0j} \\ \gamma_{10} + u_{1j} \end{pmatrix} \\ =& \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \gamma_{00} \\ \gamma_{10} \end{pmatrix} + \begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \\ =& \mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{u}_j, \qquad \mathbf{u}_j \sim BVN(\mathbf{0}, \boldsymbol{\Phi}) \end{aligned} \tag{19}$$

$$\text{where } \boldsymbol{\Phi} = \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix}$$

Where BVN is bivariate normal distribution. The effect of the randomness, $\mathbf{u}_j$, between the level-2 units is assumed to be bivariate normally distributed with mean $\mathbf{0} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and a symmetric variance-covariance matrix $\boldsymbol{\Phi}$ with elements τ_{00} as an unconditional measure of the variability between level-1 intercepts, β_{0j}, for $j = 1, 2, ..., N$; τ_{11} as a measure of the variability between the level-1 slopes, β_{1j}, for $j = 1, 2, ..., N$; and τ_{01} or τ_{10} is a measure of the unconditional covariance between the level-1 intercepts and slopes. The τ's described above are referred to as unconditional because the level-2 model in (19) has no predictors.

Substituting (19) into (18) yields the combined model:

$$\mathbf{Y}_j = \mathbf{X}_j(\mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{u}_j) + \mathbf{r}_j$$

$$
\begin{aligned}
&= \mathbf{X}_j \mathbf{Z}_j \boldsymbol{\gamma} + \mathbf{X}_j \mathbf{u}_j + \mathbf{r}_j \\
&= \mathbf{X}_j \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \boldsymbol{\gamma} + \begin{pmatrix} \mathbf{1} & \mathbf{x}_j \end{pmatrix} \begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} + \mathbf{r}_j \\
&= \mathbf{X}_j \boldsymbol{\gamma} + u_{0j}\mathbf{1} + u_{1j}\mathbf{x}_j + \mathbf{r}_j
\end{aligned} \tag{20}
$$

This model has two useful variants: one is an extension and the other is a limitation. The first sub-sub model models the variability in the level-1 coefficients by controlling for a level-2 predictor, from (20) it follows that the level-2 model is specified as

$$
\boldsymbol{\beta}_j = \begin{pmatrix} \gamma_{00} + \gamma_{01} z_j + u_{0j} \\ \gamma_{10} + \gamma_{11} z_j + u_{1j} \end{pmatrix} = \begin{pmatrix} 1 & z_j & 0 & 0 \\ 0 & 0 & 1 & z_j \end{pmatrix} \boldsymbol{\gamma} + \mathbf{u}_j = \mathbf{Z}_j^* \boldsymbol{\gamma} + \mathbf{u}_j \tag{21}
$$

The combined model is given by

$$
\begin{aligned}
\mathbf{Y}_j &= \mathbf{X}_j \begin{pmatrix} 1 & z_j & 0 & 0 \\ 0 & 0 & 1 & z_j \end{pmatrix} \boldsymbol{\gamma} + \begin{pmatrix} \mathbf{1} & \mathbf{x}_j \end{pmatrix} \begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} + \mathbf{r}_j \\
&= \mathbf{X}_j \boldsymbol{\gamma} + z_j \boldsymbol{\gamma} + z_j \mathbf{x}_j \boldsymbol{\gamma} + u_{0j}\mathbf{1} + u_{1j}\mathbf{x}_j + \mathbf{r}_j
\end{aligned} \tag{22}
$$

The variable z_j is the level-2 predictor and u_{0j} and u_{1j} are as those specified in (19). The model specified in this manner is referred to as the *random intercept and slopes as outcomes* model.

The second sub-sub model controls for a level-2 predictor but only allows the intercept to randomly vary across level-2 units, whilst the variability in the group slopes is not random but purely due to the differences in the value of the level-2 predictor of each group, the level-2 model follows from (21) by setting u_{1j} equal to zero

$$
\boldsymbol{\beta}_j = \begin{pmatrix} \gamma_{00} + \gamma_{01} z_j + u_{0j} \\ \gamma_{10} + \gamma_{11} z_j + 0 \end{pmatrix} = \begin{pmatrix} 1 & z_j & 0 & 0 \\ 0 & 0 & 1 & z_j \end{pmatrix} \boldsymbol{\gamma} + \begin{pmatrix} u_{0j} \\ 0 \end{pmatrix} = \mathbf{Z}_j^* \boldsymbol{\gamma} + \mathbf{u}_j \tag{23}
$$

The combined model is given by (24). This model is referred to as a *non-randomly varying slope*

model.

$$\begin{aligned}\mathbf{Y}_j =& \mathbf{X}_j \begin{pmatrix} 1 & z_j & 0 & 0 \\ 0 & 0 & 1 & z_j \end{pmatrix} \boldsymbol{\gamma} + \begin{pmatrix} \mathbf{1} & \mathbf{x}_j \end{pmatrix} \begin{pmatrix} u_{0j} \\ 0 \end{pmatrix} + \mathbf{r}_j \\ =& \mathbf{X}_j\boldsymbol{\gamma} + z_j\boldsymbol{\gamma} + z_j\mathbf{x}_j\boldsymbol{\gamma} + u_{0j}\mathbf{1} + \mathbf{r}_j \end{aligned} \tag{24}$$

2.1.3 Centering

In quantitative research, the interpretation of regression coefficients is important as their meaning is often used for decision making. In a two-level analysis, the coefficients coefficients of the level-1 model become outcomes variable (s) for the level-2 model as in (2). The regression coefficient β_{0j} is interpreted as the expected outcome for a level-1 unit in the j^{th} level-2 unit with a value of x_{ij} equal to zero, but a value of zero might not be within the domain of x_{ij} as result β_{0j} is meaningless. In order to render the regression coefficients interpretable with valid meaning, the location of the level-1 predictors must be altered to reach a meaningful interpretation. Similarly, for the interpretation of the fixed effects (i.e γ_{00} and γ_{10}), the location of z_j must be changed.

The location of a predictor can be altered by either centering the predictor about its group mean or grand mean. The former produces a centered variable of the form $(x_{ij} - \bar{x}_{\bullet j})$, where $\bar{x}_{\bullet j}$ is the group mean, the intercept is interpreted as the expected outcome for a level-1 unit in the j^{th} level-2 unit with a value of x_{ij} equal to the group mean. The latter produces a variable of the form $(x_{ij} - \bar{x}_{\bullet\bullet})$, where $\bar{x}_{\bullet\bullet}$ is the grand mean, the intercept is now interpreted as the expected outcome for a level-1 unit in the j^{th} level-2 unit with a value of x_{ij} equal to the grand mean. The method of centering has been found to improve the parameter estimation performance of iterative procedures [13]. Another option is to take the predictor in its natural form, in which case the domain of x_{ij} contains a zero.

As with classical linear regression, multilevel linear modeling requires some distributional assumptions to hold ([2]):

1. $\mathbf{X}_j$ and $\boldsymbol{\beta}$ are non-random, i.e they are not randomly generated
2. $\mathbf{r}_j$ has expected value zero and variance-covariance matrix $\sigma^2\mathbf{I}_{n_j}$
3. $\mathbf{u}_j$ has mean zero and variance-covariance matrix $\boldsymbol{\Phi}$, and
4. $\mathrm{Cov}(\mathbf{r}_j, \mathbf{u}'_j)$=$\mathbf{0}$.

2.2 Theory of model estimation

Two level hierarchical modeling involves the estimation of three types of parameters: the fixed effects (e.g γ_{00}), random level-1 coefficients (e.g β_{0j}), and the variance-covariance components (e.g τ_{00}). The estimation of each requires knowledge of others. In the following discussion, the theoretical considerations underlying estimation of each, in the order provided above, are given.

2.2.1 Fixed effects

These parameters do not vary across level-2 units, recall that the level-1 model in (4) was given by

$$\begin{aligned}\mathbf{Y}_j =&\mathbf{X}_j\boldsymbol{\beta}_j + \mathbf{r}_j\\ \text{where } \mathbf{r}_j &\sim N(\mathbf{0}, \sigma^2\mathbf{I}_{n_j})\end{aligned}$$

Assuming that $\mathbf{X}_j$ is of full column rank, $(k \times 1)$, the ordinary least squares (OLS) estimator of $\boldsymbol{\beta}_j$ is

$$\hat{\boldsymbol{\beta}}_j = (\mathbf{X}_j'\mathbf{X}_j)^{-1}\mathbf{X}_j\mathbf{Y}_j$$

the variance of this estimator, also known as the dispersion matrix, is given by

$$\begin{aligned}var(\hat{\boldsymbol{\beta}}_j) &= \sigma^2(\mathbf{X}_j'\mathbf{X}_j)^{-1}\\ &= \mathbf{V}_j\end{aligned} \tag{25}$$

The variance in (25) can be referred to as the error-variance matrix, it can be seen by premultiplying (4) by $(\mathbf{X}_j'\mathbf{X}_j)^{-1}\mathbf{X}_j'$, the following is obtained for $\hat{\boldsymbol{\beta}}_j$:

$$\begin{aligned}\hat{\boldsymbol{\beta}}_j =&\boldsymbol{\beta}_j + \mathbf{e}_j\\ \text{where } \mathbf{e}_j &\sim N(\mathbf{0}, \sigma^2\mathbf{I}_j)\end{aligned} \tag{26}$$

Recall from (8) that the level-2 model was specified as

$$\begin{aligned}\boldsymbol{\beta}_j =&\mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{u}_j\\ \text{where } \mathbf{u}_j &\sim N(\mathbf{0}, \boldsymbol{\Phi})\end{aligned} \tag{27}$$

And when (27) is substituted into (26) the following results:

$$\hat{\boldsymbol{\beta}}_j = \mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{u}_j + \mathbf{e}_j \tag{28}$$

The dispersion matrix given in (25) is redefined to account for the effect of $\mathbf{W}_j$, the block-diagonal matrix with level- predictors, in (28) :

$$\begin{aligned} var(\hat{\boldsymbol{\beta}}_j) =& var(\mathbf{u}_j + \mathbf{e}_j) \\ =& var(\mathbf{u}_j) + var(\mathbf{e}_j) \\ =& \boldsymbol{\Phi} + \sigma^2\mathbf{I}_{n_j}, \qquad Cov(\mathbf{u}_j, \mathbf{e}_j) = \mathbf{0} \\ =& \boldsymbol{\Delta}_j \end{aligned} \tag{29}$$

In Equation (29), $\boldsymbol{\Delta}_j$ is the total dispersion and it is partitioned into a sum of ,$\boldsymbol{\Phi}$, the parameter dispersion (i.e the variance-covariance matrix for the variability between level-2 units) and, $\sigma^2\mathbf{I}_{n_j}$, the error dispersion (i.e the variance-covariance matrix for the variability within each level-2 unit between the level-1 units), the former is constant across level-2 units whereas the latter varies depending on the level-2 group sample sizes, the subscript n_j. If the number of observations were the same between level-2 units then (29) would be written as

$$\begin{aligned} var(\hat{\boldsymbol{\beta}}_j) =& \boldsymbol{\Phi} + \sigma^2\mathbf{I} \\ =& \boldsymbol{\Delta} \end{aligned} \tag{30}$$

Equation (30) shows that the total dispersion is constant across level-2 units. In this instance the OLS regression estimate for $\boldsymbol{\gamma}$ would have been

$$\hat{\boldsymbol{\gamma}}_{OLS} = (\Sigma_{j=1}^{N}\mathbf{W}_j'\mathbf{W}_j)^{-1}\Sigma_{j=1}^{N}\mathbf{W}_j'\hat{\boldsymbol{\beta}}_j$$

But for different group (level-2 units) sample sizes with the total dispersion matrix, $\boldsymbol{\Delta}_j$, varying between level-2 units, the unique, minimum-variance and unbiased estimator of $\boldsymbol{\gamma}$ will be the generalized least squares (GLS) estimator:

$$\hat{\boldsymbol{\gamma}}_{GLS} = (\Sigma_{j=1}^{N}\mathbf{W}_j\boldsymbol{\Delta}_j^{-1}\mathbf{W}_j)^{-1}\Sigma_{j=1}^{N}\mathbf{W}_j'\boldsymbol{\Delta}_j^{-1}\hat{\boldsymbol{\beta}}_j \tag{31}$$

The inverse of the total dispersion matrix, $\boldsymbol{\Delta}_j^{-1}$, is called the precision matrix. In Equation (31) the weights are represented by each level-2 unit's precision matrix. Note from the discussion above that in estimating the fixed effects, it is assumed that the random level-1 coefficients, $\hat{\boldsymbol{\beta}}_j$, are known. In practice however these are not known prior to the analysis and in such situations to obtain efficient estimates for the fixed effects iterative estimation procedures are employed (see subsection 2.2.3).

2.2.2 Random level-1 coefficients

Multilevel modeling allows a variety of approaches to obtain the level-1 coefficients, one approach is to determine the best estimator for a set of regression coefficients for each level-2 unit by combining two potential estimators of $\boldsymbol{\beta}_j$ and computing an optimal weighted estimator, referred to as an empirical Bayes estimator (EB).

The two estimators are the OLS estimator $\hat{\boldsymbol{\beta}}_j^{OLS} = (\mathbf{X}_j'\mathbf{X}_j)^{-1}\mathbf{X}_j'\mathbf{Y}_j$ and borrowing information from an estimate based on the level-2 fixed effects, $\hat{\hat{\boldsymbol{\beta}}}_j = \mathbf{W}_j\hat{\boldsymbol{\gamma}}$. The weights are given by the reliability of the OLS estimate, that is the ratio between the parameter dispersion and the total dispersion ($\boldsymbol{\Phi} + \sigma^2\mathbf{I}_{n_j}$ or $\boldsymbol{\Delta}_j$).

Mathematically,

$$\boldsymbol{\Lambda}_j = \boldsymbol{\Phi}(\boldsymbol{\Phi} + \sigma^2\mathbf{I}_{n_j})^{-1}$$

where $\boldsymbol{\Delta}_j$ is the reliability matrix having, for each Level-2 unit, the reliability of the OLS estimator associated with that group. Therefore the optimal or EB estimator is given by

$$\boldsymbol{\beta}_j^* = \boldsymbol{\Lambda}_j\hat{\boldsymbol{\beta}}_j^{OLS} + (\mathbf{I} - \boldsymbol{\Lambda}_j)\hat{\hat{\boldsymbol{\beta}}}_j \tag{32}$$

In Equation (32), $\boldsymbol{\beta}_j^*$ is referred to as an EB estimator. Notice that if $\hat{\boldsymbol{\beta}}_j$ is a less reliable estimate of $\boldsymbol{\beta}_j$, that is for small $\boldsymbol{\Lambda}_j$, the EB estimator $\boldsymbol{\beta}_j^*$ will pull $\hat{\boldsymbol{\beta}}_j$ towards $\hat{\hat{\boldsymbol{\beta}}}_j$ in estimation, consequently the EB estimator is also called the shrinkage estimator. Note that the EB estimator in (32) assumes that the fixed effects, $\hat{\boldsymbol{\gamma}}$, are known.

Alternatively, the random level-1 coefficients can be estimated separately for each level-2 unit using the method of OLS, that is only $\hat{\boldsymbol{\beta}}_j^{OLS}$ can be used as an estimator of the level-1 coefficients. Unfortunately this estimator lacks precision because of small group sample sizes [25]. Another approach is the pooled regression estimation, in this case single Level-1 coefficients are estimated from the whole data set disregarding the hierarchy. The downside of this approach is that a level-1 coefficient for any level-2 unit might be biased because the coefficient of that group might differ significantly from the pooled estimate

[25].

2.2.3 Variance-covariance components

In the discussion of the estimation of fixed effects and random level-1 coefficients it was assumed for convenience that the variance-covariance components were known but in practice these are often not given and have to be estimated. When the group sample sizes are the same and the groups (i.e level-2 units) have predictors that are distributed identically (i.e $\mathbf{X}_j = \mathbf{X}$) formulas for estimating these components are available, but for unbalanced designs (i.e unequal group sample sizes), iterative procedures are employed to achieve efficiency (i.e greater precision), the following are the commonly used iterative procedures found in most statistical packages (e.g SAS):

1. *Full maximum likelihood* (MLF)

   In this procedure consistent and asymptotically efficient estimates of $\boldsymbol{\gamma}$, $\boldsymbol{\Phi}$ and σ^2 are obtained for which the likelihood of observing the values of the response $\mathbf{Y}$:$(\Sigma_{j=1}^{N} n_j \times 1)$ is a maximum.

2. *Restricted maximum likelihood* (REML)

   The drawback of the MLF estimation procedure is that it does not take into account the loss in degrees of freedom resulting from the estimation of the fixed effects (i.e the γ's), in other words the difference between the MLF and REML estimators is that the former is biased downwards (but asymptotically unbiased) [25], whereas the former is unbiased.

3. *Other iterative procedures*

   Goldstein [5, 4] developed an iterative generalized least squares (IGLS) procedure and showed that it can provide unbiased estimates of the variance components. And Longford [14] proposed a Fisher scoring algorithm, Goldstein [6] later proved that under normality the Fisher scoring algorithm and the iterative generalized least squares procedure are equivalent.

Note that for the estimation of the fixed effects and random level-1 coefficients the only methods of estimation considered were OLS and GLS, these apply only when the group (level-2 units) sample sizes are equal and sufficiently large in order to achieve the highest precision [11]. In practice however this is a rare situation, for example, in repeated measures studies responses nested within an individual collected overtime are often different for different individuals as there can be non response. Thus in general to account for the loss in the precision of estimation of the random level-1 coefficients due to small group sample sizes the iterative procedures are used in the estimation of the multilevel parameters [2].

3 Simulation study

To get an idea of how the multilevel analysis approach is applied in practice, in this section a series of simulated two-level analysis are conducted in the form of an illustration. The dataset for each model fitted model has 10000 level-1 units nested in 400 level-2 units each of size 25. A comprehensive description of the simulated dataset is provided for each model. The analysis proceeds similar to that of Singer [23]. That is, the variation in the response variable ($\mathbf{Y}_j$) between level-2 units is examined first, followed by a separate examination of the effect of the level-1 predictor ($\mathbf{X}_j$) and a level-2 predictor (z_j) on the response variable. Lastly, a combined effect of both the level-1 and level-2 predictors is examined. Both level predictors are scaled: level-1 predictors are group-mean centered and level-2 predictors are grand-mean centered. The models are estimated using the SAS/STAT® software's PROC MIXED[1], its default estimation method is REML. The SAS code is provided in Appendix 7.1.1.

3.1 Unconditional Model

To ascertain the extent of the homogeneity or heterogeneity in the response, $\mathbf{Y}_j$, between the level-2 units, the one-way ANOVA model is fitted. The level-1 and level-2 models are as specified in equations (10) and (11), respectively. A description of the simulated data is provided in Table 1.

Two-level data structure			
Variable	Generated by	Number of Level-1 units	10000
Response (Outcome)	$\mathbf{Y}_j = \mathbf{X}_j\boldsymbol{\beta}_j + \mathbf{r}_j$	Number of Level-2 units	400
Level-1 predictor	No predictors at	Size of each level-2 unit	25
Level-2 predictor	both levels.		
Parameters chosen			
Random effects	Level-1	$\mathbf{r}_j \sim N(\mathbf{0}, 5^2\mathbf{I})$	
	Level-2	$u_{0j} \sim N(0, 100)$	
Fixed effects	$\gamma_{00} = 250$		
Level-1 coefficients	$\beta_{0j} = \gamma_{00} + u_{0j}$, for j=1,2,...,400		

Table 1: Description of the simulated data-Unconditional model

Output-and-results

The results obtained from fitting the model are given in Table 2, the interpretation follows.

[1]SAS Institute Inc. 2009. *SAS/STAT ® 9.2 User's Guide, Second Edition.* Cary, NC: SAS Institute Inc.

Fixed effects				
Parameter	Parameter estimates	Standard error (s.e)	P-value	Significance test
Overall response mean ,γ_{00}	250.39	0.5008	$<.0001$	The estimate is significant at a 5% level of significance.
Variance components				
Between-Level-2 variation, τ_{00}	99.3241	7.1035	$<.0001$	Both estimates are significant at a 5% level of significance.
Within level-2 variation, σ^2	25.2024	0.3638		

Table 2: Unconditional Model-simulation results

Fixed-effects

The fitted model has one fixed effect, γ_{00}, that is the mean (average) of the response variable, and it is estimated as 250.39. The estimate is statistically significant (see Table 2).

Variance-components

The fitted model has two unconditional random effects, u_{0j} and r_{ij}, that is, the deviation of any level-2 unit's response from the overall mean of the response (i.e γ_{00}) and the deviation of any level-1 unit's response from the overall response where the unit belongs (i.e the level-2 unit). The random effects are measured by the variance components τ_{00} and σ^2, respectively. The former is estimated as 99.3241 and the latter is estimated as 25.2024. Both estimates are statistically significant (see Table2). These are unconditional because there is no predictor explaining the variation in the response variable in either level.

As mentioned, this model is important for preliminary analysis because it gives the between and within level-2 units distribution of the variation in the response, to quantify this distribution, the ICC is computed for the proportional variation between level-2 units using (13)

$$\begin{aligned}\rho =& \frac{\hat{\tau}_{00}(\text{unconditional})}{\hat{\tau}_{00}(\text{unconditional}) + \hat{\sigma}^2(\text{unconditional})} \\ =& \frac{99.3241}{99.3241 + 25.2024} \\ =& 0.797 \text{ or } 79.7\%\end{aligned}$$

Thus, it can be said that about 80% of the variability in the response is attributable to the level-2 units. This measure is useful since it tells us that disaggregation (alternative to multilevel modeling, see Section 5) will likely yield misleading results.

3.2 Level-1 predictor only model

To investigate the variation in the response explained by the level-1 predictor $\mathbf{X}_j$, the one-way ANCOVA model is fitted. In this model, the intercepts (i.e the β'_{0j}s) are allowed to vary while keeping the slopes constant across groups. The level-1 and level-2 models are specified in Equations (33) and (34), respectively. A description of the simulated data is provided in Table 3.

The simulated data was generated by the following level-1 and level-2 model

Level-1: Model

$$\begin{aligned}\mathbf{Y}_j =& \mathbf{X}_j\boldsymbol{\beta}_j + \mathbf{r}_j \qquad \text{, where } \mathbf{r}_j \sim N(\mathbf{0}, \sigma^2\mathbf{I}_j) \\ =& \begin{pmatrix} 1 & (\mathrm{X}_{1j} - \mathrm{X}_{\bullet j}) \\ 1 & (\mathrm{X}_{2j} - \mathrm{X}_{\bullet j}) \\ \vdots & \vdots \\ 1 & (\mathrm{X}_{n_j j} - \mathrm{X}_{\bullet j}) \end{pmatrix} \begin{pmatrix} \beta_{0j} \\ \beta_{1j} \end{pmatrix} + \mathbf{r}_j \end{aligned} \tag{33}$$

where $X_{\bullet j}$ = the group mean of X_{ij}

Level-2: Model

$$\begin{aligned}\boldsymbol{\beta}_j =& \mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{u}_j \\ =& \begin{pmatrix} 1 & \mathrm{x}_{\bullet j} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \gamma_{00} \\ \gamma_{01} \\ \gamma_{10} \end{pmatrix} + \mathbf{u}_j \\ \mathbf{u}_j \sim& N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & 0 \\ 0 & 0 \end{pmatrix} \right] \end{aligned} \tag{34}$$

By substituting (34) into (33), the multilevel model is obtained

$$\mathbf{Y}_j = \mathbf{X}_j\mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{X}_j\mathbf{u}_j \tag{35}$$

Because of group mean centering τ_{00} is now the variance of $\mathbf{Y}_j$ excluding the effect of the level-1 predictor (i.e having not accounted for the effect of X_{ij}) [10], β_{0j} in (34) is the unadjusted mean of the $\mathbf{Y}_j$ for the j^{th} level-2 unit [26]. This implies that the level-1 predictor, X, does not explain the variability in $\mathbf{Y}_j$. To account for the variation explained by this variable an aggregated variable, z_j, of the level-1 predictor between groups (i.e the between-group mean of X_{ij}) is included as a level-2 predictor. The

inclusion of this aggregate variable controls for the level-1 predictor[10].

Two-level data structure			
Variable	Generated by	Number of Level-1 units	10000
Response (Outcome)	$\mathbf{Y}_j = \mathbf{X}_j\boldsymbol{\beta}_j + \mathbf{r}_j$	Number of Level-2 units	400
Level-1 predictor	$\mathbf{X}_j \sim N(20, 36)$	Size of each level-2 unit	25
Level-2 predictor(aggregate of $\mathbf{X}_j$)	$\mathbf{X}_j$		
Parameters chosen			
Random effects	Level-1	$\mathbf{r}_j \sim N(\mathbf{0}, 5^2\mathbf{I})$	
	Level-2	$\mathbf{u}_j \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 120 & 0 \\ 0 & 0 \end{pmatrix}\right]$	
Fixed effects	$\boldsymbol{\gamma} = \begin{pmatrix} 130 & 100 & 240 \end{pmatrix}'$		
Level-1 coefficients	$\boldsymbol{\beta}_j = \boldsymbol{\gamma} + \mathbf{u}_j$, for j=1,2,...,400		

Table 3: Description of the simulated data One-way ANCOVA model

Output-and-results

The results from fitting the combined model in (20) are provided in Table (4) followed by the interpretation of the results.

Fixed effects				
Parameter	Parameter estimates	Standard error (s.e)	P value	Significance test
Overall response mean, γ_{00}	131.50	9.9613	<.0001	Both estimates are significant at a 5% level of significance.
Aggregated X_{ij}, γ_{01}	339.96	0.4974		
Overall slope, γ_{10}	240.00	0.008633		
Variance components				
Between level-2 units variation,τ_{00}	104.78	11.2114	<.0001	Significant at a 5% level of significance.
Within level-2 units variation, σ^2	25.2222	0.3641		

Table 4: One-way ANCOVA model-simulation results

Fixed-effects

The model has two fixed effects: γ_{00} and γ_{10}, that is, the overall mean of Y and the average increase or decrease in the response for a unit increase in X across the level-2 units (i.e average slope), respectively. The estimate for the former is 131.74 and for the latter is 240. The coefficient of the aggregated level-1 predictor is unbiased, 339.95, because it is not equal to the one used in the simulation, 200. The unbiasedness is introduced by group-mean centering (see below). All estimates are statistically significant (see Table 4).

Variance-components

The model has three random effects: u_{0j} and r_{ij}, that is, the deviation of each level-2 unit's response from the overall average (i.e γ_{00}), the deviation of each level-2 unit's slope (i.e Y-X relationship) from

the overall average slope (i.e γ_{11}) and the deviation of each level-2 unit's slope (i.e β_{1j}) from the overall average (i.e γ_{10}) and the randomness in Y between level-1 units (or within level-2 units). These are, respectively, measured by their associated variance terms, $\hat{\tau}_{00} = 106.82$ and $\hat{\sigma}^2 = 260.93$. All the variance components estimates are statistically significant.

3.3 Level-2 predictor only model

To investigate the effect of the level-2 predictor, Z_j, on the mean response across level-2 units, the means-as-outcomes model is fitted. The level-1 and level-2 models are specified in (36) and (37), respectively. A description of the simulated data is provided in Table 5.

The simulated data is generated by the following level-1 and level-2 models

Level-1 Model

$$\begin{aligned}\mathbf{Y}_j =& \mathbf{X}_j\boldsymbol{\beta}_j + \mathbf{r}_j, \quad \text{where } \mathbf{r} \sim N(\mathbf{0}, \sigma^2\mathbf{I}_{n_j}) \\ =& \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \beta_0 \\ 0 \end{pmatrix} + \mathbf{r}_j\end{aligned} \tag{36}$$

Level-2 Model

$$\begin{aligned}\boldsymbol{\beta}_j =& \mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{u}_j \\ =& \begin{pmatrix} 1 & (z_1 - z_{\bullet\bullet}) \\ 1 & (z_2 - z_{\bullet\bullet}) \\ \vdots & \vdots \\ 1 & (z_j - z_{\bullet\bullet}) \end{pmatrix} \boldsymbol{\gamma} + \mathbf{u}_j \\ \text{where } \mathbf{u}_j \sim N&\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & 0 \\ 0 & 0 \end{pmatrix} \right] \text{ and } z_{\bullet\bullet} = \text{Overall mean}\end{aligned} \tag{37}$$

By substituting (37) into (36), the multilevel model is obtained

$$\mathbf{Y}_j = \mathbf{X}_j\mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{X}_j\mathbf{u}_j + \mathbf{r}_j \tag{38}$$

Unlike group-mean centering, grand-mean centering takes into account the effect of the level-2 predictors included in the model and thus for this model it is not necessary to make adjustments [10].

Two-level data structure			
Variable	Generated by	Number of Level-1 units	10000
Response (Outcome)	$\mathbf{Y}_j = \mathbf{X}_j\boldsymbol{\beta}_j + \mathbf{r}_j$	Number of Level-2 units	400
Level-2 predictor	$z_j \sim N(15, 25)$	Size of each level-2 unit	25
Parameters chosen			
Random effects	Level-1	$\mathbf{r}_j \sim N(\mathbf{0}, 5^2\mathbf{I})$	
	Level-2	$u_{0j} \sim N(0, 400)$	
Fixed effects	$\boldsymbol{\gamma} = \begin{pmatrix} 150 & 75 \end{pmatrix}'$		
Level-1 coefficients	$\beta_{0j} = \begin{pmatrix} 1 & z_j \end{pmatrix}\boldsymbol{\gamma} + u_{0j}$, for j=1,2,...,400		

Table 5: Description of the simulated data-Means as outcomes model

Output-and-results

Table (6) gives the results from fitting the multilevel model in (17). The interpretation of the results follows.

Fixed effects				
Parameter	Parameter estimates	Standard error (s.e)	P value	Significance test
Overall mean response, γ_{00}	152.93638	0.9878	$<.0001$	Both estimates are significant at a 5% level of significance
Z, γ_{01}	74.6426	0.1875	$<.0001$	
Variance components				
Between level-2 units variation, τ_{00}	388.16	27.5870	$<.0001$	Both estimates are significant at a 5% level of significance.
Within level-2 units variation,σ^2	25.0711	0.3619		

Table 6: Means as outcomes model-simulation results

Fixed-effects

The fitted model has two fixed effects, γ_{00}, the overall mean response, and γ_{01}, the increase in a level-2 unit's y-response for a unit increase in the the z-predictor value. The estimated fixed effects are 152.93638 and 74.6426, respectively. Both estimates are statistically significant (see Table 6).

Variance-components

The fitted model has two random effects, u_{0j} and r_{ij}. The former is a random term indicating differences in Y between level-2 units, after controlling for the level-2 variable Z, whereas the latter is a random term indicating the deviation of any level-1 unit's value of Y from the group average. They are measured by the covariance components, τ_{00} and σ^2, respectively. The covariance components are estimated as 388.33 and 49.1393. Both estimates are statistically significant (see Table 6).

3.4 Level-1 and Level-2 predictors only model

To investigate what the combined effect of both X and Z has on Y, the non-varying slope model is fitted, so that only the intercepts vary across level-2 units. The level-1 and level-2 models are specified in (39) and (40). A description of the simulated data is provided in Table 7.

The simulated data is generated by the following model.

Level-1 Model

$$\begin{aligned}\mathbf{Y}_j =& \mathbf{X}_j\boldsymbol{\beta}_j + \mathbf{r}_j, \;\; \mathbf{r}_j \sim N(\mathbf{0}, \sigma^2\mathbf{I}_{n_j}) \\ =& \begin{pmatrix} 1 & (\mathrm{X}_{1j} - \mathrm{X}_{\bullet j}) \\ 1 & (\mathrm{X}_{2j} - \mathrm{X}_{\bullet j}) \\ \vdots & \vdots \\ 1 & (\mathrm{X}_{n_j j} - \mathrm{X}_{\bullet j}) \end{pmatrix} \begin{pmatrix} \beta_{0j} \\ \beta_{1j} \end{pmatrix} + \mathbf{r}_j \end{aligned} \tag{39}$$

Level-2 Model

$$\begin{aligned}\boldsymbol{\beta}_j =& \mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{u}_j \\ =& \begin{pmatrix} 1 & [z_j - z_{\bullet\bullet}] & x_{\bullet j} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \boldsymbol{\gamma} + \begin{pmatrix} u_{0j} \\ 0 \end{pmatrix} \\ \mathbf{u}_j \sim& N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & 0 \\ 0 & 0 \end{pmatrix} \right] \end{aligned} \tag{40}$$

Combined-Model

$$\mathbf{Y}_j = \mathbf{X}_j\mathbf{Z}_j\boldsymbol{\gamma} + \mathbf{X}_j\mathbf{u}_j + \mathbf{r}_j$$

$x_{\bullet j}$ and $z_{\bullet\bullet}$ are as specified in the previous models.

Two-level data structure			
Variable	Generated by	Number of Level-1 units	10000
Response (Outcome)	$\mathbf{Y}_j = \mathbf{X}_j\boldsymbol{\beta}_j + \mathbf{r}_j$	Number of Level-2 units	400
Level-1 predictor	$\mathbf{X}_j \sim N(20, 36)$	Size of each level-2 unit	25
Level-2 predictor	$Z_j \sim N(15, 25)$		
Parameters chosen			
Random effects	Level-1	$\mathbf{r}_j \sim N(\mathbf{0}, 5^2\mathbf{I})$	
	Level-2	$\mathbf{u}_j \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 100 & 0 \\ 0 & 0 \end{pmatrix} \right]$	
Fixed effects	$\boldsymbol{\gamma} = \begin{pmatrix} 200 & 400 & 200 & 100 \end{pmatrix}'$		
Level-1 coefficients	$\boldsymbol{\beta}_j = \mathbf{I}_{2x2} \otimes \begin{pmatrix} 1 & z_j \end{pmatrix} \boldsymbol{\gamma} + \mathbf{u}_j$, for j=1,2,...,400		

Table 7: Description of the simulated data- Non-varying slope model

Output-and-results

Table 8 gives the results from fitting the model, the interpretation follows.

Fixed effects				
Parameter	Parameter estimates	Standard error (s.e)	P value	Significance test
For response variable: β_{0j}				
Overall mean response,γ_{00}	176.62	16.4924	<.0001	All parameter estimates are significant at a 5% level of significance.
Z, γ_{01}	397.87	0.7385	<.0001	
X., γ_{02}	301.43	1.6375	<.0001	
For response variable: β_{1j}				
Overall mean slope, γ_{10}	99.9979	0.01649	<.0001	
Variance components				
Between Level-2 units variation, τ_{00}	93.3706	9.4571	<.0001	All variance-covariance estimates significant at a 5% level of significance.
Within Level-2 units variation, σ^2	24.4364	0.3491		

Table 8: Non-varying slope model-simulation results

Fixed-effects

The fitted model has four fixed effects: the overall mean response, γ_{00}, estimated to be 176.62; the pooled within-group regression coefficient of Y on X, γ_{10}, estimated to be 99.9979 and the increase in any level-2 unit's response for a unit increase in its Z value, γ_{01}, estimated to be 398.87. Lastly the coefficient of the aggregated level-1 predictor is estimated as 301.43. The simulation was conducted with $\gamma_{02} = 200$, because of group-mean centering the overall within-group slope, γ_{10}, is added to γ_{02}. This shows that caution has to be applied to the use of group-mean centering. All the regression coefficients are statistically significant (see Table 8).

Variance-components

The fitted model has three random effects: u_{0j}, the unexplained variation in the within-group Y after controlling for the level-2 predictor Z and $\mathbf{r}_j$, the unexplained variation in a level-1 unit's Y value after controlling for the unit's X value. Their corresponding variability measures are τ_{00} and σ^2, respectively. They are estimated as 93.3706 and 24.4364, respectively. All the variance components are statistically significant (see Table 8).

In summary, from the simulation study it can be seen that in multilevel analysis care must be exercised when drawing conclusions especially when the predictors are group-mean centered, under group-mean centering the models get more complex when there are more levels in the hierarchy. For practical purposes the recommended method of scaling is grand-mean centering.

4 Application

The TIMSS 2015 science data featured 12514 South African grade 9 students (The highest international assessment level is grade 8) from 292 schools, a summary of the results is provided in the Table 9.

Assessment	TIMSS (Trends In International Mathematics and Science Study)					
Subject	Science					
Country	South Africa					
Level	Grade 9					
Year	2015					
Summary Assessment Results						
	Number of schools	Number of students	Average Score	Minimum Score	Maximum Score	Range Score
	292	12514	355.6606	22.56688	795.7567	773.189

Table 9: The South African TIMSS 2015 assessment results (Author's own calculations)

Note that the TIMSS uses five plausible values to measure achievement in science, the result above are based on the first plausible value as the measure of the outcome variable (science achievement). An investigation is undertaken to determine the factors that gave rise to the above performance, significance studies are conducted in identifying influential factors and to better understand the variation in science achievement multilevel modeling techniques are employed to fit models at different levels of education, this is extremely essential in order to be able to design interventions at any level of the hierarchy [7].

4.1 Multilevel variables

4.1.1 Response (Outcome) variable

TIMSS obtained a science achievement measure by making use of, among others, plausible values, five plausible values, based on a sample of the items (questions), are used as a multiple estimate of how a student might have performed if the student attempted all the items [15], because the unit of analysis in this study is the student the response variable forms part of the student-level variables. In this analysis the first plausible value is used as a measure of science achievement.

4.1.2 Level-1: student-level variables

According to Mohammadpour [15] factors contributing to the variation in science achievement can be categorized as either attitude (e.g student's confidence in science), personal (e.g. language spoken by student at home) or socioeconomic based (e.g. parent's level of education) factors. Using the South African TIMSS 2015 science data, the first plausible value will serve as the outcome variable (science

achievement) as detailed above, student-level factors, from each category, will be considered and the highly influential, that is, successful at adequately accounting for the differences in science performance of the South African 9th graders, will be included in the model. Table 10 gives the description of each of the student-level variables considered in the study as well the degree of their linear relationship with the response variable.

Response variable	Student-level variable(s)	Category	Degree of correlation[2]	Description of variable
SCIACH	SCS	Attitudinal	0.20924	Student's confidence in science, as measured by student's response to how much the agree with eight statements about their performance in science. More confidence in science receives the largest scale whereas small scaling for less confidence.
	SATS		0.20486	Student's attitude towards science, as measured by the student's level of agreement to nine statements about their feeling towards science. More positive feeling receives large scaling whereas less scaling for more negative feelings.
	STUB	Personal	0.24458	Student bullying, as measured by the student's response to how often they were bullied (e.g weekly,monthly). More frequent experience of bullying receives the lowest scale whereas large scaling for less frequent bullying.
	HER	Socioeconomic	0.27517	Student's home educational resources as measured by the student's response to two questions on the number of educational resources available at home and one on the highest level of education of either of the student's parents. More resources and highest education level receives the largest scale and smaller scale otherwise.

Table 10: Defining the student-level predictors and their correlation with the response variable science achievement

4.1.3 Level 2: school-level variables

Students are clustered in schools and the students within these schools share many characteristics of instruction (i.e. teaching) and learning (e.g. same desk), Mohammadpour [15] categorizes the school-level factors that impact on academic (e.g. science) achievement under school climate (e.g. school discipline problems, learning environment) and school contextual factors (e.g school science resources,

school location). Table 11 gives a description of the school-level variables and their relationship with science achievement.

Response variable	School-level variables	Category	Degree of correlation	Description
SCIACH	SCHSLAB	Contextual factors	0.24479	School has a science laboratory, a categorical variable with 1=Yes and 2=No.
	SCHSRS		0.20237	School's resources used in the learning and teaching of science.
	SCHDP	School climate	0.19547	School's discipline problems as perceived by the school principal.
	SCHEAS		0.25880	School's emphasis on the student's academic success as perceived by the school teachers.

Table 11: Defining school-level predictors and their correlation with the science achievement

4.2 Multilevel models

The models are fitted in connection with questions concerning the science achievement of students in South African schools, provided in Table 12.

Questions	Model
1. Is the science achievement of South African students largely influenced by student- or school-level factors?	Unconditional model
2. What is the portion of the variation, in science achievement, that is accounted for by the student-level factors in Table 10	Student predictors only model
3. What is the portion of the variation, in science achievement, that is accounted for by the school-level factors in Table 11	School predictors only model
4. How much variability is explained by the combined effect of the student- and school-level factors (only those in Tables 10 and 11)?	Student and school predictors model

Table 12: Research Questions

The models are fitted using SAS PROC MIXED. Both student- and school-level variables are grand mean centered. The necessary SAS code for fitting the models is provided in Appendix 7.1.2.

4.2.1 Unconditional means model

The first model to be fit in multilevel analysis is the unconditional means model [24].This model can be interpreted as a one-way ANOVA with random effects [23], it is recognized by Raudenbush and Bryk [18] as an essential tool in preliminary multilevel data analysis, because it quantifies the variation in the outcome variable between and within each level in the hierarchy. Without the inclusion of any student- or school-level predictors, this model gives the structure of how the variability in science achievement is

apportioned between the levels in the hierarchy.

$$\mathbf{SCIACH}_j = \beta_{0j}\mathbf{1} + \mathbf{r}_j \tag{41}$$

$$\beta_{0j} = \gamma_{00} + u_{0j} \tag{42}$$

$$\text{where } \mathbf{r}_j \sim N(\mathbf{0}, \sigma^2\mathbf{I}_{n_j}), \qquad u_{0j} \sim N(0, \tau_{00})$$

The level-1 model predicts the SCIACH, for each school, using a linear combination of the j^{th} school's mean SCIACH, $\beta_{0j}\mathbf{1}$, with vector $\mathbf{1}$ for the intercept, and some random deviations, $\mathbf{r}_j$, assumed to be normally distributed with mean $\mathbf{0}$ and a constant variance, $\sigma^2\mathbf{I}_{n_j}$, which gives the within-school variation in SCIACH. The level-2 model is a function of the overall mean SCIACH, γ_{00}, plus a random term, u_{0j}, assumed to be normally distributed with mean 0 and variance τ_{00} which gives the between-school variation in SCIACH. When substituting (42) into (41) the following combined (multilevel) model is obtained:

$$\mathbf{SCIACH}_j = \gamma_{00}\mathbf{1} + u_{0j}\mathbf{1} + \mathbf{r}_j \tag{43}$$

The vector $\mathbf{1}$ on the level-2 random term shows that the intercept is varied between schools.

Output and results

Table 13 gives the output obtained from fitting the model in (43), the interpretation of the output follows.

Fixed Effects				
Parameter	Parameter estimate	Standard error (s.e)	P-value	Significance test
γ_{00}	361.96	4.5907	$< .0001$	The estimate is statistically significant at a 5% level of significance.
Variance-covariance estimates				
Between-school variation,τ_{00}	6009.20	511.70	$< .0001$	Both estimates are statistically significant at a 5% level of significance.
Within school-variation,σ^2	5160.60	66.0201	$< .0001$	

Table 13: One-way ANOVA model results

Fixed-effects: Under this model there is only one fixed effect, γ_{00}, its estimate is 361.96 and it is statistically significant with $p-value < .0001$ (thus reject the null hypothesis that $H_0 : \gamma_{00} = 0$)(see Table 13). The estimate gives the mean school-level science achievement of the 292 schools. Note the caveat by Singer[23] of misinterpreting this value as a student level average.

Variance-components: Since the fitted model is unconditional (i.e predictor-free) the variances are unconditional estimates of the random-effects portion of the model, σ^2, for the variation in $\mathbf{r}_j$, is estimated by $\hat{\sigma}^2 = 5160$ and τ_{00}, for the variation in u_{0j}, is estimated by $\widehat{\tau}_{00} = 6009.2$, both of these estimators are significant with a both their $p - value < .0001$. These estimates are revealing what was expected, that is a considerable amount of variability in science achievement between and within schools, but there is 14% ($\frac{6009.2-5160}{6009.2}$) more variability between schools than within schools, to quantify this variation the ICC, ρ, is estimated, it answers the first research question

$$\hat{\rho} = \frac{\hat{\tau}_{00}}{\hat{\tau}_{00} + \hat{\sigma}^2} = \frac{6009.2}{6009.2 + 5160} \approx .54 \; or \; 54\% \tag{44}$$

Indicating that about 54% of the variation in science achievement is between schools. This emphasizes the need to make use of multilevel models and thus suggesting that a single-level (ordinary regression) model might yield misleading results.

4.2.2 Level-1 predictor(s) only model

To see how the student-level factors affect the student's science achievement, the random coefficients model is fitted. The slope(s) and intercept (i.e. the level-1 coefficients) included in the model will vary across schools which allows each school to have its own slope and intercept. This is the typical intercepts and slopes as outcomes model. The student-level model (45) and school-level model (46) are specified as follows

$$\begin{aligned} \mathbf{SCIACH}_j =& \mathbf{X}_j\boldsymbol{\beta}_j + \mathbf{r}_j \\ =& \begin{pmatrix} \mathbf{1} & \mathbf{SCS}_j & \mathbf{SATS}_j & \mathbf{STUB}_j & \mathbf{HER}_j \end{pmatrix} \begin{pmatrix} \beta_{0j} \\ \beta_{1j} \\ \beta_{2j} \\ \beta_{3j} \\ \beta_{4j} \end{pmatrix} + \mathbf{r}_j \end{aligned} \tag{45}$$

$$\boldsymbol{\beta}_j = \begin{pmatrix} \gamma_{00} + u_{0j} \\ \gamma_{10} + u_{1j} \\ \gamma_{20} + u_{2j} \\ \gamma_{30} + u_{3j} \\ \gamma_{40} + u_{4j} \end{pmatrix} = \boldsymbol{\gamma} + \mathbf{u}_j \tag{46}$$

$$\text{for } \mathbf{r}_j \sim N(\mathbf{0}, \sigma^2\mathbf{I}_{n_j}), \quad \text{and} \quad \mathbf{u}_j \sim N(\mathbf{0}, \mathbf{\Phi})$$

$$\text{where } \mathbf{\Phi} = \begin{pmatrix} \tau_{00} & \tau_{01} & \tau_{02} & \tau_{03} & \tau_{04} \\ \tau_{10} & \tau_{11} & \tau_{12} & \tau_{13} & \tau_{14} \\ \tau_{20} & \tau_{21} & \tau_{22} & \tau_{23} & \tau_{24} \\ \tau_{30} & \tau_{31} & \tau_{32} & \tau_{33} & \tau_{34} \\ \tau_{40} & \tau_{41} & \tau_{42} & \tau_{43} & \tau_{44} \end{pmatrix}$$

The student-level model in (45) is a conditional model, conditional upon the inclusion of student-level factors, thus the random error term, $\mathbf{r}_j$, is a residual assumed to be normally distributed with mean $\mathbf{0}$ and variance $\sigma^2\mathbf{I}_{n_j}$, where $\mathbf{I}_{n_j}$ is an n_j x n_j identity matrix. The elements of the vector $\boldsymbol{\beta}_j$, level-1 coefficients, are allowed to vary across schools in the level-2 model as functions of their school-level averages (e.g γ_{00}), and a random term (e.g u_{0j}). The random term in (46), $\mathbf{u}_j$, is assumed to be multivariate normally distributed with mean $\mathbf{0}$ and variance-covariance matrix $\mathbf{\Phi}$. Now each diagonal element of the variance-covariance matrix (i.e the τ_{kk}'s for k = 1,2,...,K) measures the variability of each regression coefficient (i.e β_{kj} for k = 1,2,...,K and j = 1,2,...,J) across schools and the off-diagonal elements measure the covariance between regression coefficients.

When substituting (46) into (45), a combined (multilevel) model is obtained and can be written as follows:

$$\mathbf{Y}_j = \begin{pmatrix} \mathbf{1} & \mathbf{SCS}_j & \mathbf{SATS}_j & \mathbf{STUB}_j & \mathbf{HER}_j \end{pmatrix} \begin{pmatrix} \gamma_{00} \\ \gamma_{10} \\ \gamma_{20} \\ \gamma_{30} \\ \gamma_{40} \end{pmatrix} +$$

$$\begin{pmatrix} \mathbf{1} & \mathbf{SCS}_j & \mathbf{SATS}_j & \mathbf{STUB}_j & \mathbf{HER}_j \end{pmatrix} \begin{pmatrix} u_{0j} \\ u_{1j} \\ u_{2j} \\ u_{3j} \\ u_{4j} \end{pmatrix} + \mathbf{r}_j$$

$$= \mathbf{X}_j\boldsymbol{\gamma} + \mathbf{X}_j\mathbf{u}_j + \mathbf{r}_j \tag{47}$$

Output and results

Table 14 gives the results from fitting the model in (47), the interpretation of the output follows.

Fixed Effects				
Parameter	Parameter estimates	Standard error (s.e)	P value	Significance test
γ_{00}	362.78	4.2239	<.0001	All estimates are statistically significant at a 5% level of significance.
γ_{10}	5.8676	0.4642		
γ_{20}	7.2540	0.4968		
γ_{30}	3.9301	0.4487		
γ_{40}	1.9391	0.5094	0.0002	
Variance components				
Within-school variance, σ^2	4343.69	59.5699	<.0001	
The school-level variance components are given in (48) in matrix form.				

Table 14: Random coefficients model results

Fixed-effects: The estimate for γ_{00}, the overall mean SCIACH after controlling for the SCS, SATS, STUB and HER is 362.78. The estimates for γ_{10}, γ_{02}, γ_{03} and γ_{04} are 5.8676, 7.2540, 3.9301 and 1.9391, respectively and they indicate the average slope across schools between science achievement and student confidence in science, student attitude towards science, frequency of student bullying per student and the student's home educational resources, respectively. All the coefficients are statistically significant as reflected by their p-values as shown in Table 14. The implication is that a student with a positive attitude and confidence towards science, who experiences less bullying and has enough educational resources at home with either of his or her parents possessing a diploma or above will perform better in science. A good attitude towards science followed by confidence in science are seen to be the most important, whereas the latter two are moderate to less than important, more especially the effect of home educational resources.

Variance-components: For the random effects, from the output the variance-covariance matrix is specified in (48).

$$\begin{pmatrix} & \text{Intercept} & \text{SCS} & \text{SATS} & \text{STUB} & \text{HER} \\ \text{Intercept} & \hat{\tau}_{00} & \hat{\tau}_{01} & \hat{\tau}_{02} & \hat{\tau}_{03} & \hat{\tau}_{04} \\ \text{SCS} & & \hat{\tau}_{11} & \hat{\tau}_{12} & \hat{\tau}_{13} & \hat{\tau}_{14} \\ \text{SATS} & & & \hat{\tau}_{22} & \hat{\tau}_{23} & \hat{\tau}_{24} \\ \text{STUB} & & & & \hat{\tau}_{33} & \hat{\tau}_{34} \\ \text{HER} & & & & & \hat{\tau}_{44} \end{pmatrix}$$

$$= \begin{pmatrix} 5063.89^{*} & -8.3968 & -189.48^{*} & -167.73^{*} & 204.61^{*} \\ & 6.8803 & -0.7359 & -3.1507 & -3.7484 \\ & & 15.1794^{*} & 12.1259^{*} & -6.4112 \\ & & & 14.8358^{*} & -5.4178 \\ & & & & 24.3760^{*} \end{pmatrix} \tag{48}$$

*p-value$<$0.05, statistically significant, otherwise statistically insignificant

The fitted model has five random effects measured by their corresponding variance-covariance components. For the variance in the intercepts, the variability in science achievement between schools has remained virtually the same (i.e from 6009.2 to 5915.24) after controlling for the student-level factors. For the covariance between slopes and intercepts, there exists statistically significant correlations between SATS-, STUB-, and HER-SCIACH slopes and the school science achievement (intercept). But the correlation is not significant for the SCS-SCIACH slope and the intercept (see (48), $\hat{\tau}_{01}$). This can be visually seen from Figure 1, in which the empirical Bayes (EB) estimates for the slopes (vertical axis) and intercepts (horizontal axis), the β_{0j}'s, are plotted[3]. In other words, the effects of SATS, STUB and HER on science achievement differ depending on school average science achievement, but for the effect of SCS on science achievement there is no evidence that suggests the above case. For the variances in slopes, all the slopes except, SCS-SCIACH slope, are each statistically significantly different across schools (see (48)). For the covariance between the slopes, only the effect of SATS on science achievement is significantly (statistically) correlated with that of STUB on science achievement. The other slope-slope relationships across schools are not statistically significant (see (48)). The variance within-schools declined by a considerable amount from that observed in the unconditional model. The decline can be used to answer the second research question.

$$\begin{aligned} &= \frac{\text{unconditional model}(\hat{\sigma}^2) - \text{conditional model}(\hat{\sigma}^2)}{\text{unconditional model}(\hat{\sigma}^2)} \\ &= \frac{5160.60 - 4341.94}{5160.6} \\ &= 0.1586 \text{ or} 15.86\% \end{aligned}$$

It follows that, about 16% of the explainable variation in science achievement within schools is explained by SCS, SATS, STUB and HER. Taking into account the caveat provided in Singer [23], that

[3]Note that these EB estimators are approximates computed by borrowing strength from the SAS PROC MIXED results, see code in Appendix

the explainable variation in SCIACH might be small as such the combined contribution of student-level factors might be negligible.

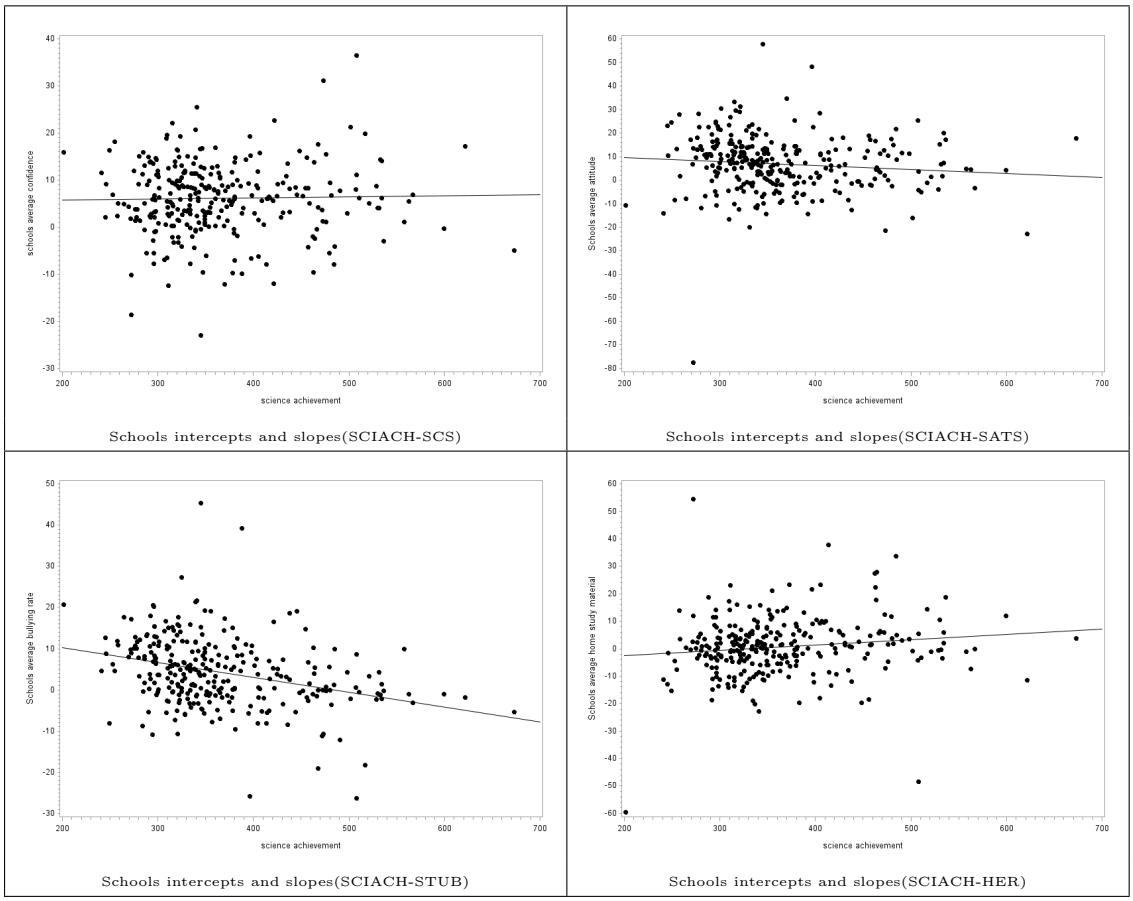


Figure 1: Plot of the empirical Bayes estimates for 291 school for the TIMSS 2015 data

4.2.3 Level 2 predictor(s) only model

To investigate and answer the third question, that is, to determine the effect of school-level factors on science achievement, the means-as-outcomes model is fitted with all the school-level factors in Table 11. The student-level model is as specified in (10) and (41) and the school-level model is specified as follows

$$\beta_{0j} = \gamma_{00} + \gamma_{01}SCHSLAB_j + \gamma_{02}SCHSRS_j + \gamma_{03}SCHDP_j + \gamma_{04}SCHEAS_j + u_{0j} \tag{49}$$

$$= \begin{pmatrix} 1 & SCHSLAB_j & SCHSRS_j & SCHDP_j & SCHEAS_j \end{pmatrix} \begin{pmatrix} \gamma_{00} \\ \gamma_{01} \\ \gamma_{02} \\ \gamma_{03} \\ \gamma_{04} \end{pmatrix} + u_{0j}$$

$$= \mathbf{Z}_j\boldsymbol{\gamma} + u_{0j}$$

where $u_{0j} \sim N(0, \tau_{00})$

Because of the inclusion of school-level predictors u_{0j} represents the residual after controlling for school-level factors. The residual term is normally distributed with mean 0 and variance τ_{00}. τ_{00}, as in the previous models, is the variability in science achievement between schools.

To obtain the combined model, substitute (49) into (41):

$$\mathbf{SCIACH}_j = \mathbf{Z}_j\boldsymbol{\gamma} + u_{0j}\mathbf{1} + \mathbf{r}_j \tag{50}$$

The vector $\mathbf{1}$ is an n_j x 1 of ones for the intercept, the second component $u_{0j}\mathbf{1}$ indicates that the intercept is random thus the model has one fixed-effect and two random effects.

Output and results

Table 15 gives the output from fitting the model in (50) followed by the interpretation of the output.

Fixed effects				
Parameter	Parameter estimates	Standard error (s.e)	P-value	Significance test
γ_{00}	333.35	5.3984	<.0001	All the coefficients are statistically significant at a 5% level of significance.
γ_{01}	53.0392	7.6058	<.0001	
γ_{02}	9.4000	2.5311	0.0002	
γ_{03}	7.8227	2.8980	0.0070	
γ_{04}	9.7992	2.1914	<.0001	
Variance covariance components				
Parameter	Parameter estimate		P-value	Significance test
Between-school variation,τ_{00}	3897.25	341.71	<.0001	Both variance estimates are significant at a 5% level of significance
Within-school variation,σ^2	5160.18	66.7388		

Table 15: Level-2 predictor(s) only model results

Fixed-effects: The fitted model has five fixed effects: γ_{00}, the average school science achievement, γ_{01}, the differences in average science achievement between schools with science labs (SCHSLAB=1) and those without science labs (SCHSLAB=0), γ_{02},γ_{03} and γ_{04}, the increase in a school's science achievement score following a unit increase in either SCHSRS, SCHDP or SCHEAS while keeping the effect of the other school-level predictors constant, respectively. They are estimated as 333.21, 53.1859, 9.3894, 7.8305 and 9.7942, respectively. All estimates are statistically significant (see Table 15). The implication of this results is that schools with science labs and other resources essential for teaching and learning science, having zero to no discipline issues and also encouraging their students to achieve academic success tend to perform well in science. It can be seen from Table 15 that having a science lab is the most important factor to better perform in science followed by the frequency of emphasis of the school's emphasis on academic success.

Variance-components:

The model has two random effects, u_{0j}, the difference in science achievement between schools after controlling for the school-level predictors and r_{ij}, the deviations of student science achievement scores from their school average achievements. The former is measured by τ_{00} and the obtained estimate of τ_{00} is 3925.97 and the latter is measured by σ^2 and, σ^2 is estimated as 5154.47. Both variance components are significant (see Table 15). There has been a sizeable decline in the between school variability in science achievement (from 6009.2 to 3925.97) which means that the school-level predictors included contribute to the explanation of a large portion of the variation in science achievement between schools. The within-school variability remained virtually the same (5160.60 unconditional compared with 5154.47 conditional).

To answer the third research, the decline in the between schools variation as a results of including school-level factors is used to compute the portion in the variability as a result of the factors.

$$
\begin{aligned}
&= \frac{\hat{\tau}_{00}(unconditional\ model) - \hat{\tau}_{00}(level-2\ predictor(s)\ only\ model)}{\hat{\tau}_{00}(unconditional\ model)} \\
&= \frac{6009.2 - 3925.97}{6009.2} \\
&= 0.3466 \text{ or } 34.66\%
\end{aligned}
$$

This value can be interpreted by saying that 34.66% of the explainable variation in school's science achievement is explained by school-level factors.

4.2.4 Combined level-1 and level-2 predictor(s) model

So far separate models, in both the student- and school-level, have been fitted and the effect of factors at those levels have answered some of the questions concerning the variation in science achievement across

and within South African schools. To determine the combined effect of both the student- and school-level factors, the intercepts and slopes as outcomes model is fitted. The student-level model in (51) and school-level model in (52) are as specified below

$$\begin{aligned}\mathbf{SCIACH}_j =& \mathbf{X}_j\boldsymbol{\beta}_j + \mathbf{r}_j,\ \mathbf{r}_j \sim N(\mathbf{0}, \sigma^2\mathbf{I}_{n_j}) \qquad (51)\\ =& \begin{pmatrix} \mathbf{1}_j & \mathbf{SCS}_j & \mathbf{SATS}_j & \mathbf{STUB}_j & \mathbf{HER}_j \end{pmatrix} \begin{pmatrix} \beta_{0j} \\ \beta_{1j} \\ \beta_{2j} \\ \beta_{3j} \\ \beta_{4j} \end{pmatrix} + \mathbf{r}_j \\ \boldsymbol{\beta}_j =& \begin{pmatrix} \gamma_{00} + \gamma_{01}SCHSLAB_j + \gamma_{02}SCHSRS_j + \gamma_{03}SCHDP_j + \gamma_{04}SCHEAS + u_{0j} \\ \gamma_{10} + \gamma_{11}SCHSLAB_j + \gamma_{12}SCHSRS_j + \gamma_{13}SCHDP_j + \gamma_{14}SCHEAS + u_{1j} \\ \gamma_{20} + \gamma_{21}SCHSLAB_j + \gamma_{22}SCHSRS_j + \gamma_{23}SCHDP_j + \gamma_{24}SCHEAS + u_{2j} \\ \gamma_{30} + \gamma_{31}SCHSLAB_j + \gamma_{32}SCHSRS_j + \gamma_{33}SCHDP_j + \gamma_{34}SCHEAS + u_{3j} \\ \gamma_{40} + \gamma_{41}SCHSLAB_j + \gamma_{42}SCHSRS_j + \gamma_{43}SCHDP_j + \gamma_{44}SCHEAS + u_{4j} \end{pmatrix} \\ =& (\mathbf{I}_5 \bigotimes \mathbf{Z}_j)\boldsymbol{\gamma} + \mathbf{u}_j,\ \mathbf{u}_j \sim N(\mathbf{0}, \boldsymbol{\Phi}) \qquad (52)\end{aligned}$$

$$\text{where}\quad \mathbf{Z}_j = \begin{pmatrix} 1 & SCHSLAB_j & SCHSRS_j & SCHDP_j & SCHEAS \end{pmatrix}$$

$$\text{and } \boldsymbol{\Phi} = \begin{pmatrix} \tau_{00} & \tau_{01} & \tau_{02} & \tau_{03} & \tau_{04} \\ \tau_{10} & \tau_{11} & \tau_{12} & \tau_{13} & \tau_{14} \\ \tau_{20} & \tau_{21} & \tau_{22} & \tau_{23} & \tau_{24} \\ \tau_{30} & \tau_{31} & \tau_{32} & \tau_{33} & \tau_{34} \\ \tau_{40} & \tau_{41} & \tau_{42} & \tau_{43} & \tau_{44} \end{pmatrix}$$

The student-level model in (51) is the same as the one obtained from the level-1 predictor(s) only model in (45) and as such it possesses the same properties, that is, the within-school error variance, $\mathbf{r}_j$, has a mean of $\mathbf{0}$ and a constant variance of $\sigma^2\mathbf{I}_{n_j}$. The school-level model in (52) is slightly different from the one obtained in the level-2 predictor(s) only model because of the varying student-level model coefficients in the school-level model which increases the number of random effects in the school-level model. Since there are five student-level model coefficients there are five school-level random effects as captured in the vector $\mathbf{u}_j$. Now, $\mathbf{u}_j$ is distributed as a multivariate normal variable with mean $\mathbf{0}$ and a

variance-covariance matrix of $\mathbf{\Phi}$.

After substituting (52) into (51), the combined model is obtained:

$$\mathbf{SCIACH}_j = \mathbf{Z}_j^* \boldsymbol{\gamma}^* + \mathbf{X}_j \mathbf{u}_j + \mathbf{r}_j \tag{53}$$

Output and results

Table 16 gives the output from fitting the model in (53) , followed by the interpretation of the output.

The interaction terms are omitted because, (1) they are all statistically insignificant (with p-values $>$ 0.05); and (2) when included, some of the main fixed effects are statistically insignificant.

It follows that the fitted model has a school-level model as given in (54). The combined model in (53), with the school-level model as specified in (54), is referred to as a MANCOVA (Multiple ANCOVA) model with random effects.

$$\boldsymbol{\beta}_j = \begin{pmatrix} \gamma_{00} + \gamma_{01} SCHSLAB_j + \gamma_{02} SCHSRS_j + \gamma_{03} SCHDP_j + \gamma_{04} SCHEAS_j + u_{0j} \\ \gamma_{10} + u_{1j} \\ \gamma_{20} + u_{2j} \\ \gamma_{30} + u_{3j} \\ \gamma_{40} + u_{4j} \end{pmatrix} \tag{54}$$

Fixed effects				
Parameter	Parameter estimates	Standard error (s.e)	P-value	Significance test
γ_{00}	343.49	4.8028	$<.0001$	All the coefficients are statistically significant at a 5% level of significance
γ_{01}	36.8431	6.3173	$<.0001$	
γ_{02}	6.2149	2.1459	0.0038	
γ_{03}	7.3940	2.4130	0.0022	
γ_{04}	8.3593	1.8264	$<.0001$	
γ_{10}	5.6515	0.4681		
γ_{20}	7.4196	0.4970		
γ_{30}	3.9649	0.4505		
γ_{40}	1.7556	0.5094	0.0135	
Variance covariance components				
Parameter	Parameter estimates	Standard error (s.e)	P-value	
Within-school variation, σ^2	4348.92	60.2748	$<.0001$	The estimate is statistically significant at a 5% level of significance.
The school-level variance components between level-2 units are given in (55).				

Table 16: Combined level-1 and level-2 predictors model results

Fixed-effects: The fitted model has nine fixed effects: The estimated average school science achievement is about 343, γ_{00}. The average slope between science achievement and student's confidence in science, γ_{10}; student's attitude towards science, γ_{20}; student bullying, γ_{30} and student's home educational resources, γ_{40}, across schools are estimated as 5.6515, 7.4196, 3.9649 and 1.7556, respectively. Schools with science laboratories have average science achievement that is 38.8431 (estimated γ_{01}) points higher than schools without science laboratories, when keeping the effect of the other school factors constant. The implication of the rest of the school-level factors is as follows: an additional school science resource, one less discipline problem and additional emphasis on academic success for any school will increase the school science achievement by 6.2149 (estimate for γ_{02}), 7.3940 (estimate for γ_{03}), 8.3593 (estimate for γ_{04}), respectively. Each will occur while keeping the others constant. All the coefficients are statistically significant (See Table 16).

Variance-components: For the random-effects, the following is a variance-covariance matrix of the fitted model:

$$\begin{pmatrix} & \text{SCIACH} & \text{SCS} & \text{SATS} & \text{STUB} & \text{HER} \\ \text{SCIACH} & \hat{\tau}_{00} & \hat{\tau}_{01} & \hat{\tau}_{02} & \hat{\tau}_{03} & \hat{\tau}_{04} \\ \text{SCS} & & \hat{\tau}_{11} & \hat{\tau}_{12} & \hat{\tau}_{13} & \hat{\tau}_{14} \\ \text{SATS} & & & \hat{\tau}_{22} & \hat{\tau}_{23} & \hat{\tau}_{24} \\ \text{STUB} & & & & \hat{\tau}_{33} & \hat{\tau}_{34} \\ \text{HER} & & & & & \hat{\tau}_{44} \end{pmatrix}$$

$$= \begin{pmatrix} 3495.50^{*} & -6.4946 & -142.63^{*} & -146.93^{*} & 121.85^{*} \\ & 6.8253 & -0.1534 & -3.5809 & -3.6904 \\ & & 13.6913^{*} & 11.6040^{*} & -5.7266 \\ & & & 14.1904^{*} & -3.7400 \\ & & & & 22.1796^{*} \end{pmatrix} \tag{55}$$

*p-value$<$0.05, statistically significant

The fitted model has six random effects, measured by their corresponding variance-covariance components given in (55). For the variance in the school SCIACH, there exists statistical significant variability between schools in their science achievement, measured by $\hat{\tau}_{00}$. This implies that there is additional variability that is not explained by the student- and school-level factors currently accounted for in the model. For the variance in the slopes, only the effect of home educational resources on science achievement varies

Goodness of fit			
Type of model	AIC	BIC	-2LL
Random intercepts and slopes model	131136.1	131194.5	131104.1
Random intercepts only model	131269.3	131276.6	131265.3

Table 17: Model comparison

significantly across schools, measured by $\hat{\tau}_{44}$ (see Equation 55). For the covariance between slopes and intercepts, the effect of student bullying and home educational resources on science achievement each differ depending on the school's science achievement. For the covariance between slopes, the effect of both the student's confidence and attitude towards science on science achievement vary significantly (see 55). That is, there is a correlation between the SCIACH-SATS slopes and SCIACH-SCS across schools. There are significant differences within schools, the variability declined by about 12% less than that observed in the level-1 predictor(s) only model. This means that the effect of group (school) factors is over and above the individual (student) factors. To measure the combined effect of both student- and school-level factors, the declined in the total variation is obtained,

$$
\begin{aligned}
&= \frac{\text{unconditional\_model}(\hat{\tau}_{00} + \hat{\sigma}^2) - \text{conditional\_model}(\hat{\tau}_{00} + \hat{\sigma}^2)}{\text{unconditional\_model}(\hat{\tau}_{00} + \hat{\sigma}^2)} \\
&= \frac{11169.8 - 7844.42}{11169.8} \\
&= 0.297 \text{ or} 30\%
\end{aligned}
$$

Thus about 30% of the explainable variation in science achievement is explained by the student- and school-level factors accounted for in the model.

The SAS PROC MIXED output provides a window to compare two models based on their goodness of fit. Under the 'Fit Statistics' section are measures useful in comparing multiple models with identical fixed effects but different random effects, AIC (Akiake information criterion) and BIC (Bayesian information criterion). Low values are evidence of a good fit. These measures can help ascertain whether both the slopes and intercepts should vary (random coefficients model) across schools or only the intercepts should vary.

From Table 17, it can be seen that the model that provides a good fit is the one where both the intercepts and slopes are allowed to vary across schools. This is a further reflection of the heterogeneity between schools.

5 Efficiency comparison analysis

In this section a comparative analysis is conducted in the form of an efficiency study in which multilevel analysis is contrasted and compared with the methods used for the analysis of hierarchical data prior to its development. These methods are namely, disaggregation and aggregation [27]. The former pools the data across all higher level units (e.g schools) and then proceeds by fitting a single level-1 model thereby ignoring the possible presence of the between group differences, whereas the latter deals with the hierarchy by fitting a single level-2 model using group means and thereby ignoring the within group individual differences. The TIMSS 2015 data considered in these paper are utilized in this analysis. Model diagnostics is performed for these two single regression models (i.e. aggregation and disaggregation) and a guideline to the appropriate use of multilevel modeling is provided. The three methods are demonstrated using science achievement (SCIACH) as the response variable and predictor from each level: student confidence in science (SCS) for level-1 and school science resources (SCHSRS) from level-2. Note that for the purposes of rendering the model coefficients interpretable the two predictor variables used in this analysis are centered: SCS is group mean centered and SCHSRS is grand mean centered. The SAS program was used to carry out this analysis, code is provided in Appendix . The results of fitting a regression model using all three methods are given in Tables 18 (Aggregation and Disaggregation) and 19 (Multilevel modeling). The findings under each method are discussed below.

	Disaggregation			Aggregation		
Parameter	Coefficient	standard error	p-value	Coefficient[4]	standard error	p-value
Intercept	358.95011	0.88135	<.0001	360.83890	4.36488	<.0001
SCS	10.71634	0.45224	<.0001	7.95682	6.63663	0.2315
SCHSRS	13.60378	0.59252	<.0001	16.34233	2.81465	<.0001
Assumption/statistic	Model diagnostics for aggregation and disaggregation regression.					
Multicollinearity:[5] VIF statistic	1.0000			1.01968		
Normality: (P-value)	1.6496590 (<.005)**			0.942908 (<.0001)**		

**P-value is extremely small reject the assumption of normality at a 5% level of significance.

Table 18: Aggregation and Disaggregation results: comparative study

Multilevel modeling			
Parameter	Coefficient	Standard error	P-value
Intercept, γ_{00}	362.32	4.3102	<.0001
SCS, γ_{10}	11.0389	0.3951	
SCHSRS, γ_{01}	16.1647	2.7621	
SCS*SCHSRS, γ_{11}	-0.5870	0.2601	0.0369
Variance components[$]			
Within-school variance, σ^2	4635.96	61.1182	<.0001
Between-school variance,τ_{00}	5251.01	448.74	<.0001
Covariance between slopes and intercepts, τ_{01} or τ_{10}	-108.66	30.1188	0.0003
Variance between the slopes, τ_{11}	10.7650	3.5435	0.0012

[$]All the variance components are statistically significant at a 5% level of significance

Table 19: Multilevel model results: comparative study

Disaggregation

As mentioned above this method ignores the possible between group differences and pools the data from all groups to fit a single level-1 model, by doing that this method violates the assumption of independence of the classical linear regression model (CLRM). This is as a result of the fact that students are assigned to school based on their residences as such they share certain characteristics (e.g same teacher, environment and classroom) [12][17]. The consequence of modeling hierarchy in this manner is the result of small standard errors for the model coefficients leading to a high probability of significance (i.e rejecting the $H_0 : \gamma = 0$) as can be seen from Table 18, the standard error for the coefficient of SCHSRS (s.e= 0.59252) is almost three times less than that of the estimated fixed effect associated with SCHSRS (s.e = 2.7621) obtained from the multilevel model(see fourth column of Table 18). Another violation, as a consequence of modeling hierarchy using disaggregation, is that of normality. The error term, in a typical two-variable regression model, is assumed to be normally distributed [8], to test if this is true, from a fitted model the residuals (i.e estimators of the error term) are tested for normality. Yap and Sim[28] found that the Shapiro-Wilk test for normality performs better than any test, using this test, the p-value is found to be $< .005$ as a result the assumption of normality is rejected (see Table 18 second column last row). And lastly, multicollinearity is not a threat with a Variance inflation factor (VIF)[6] of 1 for both predictors.

Aggregation

This method models the hierarchy by using school means as a result it ignores the within school differences, this implies that the model is predicting the average science achievement As a consequence, there is a loss of individual variability in the the response variable[17]. The results for the fitted model are in the third column of Table 18. All the coefficients are significant except that of average SCS (p-value = 0.2315). The

[6]If the VIF is at least 9 then it signals multicollinearity but for VIF at most 1 multicollinearity is not a problem, for more on this statistic see page 351 of Gujarati [8]

standard errors are larger than those obtained from fitting a multilevel model (see Table 19), especially that of the coefficient of average SCS (s.e = 6.63663). The normality assumption is severely violated under this method (with p-value < .0001), evidence enough to reject the assumption of normality (see figure 2a in Appendix 7.2). Lastly multicollinearity is not a threat with a VIF of 1.01968.

From the analysis above it can be seen that modeling hierarchy using traditional regression methods (i.e either aggregation or disaggregation) will result in overestimated, small standard errors[7], among others. To avoid drawing false conclusions, guidelines are provided to the appropriate and necessary use of the multilevel analysis approach:

1. For a two-level data structure, if the level-2 units are randomly sampled and inference is about the differences between the level-2 units, then multilevel analysis is necessary [12];

2. Whenever group sample sizes are similar across level-2 units, using either aggregation (i.e single level-1) or disaggregation (single level-2) will yield estimates to the fixed effects similar to those obtained from multilevel analysis but they will be less efficient (i.e Var($\hat{\gamma}_{agg}$) is large) than the multilevel model estimates [20].

In summary, it can be seen from the results of aggregation and disaggregation that the use of either of these methods results in a loss of variability, in essence these methods prevents the researcher from disentangling the student and school effect on the outcome variable [17]. The failure of these methods from being able to yield both the effect of the student and school effects on the response necessitates the use of multilevel modeling. From Table 19, the multilevel analysis results reveal that all the fixed effects as well as the random effects are significant (with p-value < .0001). The ICC 54% (see Equation 44) obtained from multilevel modeling shows that an OLS analysis of this data will yield considerable misleading results [23].

6 Conclusion

In this paper, a multilevel modeling technique useful in modeling hierarchical data was employed in a two-level analysis of students nested in schools. In this analysis student- and school-level factors are investigated to determine their contribution to the variation in science achievement among South African 9th-graders. The analysis was conducted using the TIMSS 2015 data. The results reveal that school factors contribute a large portion to the differences in science achievement across schools and also that the contextual-effects (i.e the social composition of the student body) are over and above the individual student effects. This key finding shows that educational policy should be focused towards school

[7] Although, in general, small standard errors are would be desirable in estimation, in this instance lead to increases in the Type I error (i.e the probability of falsely rejecting the null hypothesis)

reformation instead of student reformation. The results further highlight that schools that perform well in science have science labs, less discipline problems and they motivate their students to achieve academic success, and also students perform well in science if they have a positive confidence and attitude towards science, experience less bullying and have enough educational resources with either of their parents possessing a diploma or any other higher qualification.

Over and above science achievement, the multilevel analysis reveal that academic performance in South Africa is unstable. This is reflected by the degree of heterogeneity between schools. For the benefit of a country's economic development, it is essential that a schooling system provide stable quality education. This is so that it can produce consistent outcomes. In order to maintain stability there has to be an improvement in quality, perhaps further research is required to investigate the role of statistical process control (SPC) in addressing this matter.

References

[1] Mee-Ok Cho, Vanessa Scherman, and Estelle Gaigher. Exploring differential science performance in Korea and South Africa: A multilevel analysis. *Perspectives in Education*, 32(4):21–39, 2014.

[2] S.H.C du Toit. Analysis of multilevel models.part 1: Theoretical aspects. research report, Human Science Research Council, Pretoria, South Africa, 1993.

[3] George Frempong, Vijay Reddy, and Anil Kanjee. Exploring equity and quality education in south africa using multilevel models. *Compare: A Journal of Comparative and International Education*, 41(6):819–835, 2011.

[4] Harvey Goldstein. Multilevel mixed linear model analysis using iterative generalized least squares. *Biometrika*, 73(1):43–56, 1986.

[5] Harvey Goldstein. Restricted unbiased iterative generalized least-squares estimation. *Biometrika*, 76(3):622–623, 1989.

[6] Harvey Goldstein. Hierarchical data modeling in the social sciences. *Journal of Educational and Behavioral Statistics*, 20(2):201–204, 1995.

[7] Leonardo Grilli, Fulvia Pennoni, Carla Rampichini, and Isabella Romeo. Exploiting timss and pirls combined data: multivariate multilevel modelling of student achievement. *The Annals of Applied Statistics*, 10(4):2405–2426, 2016.

[8] Damoder N Gujarati. *Basic econometrics*. Tata McGraw-Hill Education, 2009.

[9] E.A Hanushek, L Woessmann, E.A Jamison, and D.T Jamison. Education and economic growth. *Education Next*, 8(2), 2008.

[10] David A. Hofmann and Mark B. Gavin. Centering decisions in hierarchical linear models: Implications for research in organizations. *Journal of Management*, 24(5):623 – 641, 1998.

[11] J. Hox. *Multilevel Modeling: When and Why*, pages 147–154. Springer, 1998.

[12] H. W. Ker. Application of hierarchical models/linear mixed-effects models in school effectiveness research. *Universal Journal of Educational Research*, 2(2):173–180, 2014.

[13] Ita GG Kreft, Jan De Leeuw, and Leona S Aiken. The effect of different forms of centering in hierarchical linear models. *Multivariate behavioral research*, 30(1):1–21, 1995.

[14] Nicholas Longford. A fast scoring algorithm for maximum likelihood estimation in unbalanced mixed models with nested random effects. *ETS Research Report Series*, 1987(1), 1987.

[15] Ebrahim Mohammadpour. A multilevel study on trends in malaysian secondary school students' science achievement and associated school and student predictors. *Science Education*, 96(6):1013–1046, 2012.

[16] Ebrahim Mohammadpour. A three-level multilevel analysis of singaporean eighth-graders science achievement. *Learning and Individual Differences*, 26:212 – 220, 2013.

[17] Jason W Osborne. Advantages of hierarchical linear modeling. *Practical Assessment, Research & Evaluation*, 7(1):1–3, 2000.

[18] Stephen W Raudenbush and Anthony S Bryk. *Hierarchical linear models: Applications and data analysis methods*. Sage, 2nd edition, 2002.

[19] S.W. Raudenbush and A.S. Bryk. *Hierarchical Linear Models: Applications and Data Analysis Methods*. Advanced Quantitative Technique. SAGE Publications, 2nd edition, 2002.

[20] Stephen W. Raudensbush and Anthony S. Bryk. *Hierarchical Linear Models:Applications and Data Analysis Methods*. Sage Publications, second edition, 1992.

[21] V Reddy, M Visser, L Winnaar, F Arends, A.L Juan, C Prinsloo, and K Isdale. Timss 2015: highlights of mathematics and science achievement of grade 9 south african learners. Technical report, Human Science Research Council (HSRC), 2016.

[22] S Schulze and M van Heerden. Learning environments matter: Identifying influences on the motivation to learn science. *South African Journal of Education*, 35:01 – 9, 05 2015.

[23] Judith Singer. Using SAS PROC MIXED to fit multilevel models, hierachical models, and individual growth models. *Journal of Education and Behavioral Statistics*, 24(4):323–355, 1998.

[24] Judith D Singer and John B Willett. Applied longitudinal data analysis: Modeling change and event occurrence, 2003.

[25] Marco R Steenbergen and Bradford S Jones. Modeling multilevel data structures. *American Journal of Political Science*, pages 218–237, 2002.

[26] Jichuan Wang, Haiyi Xie, and James H. Fisher. *Multilevel Models: Applications Using SAS*. Walter de Gruyter and Co., 2011.

[27] Heather Woltman, Andrea Feldstain, J. Christine Mackay, and Meredith Rocchi. An introduction to hierarchical modeling. *Tutorials in Quantitative Methods for Psychology*, 8(1):52–69, 2012.

[28] Bee Wah Yap and Chiaw Hock Sim. Comparisons of various types of normality tests. *Journal of Statistical Computation and Simulation*, 81(12):2141–2155, 2011.

7 Appendix

7.1 SAS code

7.1.1 Simulation study code

```
 ***Simulation****;
title 'Unconditional Means Model';
proc iml;
call randseed(123);
N=10000; ni=25;
Y=J(N,1,.); r=J(N,1,.);
Level1=J(N,1,.);Level2=J(N,1,.);
k=N/ni; uj=J(1,k,.);Bhatj=J(1,k,.);
***Parameters Chosen for the simulation***;
gamma=250; t00=100;
do j=1 to k;
  uj[,j]=randnormal(1,0,t00);
  r[ni*j-(ni-1):j*ni]=randfun(ni,'normal',0,5);
  Bhatj[,j]=gamma+uj[,j];
  Y[ni*j-(ni-1):j*ni]=J(ni,1,1)*Bhatj[,j]+r[ni*j-(ni-1):j*ni];
  Level1[ni*j-(ni-1):j*ni]=(1:ni)';
  Level2[ni*j-(ni-1):j*ni]=J(ni,1,j);
end;
D=Level2||Level1||Y;
create analysis.simulation1 from D[colname={Level2 Level1 Y}];
append from D;
quit;


title 'One-Way ANCOVA model';
proc iml;
call randseed(123);
N=10000;
ni=25;
Y=J(N,1,.); X=J(N,2,.); r=J(N,1,.);
Level1=J(N,1,.);Level2=J(N,1,.);
```

```
CX=J(N,1,.); k=N/ni; uj=J(2,k,.);Bhatj=J(2,k,.);Z=J(2,3,.);Z1=J(N,1,.);
***Parameters Chosen for the simulation***;
gamma={130,100,240}; t00=100;
do j=1 to k;
  uj[1:2,j]=(randnormal(1,0,t00))//0;
  r[ni*j-(ni-1):j*ni]=randfun(ni,'normal',0,16);
  X[ni*j-(ni-1):j*ni,1:2]=J(ni,1,1)||randfun(ni,'normal',20,6);
  Z[1,]=1||mean(X[ni*j-(ni-1):ni*j,2])||0;
  Z[2,]=0||0||1;
  Bhatj[1:2,j]=Z*gamma+uj[1:2,j];
  Y[ni*j-(ni-1):j*ni]=X[ni*j-(ni-1):j*ni,1:2]*Bhatj[1:2,j]+r[ni*j-(ni-1):j*ni];
  Level1[ni*j-(ni-1):j*ni]=(1:ni)`;
  Level2[ni*j-(ni-1):j*ni]=J(ni,1,j);
  Z1[ni*j-(ni-1):ni*j]=J(ni,1,mean(X[ni*j-(ni-1):ni*j,2]));
  ****Group Mean centering****;
 CX[ni*j-(ni-1):ni*j]=X[ni*j-(ni-1):ni*j,2]-Z1[ni*j-(ni-1):ni*j];
end;
D=Level2||Level1||Y||CX||Z1;
create analysis.simulation2 from D[colname={Level2 Level1 Y CX Z}];
append from D;
quit;

title 'Means as Outcomes model';
proc iml;
call randseed(123);
N=10000;
ni=25;
Y=J(N,1,.);Z=J(N,2,.);r=J(N,1,.);
Level1=J(N,1,.);Level2=J(N,1,.);
k=N/ni; uj=J(1,k,.);Bhatj=J(1,k,.);
***Parameters Chosen for the simulation***;
gamma={150,75}; t00=100;
do j=1 to k;
  uj[,j]=randnormal(1,0,t00);
```

```
  r[ni*j-(ni-1):j*ni]=randfun(ni,'normal',0,5);
  Z[ni*j-(ni-1):j*ni,1:2]=J(ni,1,1)||J(ni,1,randfun(1,'normal',15,5));
  Bhatj[,j]=Z[ni*j-(ni-1),1:2]*gamma+uj[,j];
  Y[ni*j-(ni-1):j*ni]=J(ni,1,1)*Bhatj[,j]+r[ni*j-(ni-1):j*ni];
 Level1[ni*j-(ni-1):j*ni]=(1:ni)‘;
  Level2[ni*j-(ni-1):j*ni]=J(ni,1,j);
end;
CZ=Z[,2]-J(N,1,mean(Z[,2]));
D=Level2||Level1||CZ||Z[,2]||Y;
create analysis.simulation3 from D[colname={Level2 Level1 CZ Z Y}];
append from D;
quit;

title 'Non-Varying slopes model';
proc iml;
call randseed(123);
N=10000;
ni=50;
Y=J(N,1,.); X=J(N,2,.); Z=J(N,2,.); r=J(N,1,.);
Level1=J(N,1,.);Level2=J(N,1,.);
CX=J(N,1,.);Z_=J(2,4,.); k=N/ni; uj=J(2,k,.);Bhatj=J(2,k,.);Z1=J(N,1,.);
***Parameters Chosen for the simulation***;
gamma={200,400,200,100}; t00=100;
do j=1 to k;
  uj[,j]=(randnormal(1,0,t00))//0;
  r[ni*j-(ni-1):j*ni]=randfun(ni,'normal',0,5);
  X[ni*j-(ni-1):j*ni,1:2]=J(ni,1,1)||randfun(ni,'normal',20,6);
  Z[ni*j-(ni-1):j*ni,1:2]=J(ni,1,1)||J(ni,1,randfun(1,'normal',15,5));
  Z_[1,]=1||mean(Z[ni*j-(ni-1):ni*j,2])||mean(X[ni*j-(ni-1):ni*j,2])||0;
  Z_[2,]={0 0 0}||1;
 Bhatj[1:2,j]=Z_*gamma+uj[,j];
  Y[ni*j-(ni-1):j*ni]=X[ni*j-(ni-1):j*ni,1:2]*Bhatj[1:2,j]+r[ni*j-(ni-1):j*ni];
  Level1[ni*j-(ni-1):j*ni]=(1:ni)‘;
  Level2[ni*j-(ni-1):j*ni]=J(ni,1,j);
```

```
  Z1[ni*j-(ni-1):ni*j]=J(ni,1,mean(X[ni*j-(ni-1):ni*j,2]));
  ****Group mean centering****;
 CX[ni*j-(ni-1):ni*j]=X[ni*j-(ni-1):ni*j,2]-Z1[ni*j-(ni-1):ni*j];
end;
****Grand Mean centered Level-1 and Level-2 predictors;
CZ=Z[,2]-J(N,1,mean(Z[,2]));
D=Level2||Level1||Y||CZ||Z1||CX;
create analysis.simulation4 from D[colname={Level2 Level1 Y CZ Z CX}];
append from D;
quit;
***Multilevel modeling***;
title 'Multilevel modeling';
title1 'Unconditional Means model';
proc mixed data=analysis.simulation1 noclprint noitprint covtest;
class level2;
model Y=/solution;
random intercept/sub=level2;
run;
title1 'One-Way ANCOVA model';
proc mixed data=analysis.simulation2 noclprint covtest noitprint;
class level2;
model Y=CX Z/solution;
random intercept/sub=level2 type=un;
run;
title1 'Means as outcomes model';
proc mixed data=analysis.simulation3 noitprint noclprint covtest;
class level2;
model Y=CZ/solution;
random intercept/sub=level2;
run;
title1 'Non varying slopes model';
proc mixed data=analysis.simulation4 noitprint noclprint covtest;
class level2;
model Y=CZ Z CX/solution;
```

```
random intercept/sub=level2 type=un;
run;
```

7.1.2 TIMSS Analysis code

1. **The Unconditional model**

```
proc mixed data=analysis.ensemble noclprint noitprint covtest;
class idschool;
model sciach=/solution;
random intercept/sub=idschool;
run;
```

2. **Grand mean centering level-1 predictors**

```
proc iml;
use analysis.ensemble;
read all var{idschool sciach scs her sats stub schslab schsrs schdp scheas} into xy;
N=nrow(xy);
***Student-level predictors****;
cscs=xy[,3]-mean(xy[,3]);
cher=xy[,4]-mean(xy[,4]);
csats=xy[,5]-mean(xy[,5]);
cstub=xy[,6]-mean(xy[,6]);
schslab=J(N,1,.);
***school-level predictors****;
do i=1 to N;
if xy[i,7]=1 then schslab[i]=1;
else schslab[i]=0;
end;
cschsrs=xy[,8]-mean(xy[,8]);
cschdp=xy[,9]-mean(xy[,9]);
cscheas=xy[,10]-mean(xy[,10]);
g=xy[,1]||xy[,2]||cscs||cher||csats||cstub||schslab||cschsrs||cschdp||cscheas;
var1={idschool sciach cscs cher csats cstub schslab cschsrs cschdp cscheas};
create analysis.levelg1 from g[colname=var1];
append from g;
```

```
quit;
```

3. **Level-1 predictors only model**

```
proc mixed data=analysis.levelg1 covtest noitprint;
class idschool;
model sciach=cscs cher csats cstub/solution;
random intercept cscs csats cstub cher/sub=idschool type=un;
run;
```

4. **Level-2 predictors only model**

```
proc mixed data=analysis.levelg1 covtest noitprint noclprint;
class idschool;
model sciach=schslab cschsrs cschdp cscheas/solution;
random intercept/sub=idschool;
run;
```

5. **Level-1 and Level-2 predictors model**

*******Random intercepts and slopes model*****

```
proc mixed data=analysis.levelg1 noclprint noitprint covtest;
class idschool;
model sciach=cscs csats cstub cher schslab cschsrs cschdp cscheas/solution;
random intercept cscs csats cstub cher/sub=idschool type=un; run;
```

```
*******Random intercepts only******
proc mixed data=analysis.levelg1 noclprint noitprint covtest;
class idschool;
model sciach=cscs csats cstub cher schslab cschsrs cschdp cscheas/solution;
random intercept/sub=idschool type=un;
run;
```

7.1.3 Comparative study

1. **Disaggregation**

```
proc reg data=analysis.levelg1;
```

```
model sciach=cscs cschsrs/vif;
output out=analysis.r_disagg r=residual_disaggregation;
run;

****Testing normality for the disaggregated regression***

symbol i=r;
proc univariate data=analysis.r_disagg normal;
qqplot;
var residual_disaggregation;
histogram/normal;
run;
```

2. **Aggregation**

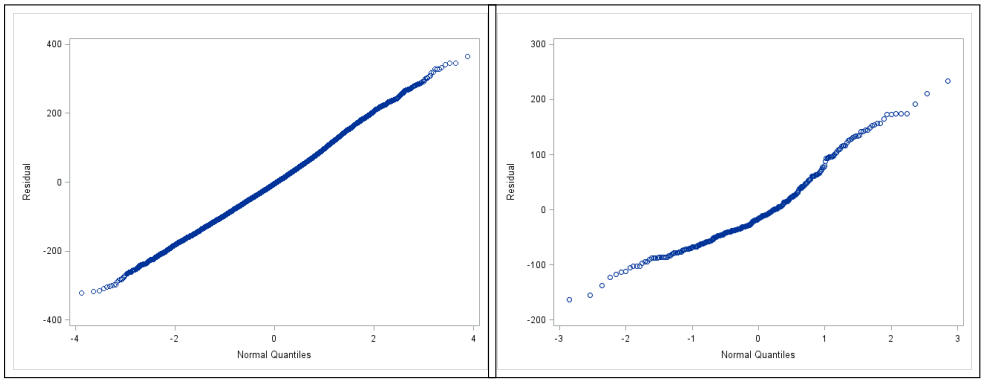
```
proc means data=analysis.levelg1;
class idschool;
var sciach;
output out=analysis.ca mean=msciach;
run;
proc means data=analysis.levelg1;
class idschool;
var cscs;
output out=analysis.cb mean=MSCS_C;
run;
proc means data=analysis.levelg1;
class idschool;
var cschsrs;
output out=analysis.cc mean=MSCHSRS_C;
run;
data analysis.d;
merge analysis.ca analysis.cb analysis.cc;
by idschool;
if idschool='.' then delete;
keep idschool msciach mscs_c mschsrs_c;
run;
```

```
proc reg data=analysis.d;
model msciach=mscs_c mschsrs_c/vif;
output out=analysis.r_agg r=residual_aggregation;
run;
***Testing normality for the aggragated regression***;
proc univariate data=analysis.r_agg normal plot;
qqplot;
var residual_aggregation;
run;
```

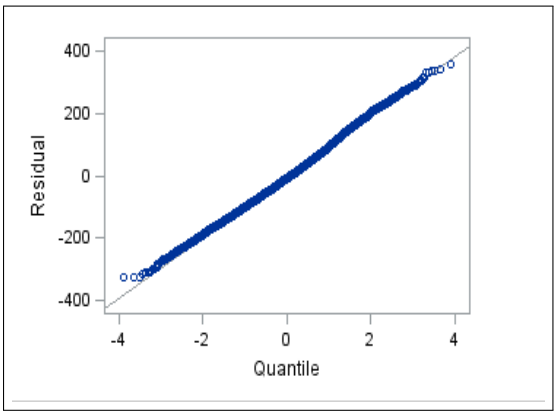
3. **Multilevel Modelling**

```
proc mixed data=analysis.levelg1 noclprint noitprint covtest plots(maxpoints=None);
class idschool;
model sciach=cscs cschsrs cscs*cschsrs/solution residual;
random intercept cscs/sub=idschool type=un;
run; \newline
```

7.2 Test of normality: QQ-plot



(a) Disaggregation (b) Aggregation



(c) Multilevel modeling

Figure 2: QQ-plot for the residuals: Normality test

Parameter estimation of Gaussian mixture models

Aaron Smith 12045129

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor : Mr. SM Millard

Department of Statistics, University of Pretoria

29 September 2017

Abstract

The presence of possible multimodality in data creates some challenges, in that a unimodal model will most likely be unable to model the full extent of the data. Mixture models have the ability to model data that is multimodal, and is therefore prevalent in modeling data of this nature. For simplicity, in this report the emphasis will be placed on a univariate two component mixture of Gaussian distributions to model the latent groups within the data. We will make use of two iterative procedures, namely the Expectation Maximization (EM) algorithm and a classification version of the Expectation Maximization (CEM) algorithm to estimate the parameters of the mixture model.

The two estimation procedures will be discussed for the univariate case as well as a brief overview of the multivariate case. Furthermore, an application of both algorithms will be conducted where both algorithms will be initialized by random selection as well as the K-means algorithm. The performance of the two methods of estimation will be compared based on a simulation study. Specific attention will be given to a comparison of the absolute bias of the estimated parameters as well as the efficiency of estimation.

Keywords: classification EM algorithm; EM algorithm; K-means; Maximum likelihood estimators; Mixture models; Simulation study.

Declaration

I, *Aaron Smith,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Aaron Smith

_ _

Mr. SM Millard

_ _

Date

Acknowledgments

I would firstly like to thank my supervisor, Mr. SM Millard, for challenging me on a week basis and pushing me to do my best. I would also like to thank you for making your time available to me and for sharing your knowledge in the countless meetings we've had over the past year. Furthermore, I acknowledge the financial support from the Center for AI Research, Meraka Institute, CSIR.

Contents

List of Figures

List of Algorithms

List of Tables

1 Introduction

In practice the presentation of data is hardly ever simplistic and may be presented as a high-dimensional data set [9]. A data set could possibly have multiple clusters of data, with each cluster having its own parametric distribution with its parameters differing from one cluster of data to the next [14]. Thus, the use of a single model presents a stumbling block due to the possible multimodality in a data set [8]. By multimodality, we mean that there is more than one region of high probability mass present in the data. The need for mixture models thus becomes ever more prevalent in the modeling of data, as the mixture model is able to model the full extent of the data. In this research report, we will focus on a univariate Gaussian mixture model as well as give a brief overview of the multivariate case.

The use of a mixture of Gaussian distributions necessitates the need for parameter estimation, which brings about one of the focal points of the research topic, the Expectation Maximization (EM) algorithm [5]. Over the last two decades many a researcher has employed the EM algorithm but it was Dempster (1977) [5] who first coined the term “EM” as well as proved the convergence of this algorithm. The algorithm is particularly useful for mixture model parameter estimation problems and has been found to have "reliable global convergence, low cost per iteration, economy of storage and ease of programming as well as a heuristic appeal" according to Li et al, (2005) [13]. The iterative procedure is used to estimate the unknown parameters of the mixture model, namely the mean, covariance and mixing coefficients. The EM algorithm according to Bishop, (2006) [3] is an "elegant and powerful method for finding maximum likelihood solutions for models with latent variables". The EM algorithm comprises of two steps, namely the Expectation step (E-step) and the Maximization step (M-step) which looks to approximate maximum likelihood estimates of parameters for incomplete data [1]. Firstly, we have the E-step which calculates the expectation of the log-likelihood function of the complete data conditional on the unobserved data, where the observed data and current estimates of the parameters are given. This leads to the next step, the M-step which maximizes the expected value of the complete data log-likelihood function found in the E-step to obtain new parameters [13]. The process alternates between the two steps until convergence has been achieved.

Once the EM algorithm has been thoroughly explained the study will be expanded by exploring a classification version of the Expectation Maximization (EM) algorithm, the classification EM (CEM) algorithm. The extension of the EM algorithm, which includes the so-called unobserved variable in a data set [14], consists of three steps as a posed to the two steps of the EM. The third step is the Classification step (C-step), which is added between the Expectation step and Maximization step [7]. This step assigns each observation to the component with the largest posterior probability [7]. This procedure is seen as

a K-means-like algorithm and according to Faria (2010) it "converges in a finite number of iterations" [7]. An application of the EM and CEM will be conducted, in which both algorithms will be initialized by random values and values generated by the K-means algorithm and the resulting log-likelihood functions will be compared. Furthermore, a simulation study will be conducted to monitor the performance of the EM and CEM respectively. This will be conducted by making use of a software program, namely SAS.

This research report is structured as follows: in Chapter 2, we present the formulation of the K-component Gaussian mixture model as well as the necessary background information and notation. From there, the maximum likelihood functions are derived and implemented in the EM and CEM algorithms respectively. The initial values that are needed for the EM and CEM algorithms will be generated by the K-means clustering algorithm, which will be covered and linked to the above mentioned algorithms. The EM and CEM algorithms for a univariate Gaussian mixture model will also be divulged in full as well as a brief overview of the multivariate cases. Chapter 3, an application of the two iterative procedures will be provided, where the log-likelihood functions of both algorithms with and without the K-means initialization will be monitored as well as a simulation study to monitor the performance of both algorithms (using randomly selected initial values) respectively. In chapter 4 conclusions of the study will be drawn and any further comments on the research report is given.

2 Theoretical Background

2.1 Mixture of normal distributions

Mixture models according to McLachlan and Peel, (2000) [15] "is the use of weighted sums of standard distributions" thus, it is concerned with the modeling of statistical distributions by a mixture or weighted sum. These models are also deemed to be semi-parametric, which means they consist of parametric and nonparametric components [10]. They are also known as “latent class models” or “unsupervised learning models”.

Data as we know it, is most often than not high-dimensional and possibly presents multimodality, which means there are several regions of high probability mass [8, 9]. Therefore, we cannot model a cluster of data points with a single distribution but rather a mixture. The most commonly used distribution in the EM algorithm would be that of the normal distribution with the t, Poisson and Gamma distributions being used from time to time. In this report we will focus our attention on the mixture of univariate K-component Gaussian distributions, with each component having a Gaussian density with their own mean μ_j and variance σ_j^2 [3]. Before we can make use of the mixture, we need to establish some notation based on Van Wyk, (2016) [23].

Let $\boldsymbol{y} = (y_1, y_2, ..., y_N)$ be an unlabeled sample of observations assumed to be independent and identically distributed, generated from a univariate K-component Gaussian mixture defined on $\Re$. The mixture probability density function of a specific observation, y_i can be denoted as follows

$$p\left(y_i|\boldsymbol{\Theta}\right) = \sum_{j=1}^{K} p\left(y_i|\theta_j\right)\omega_j \tag{1}$$

where ω_j are the mixing coefficients of the distribution ($0 < \omega_j < 1$ and $\sum_{j=1}^{K}\omega_j = 1$),
$\boldsymbol{\Theta} = (\theta_1, ..., \theta_K, \omega_1, ..., \omega_K)$ denotes the complete set of parameters that specify the mixture with $\theta_j = \left\{\mu_j, \sigma_j^2\right\}$ and $\boldsymbol{\eta} = (\theta_1, \theta_2, ..., \theta_K)$ representing the parameters of each component density.

Since the individual observations are independent and identically distributed ($i.i.d$), i.e. $Y_i \sim N\left(\mu_j, \sigma_j^2\right)$ for $i = 1, 2, ..., N$ and $j = 1, 2, ..., K$, the Gaussian component-conditional probability density function has the form:

$$\begin{aligned} p\left(y_i|\theta_j\right) &= p\left(y_i|\mu_j, \sigma_j^2\right) \\ &= \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left(-\frac{\left(y_i - \mu_j\right)^2}{2\sigma_j}\right) \end{aligned} \tag{2}$$

We now have the necessary information to start the process of estimating the parameters by means of the EM algorithm. Before we can start, it is important to note that for simplicity we have specified the number of components of the mixture model. Therefore, the number of components will not have to be estimated, but for interest sake, these can be estimated using Bayes Information Criterion (BIC) and Akaike Information Criterion (AIC) [10].

However, our main focus in this report is the estimation of the unknown parameters and mixing coefficients. This can be achieved by using maximum likelihood as well as corresponding Bayesian approach, with the former being discussed in detail at a later stage.

2.2 Maximum likelihood estimation

In data analytics and statistical theory, maximum likelihood estimation and likelihood-based inference is of high importance. Maximum likelihood (ML) estimation can be described as a general estimation method that is extensively used in areas where statistical techniques are used [16]. ML estimation according to Moon, (1996) [18] "is a means of estimating the parameters of a distribution based upon the observed data drawn according to that distribution". We will now establish some notation and give an overview of ML estimation.

Following on from section 2.1, the mixture probability density function of a specific observation y_i is denoted by

$$p\left(y_i|\boldsymbol{\Theta}\right) = \sum\nolimits_{j=1}^{K} p\left(y_i|\theta_j\right)\omega_j$$

Since the parameters are estimated by means of maximum likelihood, we first need to denote the likelihood function

$$\begin{aligned} p\left(\boldsymbol{y}|\boldsymbol{\Theta}\right) &= \prod_{i=1}^{N} p\left(y_i|\boldsymbol{\Theta}\right) \\ &= \prod_{i=1}^{N} \left(\sum_{j=1}^{K} p\left(y_i|\theta_j\right)\omega_j \right) \end{aligned} \tag{3}$$

This function differs from that of the probability density function as the sample $\boldsymbol{y}$ is fixed where as with the probability density function, the parameter $\boldsymbol{\Theta}$ is fixed.

Since the main concept of maximum likelihood estimation is to produce a value for the estimator $\hat{\boldsymbol{\Theta}}$ that ensures the observed data is high as possible. The value of the parameters which maximize the likelihood function i.e. ML estimate of the parameter, is illustrated as follows

$$\hat{\boldsymbol{\Theta}}_{ML} = \arg\max_{\boldsymbol{\Theta}} p\left(\boldsymbol{y}|\boldsymbol{\Theta}\right) \tag{4}$$

One needs to note that the log-likelihood function is more convenient to use than the conventional likelihood function [8]. According to Moon, (1996) [18], the logarithm function is "monotonically increasing" therefore, the maximization of the logarithm function is equivalent to that of the likelihood function**.** This can be defined as follows

$$\begin{aligned} \log p\left(\boldsymbol{y}|\boldsymbol{\Theta}\right) &= \log \prod_{i=1}^{N} p\left(y_i|\boldsymbol{\Theta}\right) \\ &= \sum_{i=1}^{N} \log p\left(y_i|\boldsymbol{\Theta}\right) \\ &= \sum_{i=1}^{N} \log \left(\sum_{j=1}^{K} p\left(y_i|\theta_j\right)\omega_j \right) \end{aligned} \tag{5}$$

The main aim of ML estimation is for the log-likelihood function to be maximized with regards to $\boldsymbol{\Theta}$, which implies one would need to set the derivatives of the equation 6 mentioned below equal to zero [16].

$$\frac{\partial}{\partial \boldsymbol{\Theta}} \log p\left(\boldsymbol{y}|\boldsymbol{\Theta}\right) |_{\boldsymbol{\Theta}=\boldsymbol{\Theta}_{ML}} = 0 \tag{6}$$

However, literature shows that log-likelihood functions are somewhat difficult to optimize, due to the sum of terms in the logarithm [8]. Since the component labels of the data are unknown, we can refer to this problem as a missing-data problem which can be solved by the EM algorithm [4].

Let $\boldsymbol{y} = (y_1, y_2, ..., y_N)\, \epsilon \Re$ be a sample of observed data, which is viewed as "incomplete" since the component labels of the data, denoted by $\boldsymbol{\gamma} = \{\gamma_1, \gamma_2, ..., \gamma_N\}$ are unknown [23]. These component labels, also known as responsibilities, are each equal to one of the values in the set $\{1, 2, ..., K\}$, depending on whether the j^{th} component produced the i^{th} observation.
Therefore, the responsibilities can be seen as a binary vector denoted by

$$\boldsymbol{\gamma_i} = (\gamma_{i1}, \gamma_{i2}, ..., \gamma_{iK})$$

where

$$\begin{cases} \gamma_{ij} = 1 \\ \\ \gamma_{im} = 0 \end{cases} \quad for\, j \neq m$$

if observation y_i was produced by the j^{th}component [23, 3].
We therefore can conclude that, γ_i has a multinomial distribution with parameters 1 and $\boldsymbol{\pi}$ i.e. $\boldsymbol{\gamma_i} \sim Mult\,(1, \boldsymbol{\pi})$ where $\boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_K)$. Therefore, the observed data and corresponding component labels of the data form the complete data likelihood function as follows

$$p\,(\boldsymbol{y}, \boldsymbol{\gamma}|\boldsymbol{\Theta}) = \prod_{i=1}^{N} \prod_{j=1}^{K} \left(p\,(y_i|\theta_j)\,\omega_j\right)^{I\{\gamma_i = j\}}$$

where

$$I\,\{\gamma_i = j\} = \gamma_{ij} = \begin{cases} 1 & if\, \gamma_i = j \\ \\ 0 & otherwise \end{cases}$$

which is the binary indicator variable that represents the unobserved data.
The complete data log-likelihood function can further be written as:

$$\begin{aligned} \log p\,(\boldsymbol{y}, \boldsymbol{\gamma}|\boldsymbol{\Theta}) &= \sum_{i=1}^{N} \sum_{j=1}^{K} \gamma_{ij} \log\left(\omega_j p\,(y_i|\theta_j)\right) \\ &= \sum_{i=1}^{N} \sum_{j=1}^{K} \gamma_{ij} \left(\log \omega_j + \log p\,(x_i|\theta_j)\right) \end{aligned} \tag{7}$$

2.3 The K-means clustering algorithm

2.3.1 Introduction

The initialization of the EM algorithm is of great importance, as it heavily affects the speed of convergence as well as its ability to locate global maxima [12]. In literature, the initial values used in the EM algorithm are usually "guesses", which creates the possibility of non-convergence of the algorithm. Therefore, the natural choice would be to start with estimates obtained by other unsupervised learning methods, such as the K-means algorithm. In the following section, a brief overview will be given as well as how the algorithm can be used to initialize the EM algorithm.

2.3.2 The K-means clustering algorithm

The K-means algorithm is a type of partitional clustering which looks to classify data into clusters and to find their corresponding cluster centers [8]. It is often used in initializing Gaussian mixtures and Learning Vector Quantization (LVQ), that are deemed to be more computationally expensive algorithms. Suppose that $\boldsymbol{y} = (y_1, y_2, ..., y_N) \in \Re$ is a sample of N observations, which are assigned to K clusters, which we will assume to be known for simplicity [23]. The main idea of K-means clustering according to Melnykov (2012) [17] "is to partition observations so that the within-cluster sum of squares is minimized".

Firstly, we need to calculate the distance between the "new" and "old" centers of the clusters. Note that in this context the "new" center refers to the newly updated center of the cluster and the "old" center refers to the initial center before an iteration. This would require a dissimilarity measure, namely the squared Euclidean distance, also known as a *distortion measure*. The formula for dissimilarity is given by [8]

$$d(y_i, y_{i'}) = ||y_i - y_{i'}||^2 \tag{8}$$

where y_i and $y_{i'}$ denote the "new" and "old" cluster centers respectively.

Suppose that $K < N$ is a fixed number of clusters, where $y_1, y_2, ..., y_N$ is a cluster of points that are a function of C that assigns each observation y_i to a cluster $j \in \{1, 2, ..., K\}$. These are characterized by an encoder $C(i) = j$ which indicates that observation y_i has been assigned to the j^{th} cluster. The encoder function can be defined as follows

$$I\{C(i) = j\} = \begin{cases} 1 & if\, C(i) = j \\ 0 & otherwise \end{cases}$$

The natural loss function, which is referred to as the "within-cluster" scatter is expressed as follows in terms of the squared Euclidean distance [8]

$$W(C) = \frac{1}{2}\sum_{j=1}^{K}\sum_{C(i)=j}\sum_{C(i')=j} d(y_i, y_{i'})$$
$$= \sum_{j=1}^{K} N_j \sum_{C(i)=j} ||y_i - \overline{y}_j||^2$$

where $N_j = \sum_{i=1}^{N} I\{C(i) = j\}$ is the number of observations and $\overline{y}_j = (y_{1j}, y_{2j}, ..., y_{pj})$ is the means of the observations in the j^{th} cluster.

Hence, K-means clustering according to Hastie et al. (2001) "is an iterative descent algorithm" [8] aimed at minimizing the natural loss function as follows

$$C^* = \min_{C} \sum_{j=1}^{K} N_j \sum_{C(i)=j} ||y_i - \overline{y}_j||^2$$

The K-means clustering algorithm is summarized in Algorithm 1 based on Hastie et al. (2001) [8].

Algorithm 1 The K-means clustering algorithm.

1) The total "within-cluster" variance is minimized over a cluster assignment C

$$\min_{C, \{c_j\}_1^K} \sum_{j=1}^{K} N_j \sum_{C(i)=j} \|y_i - c_j\|^2 \tag{9}$$

with regards to $\{c_1, c_2, ..., c_K\}$ which produces the average of the data points for the clusters currently assigned

$$\overline{y}_J = \arg\min_{m} \sum_{i \epsilon J} \|y_i - c\|^2$$

where J is any set of observations.

2) Equation (9) is minimized over the current cluster means $\{c_1, c_2, ..., c_K\}$, by finding the current means closest to each observation. That is

$$C(i) = \arg\min_{1 \leq j \leq K} \|y_i - c_j\|^2, \;\; i = 1, ..., N$$

which represents the cluster assignment for i^{th} observation.

3) Alternate between step 1 and 2 until convergence

Once the K-means clustering algorithm has been applied, the cluster centers are used as initial values for the mean of the univariate Gaussian mixture model. Furthermore, these cluster centers are used to calculate the initial values for the variance and mixing coefficients.

The K-means clustering algorithm has the advantage that no matter the choice of the initial cluster center, the algorithm will converge. However, the algorithm does have its drawbacks in that in certain

situations it does not produce an initial parameter vector that leads to the correct solutions. According to Melnykov (2012):

"*A major drawback of this method is that by construction it is designed to work well for spherical well-separated clusters of similar representation. If clusters are elongated, have different sizes, or suffer considerable overlaps K-means can face challenges.*"

2.4 The EM algorithm

2.4.1 Introduction

The Expectation Maximization (EM) algorithm, which is known as the likelihood maximizer, is a popular method to obtain the ML estimators, $\hat{\boldsymbol{\Theta}}$ from incomplete data [5]. This iterative method, which was first introduced by Dempster et al. (1977) [5] in 1977 consists of two steps, namely the Expectation (E-step) and Maximization steps (M-step). The E-step calculates the expectation of the log-likelihood function of the complete data conditional on the unobserved data, where the observed data $\boldsymbol{y}$ and current estimates of the parameters are given. This function is also referred to as the objective function or Q function [23]. The M-step computes the parameter estimates, $\hat{\boldsymbol{\Theta}}$ that maximize the expected log-likelihood function found in the E-step [18, 7]. The process alternates between the two steps until convergence has been achieved [23].

In the following sections, the E and M steps for a univariate K-component Gaussian mixture model will be derived as well as a brief overview of the multivariate case.

2.4.2 The EM algorithm for univariate Gaussian mixture models

We now apply the principles mentioned above as well as the notation that was established in the mixture of normal distributions and maximum likelihood estimation sections for a univariate K-component Gaussian mixture model to derive expressions for the E-step and the M-step, respectively.

E-step:

For the E-step, at $t \geq 0$ we compute the expectation of the log-likelihood function of the complete data conditional on the unobserved data, assuming that initial values for the parameters are given as well as the observed data $\boldsymbol{y}$ at $t = 0$. This function is referred to as the Q function and is expressed as follows:

$$Q\left(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}(t)\right) = E\left[\log p\left(\boldsymbol{y}, \boldsymbol{\gamma} | \boldsymbol{\Theta}\right) | \boldsymbol{y}, \hat{\boldsymbol{\Theta}}(t)\right]$$

where $\hat{\boldsymbol{\Theta}}(t)$ denotes the maximum likelihood estimates at time step t. Due to the linearity of the log-likelihood function of the complete data, we can simplify the E-step by calculating the expectation of

the binary indicator variable γ_{ij}, $i = 1, 2, ..., N$; $j = 1, 2, ..., K$ where the observed data $\boldsymbol{y}$ and current estimates of the parameters are given, as follows:

$$\begin{aligned} E\left[\gamma_{ij}|\boldsymbol{y},\hat{\boldsymbol{\Theta}}(t)\right] &= 1 \times P\left(\gamma_{ij} = 1|\boldsymbol{y},\hat{\boldsymbol{\Theta}}(t)\right) + 0 \times P\left(\gamma_{ij} = 0|\boldsymbol{y},\hat{\boldsymbol{\Theta}}(t)\right) \\ &= P\left(\gamma_{ij} = 1|\boldsymbol{y},\hat{\boldsymbol{\Theta}}(t)\right) \\ &= P\left(\gamma_i = j|y_i,\hat{\boldsymbol{\Theta}}(t)\right) \quad since\, \gamma_{ij} = 1 \iff \gamma_i = j \end{aligned} \tag{10}$$

which is the probability that the y_i observation was produced by the j^{th} component. Thus, using Bayes' Theorem (1) on equation 10, the responsibility of the j^{th} component for the i^{th} observation is

$$\begin{aligned} \hat{\gamma_{ij}}(t) &= P\left(\gamma_i = j|\hat{\boldsymbol{\Theta}}(t)\right) \\ &= \frac{p\left(y_i|\hat{\theta}_j(t)\right)\hat{\omega}_j(t)}{\sum_{k=1}^{K} p\left(y_i|\hat{\theta}_k(t)\right)\hat{\omega}_k(t)} \\ &= \frac{\left[\frac{1}{\sqrt{2\pi\hat{\sigma}_j(t)}}\exp\left(-\frac{1}{2\hat{\sigma}_j^2(t)}(y_i - \hat{\mu}_j(t))^2\right)\right]\hat{\omega}_j(t)}{\sum_{k=1}^{K}\left[\frac{1}{\sqrt{2\pi\hat{\sigma}_k^2(t)}}\exp\left(-\frac{1}{2\hat{\sigma}_k^2(t)}(y_i - \hat{\mu}_k(t))^2\right)\right]\hat{\omega}_k(t)} \end{aligned} \tag{11}$$

for $i = 1, 2, ..., N; j = 1, 2, ..., J$ and $t \geq 0$

Therefore, in the E-step each observation is assigned to each component, making use of the current parameter estimates to calculate the relative densities of the observations under each component model [8]. Therefore, the Q function is as follows:

$$\begin{aligned} Q\left(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}(t)\right) &= E\left[\log p\left(\boldsymbol{y}, \boldsymbol{\gamma}|\boldsymbol{\Theta}\right)|\boldsymbol{x}, \hat{\boldsymbol{\Theta}}(t)\right] \\ &= \sum_{j=1}^{K}\sum_{i=1}^{N} \hat{\gamma}_{ij}(t)\left(\log p\left(y_i|\hat{\boldsymbol{\Theta}}_j(t)\right) + log\hat{\omega}_j(t)\right) \\ &= \sum_{j=1}^{K}\sum_{i=1}^{N} P\left(\gamma_i = j|y_i,\hat{\boldsymbol{\Theta}}(t)\right)\left(\log p\left(y_i|\hat{\boldsymbol{\Theta}}_j(t)\right) + \log\hat{\omega}_j(t)\right) \\ &= \sum_{j=1}^{K}\sum_{i=1}^{N} \log\hat{\omega}_j(t) P\left(\gamma_i = j|y_i,\hat{\boldsymbol{\Theta}}(t)\right) + \sum_{j=1}^{K}\sum_{i=1}^{N} \log p\left(y_i|\hat{\boldsymbol{\Theta}}_j(t)\right) P\left(\gamma_i = j|y_i,\hat{\boldsymbol{\Theta}}(t)\right) \end{aligned} \tag{12}$$

M-step:

For this step we maximize the Q function found in the E-step in terms of the unknown parameters $\boldsymbol{\Theta}$ to

obtain the new estimates $\hat{\boldsymbol{\Theta}}(t+1)$. These new estimates which maximize the Q function are expressed as follows:

$$\hat{\boldsymbol{\Theta}}\left(t+1\right) = \arg\max_{\boldsymbol{\Theta}} Q\left(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}(t)\right)$$

For the Q function, equation 12, we see that since the terms containing ω_j and θ_j are independent, these terms can be independently maximized with respect to the parameters of each component density and the mixing coefficients [2]. Therefore, the updated estimates will be calculated independently for the mixing coefficients $\hat{\omega}_j(t+1)$, $j = 1, 2, ..., K$ and the parameters $\hat{\boldsymbol{\eta}}(t+1)$, with $\boldsymbol{\eta} = (\theta_1, \theta_2, ..., \theta_K)$.

Due to the fact that the component labels of the data are unknown, estimates for the mixing coefficients need to be iteratively approximated. Therefore, by adhering to the following constraint $\sum_{j=1}^{K} \hat{\omega}_j(t) = 1$ and making use of equation 11, an expression for $\hat{\omega}_j(t+1)$ can be acquired by setting the following equation equal to zero [23]:

$$\frac{\partial Q\left(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}(t)\right)}{\partial \hat{\omega}_j(t)} = 0$$

This is achieved by making use of the Lagrange multiplier (2) λ with the above mentioned constraint as follows [2]:

$$\frac{\partial}{\partial \hat{\omega}_j(t)} \left[\sum_{j=1}^{K} \sum_{i=1}^{N} \hat{\gamma_{ij}}(t) \left(\log p\left(y_i | \hat{\theta}_j(t)\right) + \log \hat{\omega}_j(t) \right) + \lambda \left(\sum_{j=1}^{K} \hat{\omega}_j(t) - 1 \right) \right] = 0$$

$$\therefore \sum_{i=1}^{N} \frac{1}{\hat{\omega}_j(t)} \hat{\gamma}_{ij}(t) + \lambda = 0$$

By summing the whole equation up over j, we therefore get that $\lambda = -N$, which results in

$$\hat{\omega}_j(t) = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ij}(t)}{N}, \;\; j = 1, 2, ..., K$$

Therefore, at time step $t+1$ the estimates of the mixing coefficients are as follows:

$$\begin{aligned} \hat{\omega}_j(t+1) &= \sum_{i=1}^{N} \frac{\hat{\gamma_{ij}}(t)}{N} \\ &= \sum_{i=1}^{N} \frac{P\left(\gamma_i = j | y_i, \hat{\boldsymbol{\Theta}}(t)\right)}{N}, j = 1, 2, ..., J \; and \; t \geq 0 \end{aligned} \tag{13}$$

which is achieved by equation 11 from the E-step.

By setting the partial derivatives equal to zero, we are able to find an expression for $\hat{\boldsymbol{\eta}}(t+1)$ where $\boldsymbol{\eta} = (\theta_1, \theta_2, ..., \theta_K) = \left(\left(\mu_1, \sigma_1^2\right), \left(\mu_2, \sigma_2^2\right), ..., \left(\mu_K, \sigma_K^2\right)\right)$ as follows:

$$\begin{aligned}
\frac{\partial Q\left(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}(t)\right)}{\partial \hat{\boldsymbol{\eta}}(t)} &= \frac{\partial}{\partial \hat{\boldsymbol{\eta}}(t)} \left[\sum_{j=1}^{K}\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) \left(\log p\left(y_i|\hat{\theta}_j(t)\right) + \log \hat{\omega}_j(t)\right)\right] \\
&= \sum_{j=1}^{J}\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) \frac{\partial \log p\left(y_i|\hat{\theta}_j(t)\right)}{\partial \hat{\epsilon}(t)} \\
&= 0
\end{aligned} \tag{14}$$

However, the Gaussian component-conditional probability density function is required before the roots can be found, which is defined as:

$$\begin{aligned}
\log p\left(y_i|\hat{\theta}_j(t)\right) &= \log p\left(y_i|\hat{\mu}_j(t), \hat{\sigma}_j^2(t)\right) \\
&= \log\left[\frac{1}{\hat{\sigma}_j(t)\sqrt{2\pi}} \exp\left(-\frac{\left(y_i - \hat{\mu}_j(t)\right)^2}{2\hat{\sigma}_j^2(t)}\right)\right] \\
&= \log\left(\frac{1}{\sqrt{2\pi}}\right) + \log\left(\frac{1}{\hat{\sigma}_j(t)}\right) - \frac{\left(y_i - \hat{\mu}_j(t)\right)^2}{2\hat{\sigma}_j^2(t)}
\end{aligned} \tag{15}$$

Taking the partial derivatives of equation 15, with regards to $\hat{\mu}_j(t)$ and $\hat{\sigma}_j^2(t)$, we are given the following

$$\frac{\partial \log p\left(y_i|\hat{\theta}_j(t)\right)}{\partial \hat{\mu}_j(t)} = \frac{\left(y_i - \hat{\mu}_j(t)\right)}{\hat{\sigma}_j^2(t)} \tag{16}$$

and

$$\frac{\partial \log p\left(y_i|\hat{\theta}_j(t)\right)}{\partial \hat{\sigma}_j^2(t)} = -\frac{1}{2\hat{\sigma}_j^2(t)} + \frac{\left(y_i - \hat{\mu}_j(t)\right)^2}{2\hat{\sigma}_j^4(t)} \tag{17}$$

Using equation 15, the Q function, equation 14 can be written as:

$$\begin{aligned}
\frac{\partial Q\left(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}(t)\right)}{\partial \hat{\boldsymbol{\eta}}(t)} &= \sum_{j=1}^{J}\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) \frac{\partial \log p\left(y_i|\hat{\theta}_j(t)\right)}{\partial \hat{\boldsymbol{\eta}}(t)} \\
&= \sum_{j=1}^{J}\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) \frac{\partial}{\partial \hat{\boldsymbol{\eta}}(t)} \left[\log\left(\frac{1}{\sqrt{2\pi}}\right) + \log\left(\frac{1}{\hat{\sigma}_j(t)}\right) - \frac{\left(y_i - \hat{\mu}_j(t)\right)^2}{2\hat{\sigma}_j^2(t)}\right] \\
&= 0
\end{aligned} \tag{18}$$

Equation 18 results in the following equations:

$$\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) \frac{\partial}{\partial \hat{\mu}_j(t)} \left[\log\left(\frac{1}{\sqrt{2\pi}}\right) + \log\left(\frac{1}{\hat{\sigma}_j(t)}\right) - \frac{\left(y_i - \hat{\mu}_j(t)\right)^2}{2\hat{\sigma}_j^2(t)}\right] = 0 \tag{19}$$

and

$$\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) \frac{\partial}{\partial \hat{\sigma}_j^2(t)} \left[\log\left(\frac{1}{\sqrt{2\pi}}\right) + \log\left(\frac{1}{\hat{\sigma}_j(t)}\right) - \frac{(y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^2(t)} \right] = 0 \tag{20}$$

The solutions of equations 19 and 20 can be obtained by using the results of equations 17 and 18 as follows:

Solving equation 19:

$$\begin{aligned}
\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) \frac{\partial}{\partial \hat{\mu}_j(t)} \left[\log\left(\frac{1}{\sqrt{2\pi}}\right) + \log\left(\frac{1}{\hat{\sigma}_j(t)}\right) - \frac{(y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^2(t)} \right] &= 0 \\
\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) \frac{(y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^2(t)} &= 0 \\
\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) y_i - \sum_{i=1}^{N} \hat{\gamma}_{ij}(t) \hat{\mu}_j(t) &= 0 \\
\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) y_i &= \sum_{i=1}^{N} \hat{\gamma}_{ij}(t) \hat{\mu}_j(t) \\
\hat{\mu}_j(t) &= \frac{\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) y_i}{\sum_{i=1}^{N} \hat{\gamma}_{ij}(t)}
\end{aligned} \tag{21}$$

Solving equation 20:

$$\begin{aligned}
\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) \frac{\partial}{\partial \hat{\sigma}_j^2(t)} \left[\log\left(\frac{1}{\sqrt{2\pi}}\right) + \log\left(\frac{1}{\hat{\sigma}_j(t)}\right) - \frac{(y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^2(t)} \right] &= 0 \\
\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) \left[-\frac{1}{2\hat{\sigma}_j^2(t)} + \frac{(y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^4(t)} \right] &= 0 \\
-\frac{\sum_{i=1}^{N} \hat{\gamma}_{ij}(t)}{2\hat{\sigma}_j^2(t)} + \frac{\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) (y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^4(t)} &= 0 \\
-\frac{\sum_{i=1}^{N} \hat{\gamma}_{ij}(t)}{2\hat{\sigma}_j^2(t)} &= -\frac{\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) (y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^4(t)} \\
\hat{\sigma}_j^2(t) &= \frac{\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) (y_i - \hat{\mu}_j(t))^2}{\sum_{i=1}^{N} \hat{\gamma}_{ij}(t)}
\end{aligned} \tag{22}$$

Therefore, at time $t+1$ the component parameter updates which enable equations 21 and 22 to be used in an iterative procedure are defined as follows

$$\hat{\mu}_j(t+1) = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) y_i}{\sum_{i=1}^{N} \hat{\gamma}_{ij}(t)}, j = 1, 2, ..., J \, and \, t \geq 0 \tag{23}$$

and

$$\hat{\sigma}_j^2(t+1) = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ij}(t) (y_i - \hat{\mu}_j(t))^2}{\sum_{i=1}^{N} \hat{\gamma}_{ij}(t)}, j = 1, 2, ..., J \, and \, t \geq 0 \tag{24}$$

The EM algorithm for a univariate K-component Gaussian mixture model is summarized in Algorithm 2 based on Hastie et al. (2001) [8].

Algorithm 2 The EM Algorithm for the univariate case.

1) Algorithm starts by taking initial values generated by the K-means algorithm for the unknown parameters $\hat{\mu}_j(0)$, $\hat{\sigma}_j^2(0)$ and $\hat{\omega}_j(0)$ for $j = 1, 2, ..., K$.

2) Expectation Step: Compute the responsibilities

$$\begin{aligned}\hat{\gamma_{ij}}(t) &= \frac{p\left(y_i|\hat{\theta}_j(t)\right)\hat{\omega}_j(t)}{\sum_{k=1}^{K} p\left(y_i|\hat{\theta}_k(t)\right)\hat{\omega}_k(t)} \\ &= \frac{\left[\frac{1}{\sqrt{2\pi\hat{\sigma}_j(t)}}\exp\left(-\frac{1}{2\hat{\sigma}_j^2(t)}\left(y_i-\hat{\mu}_j(t)\right)^2\right)\right]\hat{\omega}_j(t)}{\sum_{k=1}^{K}\left[\frac{1}{\sqrt{2\pi\hat{\sigma}_k(t)}}\exp\left(-\frac{1}{2\hat{\sigma}_k^2(t)}\left(y_i-\hat{\mu}_k(t)\right)^2\right)\right]\hat{\omega}_k(t)}\end{aligned}$$

with $\theta_j = \left(\mu_j, \sigma_j^2\right)$ for $i = 1, 2, ..., N$ and $j = 1, 2, ..., K$

3) Maximization Step: Determine the maximum-likelihood estimators of the unknown parameters as follows

$$\hat{\omega}_j(t+1) = \sum_{i=1}^{N}\frac{\hat{\gamma_{ij}}(t)}{N}, j = 1, 2, ..., K$$

and

$$\hat{\mu}_j(t+1) = \frac{\sum_{i=1}^{N}\hat{\gamma}_{ij}(t)y_i}{\sum_{i=1}^{N}\hat{\gamma}_{ij}(t)}, j = 1, 2, ..., K$$

and

$$\hat{\sigma}_j^2(t) = \frac{\sum_{i=1}^{N}\hat{\gamma}_{ij}(t)\left(y_i-\hat{\mu}_j(t)\right)^2}{\sum_{i=1}^{N}\hat{\gamma}_{ij}(t)}, j = 1, 2, ..., K$$

4) Alternate between steps 2 and 3 until convergence

2.4.3 The EM algorithm for multivariate Gaussian mixture models

The multivariate K-component Gaussian mixture model has a similar structure to that of the univariate case, with a component-conditional probability density function of the form

$$p\left(\boldsymbol{y}|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}\right) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma_j}|^{1/2}}\exp\left(-\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{\mu_j}\right)^T\boldsymbol{\Sigma_j^{-1}}\left(\boldsymbol{y}-\boldsymbol{\mu_j}\right)\right)$$

where $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N) \in \Re^p$ is an unlabeled sample of observations assumed to be *i.i.d.*

As with the univariate case, the EM algorithm for a multivariate K-component Gaussian mixture model is summarized in Algorithm 3 based on Hastie et al. (2001) [8].

Algorithm 3 The EM Algorithm for the multivariate case.

1) Algorithm starts by taking initial values generated by the K-means algorithm for the unknown parameters $\hat{\boldsymbol{\mu}}_{\boldsymbol{j}}(0)$,$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{j}}(0)$ and $\hat{\omega}_j(0)$ for $j = 1, 2, ..., K$.

2) Expectation Step: Compute the responsibilities

$$\begin{aligned}\hat{\gamma_{ij}}(t) &= \frac{p\left(\boldsymbol{y}_i|\hat{\boldsymbol{\theta}}_{\boldsymbol{j}}\right)\hat{\omega}_j(t)}{\sum_{k=1}^{K} p\left(\boldsymbol{y}_i|\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}\right)\hat{\omega}_k(t)} \\ &= \frac{\left[\frac{1}{(2\pi)^{p/2}|\hat{\boldsymbol{\Sigma}}_{\boldsymbol{j}}(t)|}\exp\left(-\frac{1}{2}\left(\boldsymbol{y_i}-\hat{\boldsymbol{\mu}}_{\boldsymbol{j}}(t)\right)^T\boldsymbol{\Sigma_j^{-1}}(t)\left(\boldsymbol{y_i}-\hat{\boldsymbol{\mu}}_{\boldsymbol{j}}(t)\right)\right)\right]\hat{\omega}_j(t)}{\sum_{k=1}^{K}\left[\frac{1}{(2\pi)^{p/2}|\hat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}(t)}\exp\left(-\frac{1}{2}\left(\boldsymbol{y_i}-\hat{\boldsymbol{\mu}}_{\boldsymbol{k}}(t)\right)^T\boldsymbol{\Sigma_k^{-1}}(t)\left(\boldsymbol{y_i}-\hat{\boldsymbol{\mu}}_{\boldsymbol{k}}(t)\right)\right)\right]\hat{\omega}_k(t)}\end{aligned}$$

with $\boldsymbol{\theta_j} = \left(\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}\right)$ for $i = 1, 2, ..., N$ and $j = 1, 2, ..., K$

3) Maximization Step: Determine the maximum-likelihood estimators of the unknown parameters as follows

$$\hat{\omega}_j(t+1) = \sum_{i=1}^{N}\frac{\hat{\gamma_{ij}}(t)}{N}, j = 1, 2, ..., K$$

and

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{j}}(t+1) = \frac{\sum_{i=1}^{N}\hat{\gamma}_{ij}(t)\boldsymbol{y_i}}{\sum_{i=1}^{N}\hat{\gamma}_{ij}(t)}, j = 1, 2, ..., K$$

and

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{j}}(t+1) = \frac{\sum_{i=1}^{N}\hat{\gamma}_{ij}(t)\left(\boldsymbol{y_i}-\hat{\boldsymbol{\mu}}_{\boldsymbol{j}}(t)\right)\left(\boldsymbol{y_i}-\hat{\boldsymbol{\mu}}_{\boldsymbol{j}}(t)\right)^T}{\sum_{i=1}^{N}\hat{\gamma}_{ij}(t)}, j = 1, 2, ..., K$$

4) Alternate between steps 2 and 3 until convergence

2.5 The Classification EM algorithm

2.5.1 Introduction

The Classification Expectation Maximization algorithm (CEM) is a classification version of the EM algorithm that according to Faria, (2010) "converges in a finite number of iterations" [7]. The procedure incorporates a Classification step (or C-step) between the E and M step of the EM algorithm, with the C-step assigning each observation to the component with the largest γ_{ij} [7]. The E step of the CEM algorithm is identical to that of the EM algorithm, with the process iterating until a specific convergence criterion is met. The CEM algorithm, which converges faster than the EM algorithm, is a "hard" assignment and thus is seen as a K-means-like algorithm [21]. In the following section, the E, C and M steps for a univariate K-component Gaussian mixture model will be derived as well as a brief overview of a multivariate case will be given.

2.5.2 The CEM algorithm for univariate Gaussian mixture models

E-step:

For this step we need to calculate the responsibility that the j^{th} component belongs to the i^{th} observation,

using equation 11 derived in section 2.4.2:

$$\begin{aligned}\hat{\gamma_{ij}}(t) &= P\left(\gamma_i = j|\hat{\boldsymbol{\Theta}}(t)\right) \\ &= \frac{p\left(y_i|\hat{\theta}_j(t)\right)\hat{\omega}_j(t)}{\sum_{k=1}^{K} p\left(y_i|\hat{\theta}_k(t)\right)\hat{\omega}_k(t)} \\ &= \frac{\left[\frac{1}{\sqrt{2\pi\hat{\sigma}_j(t)}}\exp\left(-\frac{1}{2\hat{\sigma}_j^2(t)}\left(y_i-\hat{\mu}_j(t)\right)^2\right)\right]\hat{\omega}_j(t)}{\sum_{k=1}^{K}\left[\frac{1}{\sqrt{2\pi\hat{\sigma}_k(t)}}\exp\left(-\frac{1}{2\hat{\sigma}_k^2(t)}\left(y_i-\hat{\mu_k}(t)\right)^2\right)\right]\hat{\omega}_k(t)}\end{aligned} \tag{25}$$

for $i = 1, 2, ..., N; j = 1, 2, ..., J$ and $t \geq 0$

C-step:

For this step we design a partition $P = (P_1, P_2, ..., P_j)$ of $(y_1, y_2, ..., y_N)$ by assigning each observation to the component which maximizes the responsibility γ_{ij} [7]. Note that according to Faria, (2010) "if the maximum responsibility is not unique, the component with the smallest index is chosen"[7]. Thus,

$$P_j = \left\{y_i : \gamma_{ij} = \underset{h}{\arg\max}\ \gamma_{ih}\right\} \tag{26}$$

if $\gamma_{ij} = \gamma_{ih}$ and $j < h$ then $y_i \epsilon P_j$ for $j = 1, 2, ..., K$.

Note that if the partition is either empty or only has a single observation, we then consider a mixture with $K - 1$ components and start the process with said components instead.

M-step:

For this step the estimates are updated using the sub-samples P_j.

It has been shown in section 2.4.2 that at time step $t+1$ the mixing coefficient estimates can be obtained as follows

$$\hat{\omega}_j(t+1) = \frac{N_j}{N} \ \ , j = 1, 2, ..., K$$

where N_j denotes the total number of observations assigned to the j^{th} component.

By setting the partial derivatives equal to zero,we are able to find an expression for $\hat{\boldsymbol{\eta}}(t+1)$ where $\boldsymbol{\eta} = (\theta_1, \theta_2, ..., \theta_K) = \left(\left(\mu_1, \sigma_1^2\right), \left(\mu_2, \sigma_2^2\right), ..., \left(\mu_K, \sigma_K^2\right)\right)$ as follows:

$$\frac{\partial Q\left(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}(t)\right)}{\partial\hat{\boldsymbol{\eta}}(t)} = \sum_{j=1}^{J}\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)\frac{\partial \log p\left(y_i|\hat{\theta}_j(t)\right)}{\partial\hat{\epsilon}(t)} = 0 \tag{27}$$

However, the Gaussian component-conditional probability density function, equation 16 is required before

the roots can be found, which is defined as

$$\begin{aligned}\log p\left(y_i|\hat{\theta}_j(t)\right) &= \log p\left(y_i|\hat{\mu}_j(t),\hat{\sigma}_j^2(t)\right)\\ &= \log\left[\frac{1}{\hat{\sigma}_j(t)\sqrt{2\pi}}\exp\left(-\frac{(y_i-\hat{\mu}_j(t))^2}{2\hat{\sigma}_j^2(t)}\right)\right]\\ &= \log\left(\frac{1}{\sqrt{2\pi}}\right)+\log\left(\frac{1}{\hat{\sigma}_j(t)}\right)-\frac{(y_i-\hat{\mu}_j(t))^2}{2\hat{\sigma}_j^2(t)}\end{aligned} \tag{28}$$

Taking the partial derivatives of equation 28, with regards to $\hat{\mu}_j(t)$ and $\hat{\sigma}_j^2(t)$, we are given the following

$$\frac{\partial \log p\left(y_i|\hat{\theta}_j(t)\right)}{\partial\hat{\mu}_j(t)} = \frac{(y_i-\hat{\mu}_j(t))}{\hat{\sigma}_j^2(t)} \tag{29}$$

and

$$\frac{\partial \log p\left(y_i|\hat{\theta}_j(t)\right)}{\partial\hat{\sigma}_j^2(t)} = -\frac{1}{2\hat{\sigma}_j^2(t)}+\frac{(y_i-\hat{\mu}_j(t))^2}{2\hat{\sigma}_j^4(t)} \tag{30}$$

Using equation 28, the Q function 27 can be written as

$$\begin{aligned}\frac{\partial Q\left(\boldsymbol{\Theta},\hat{\boldsymbol{\Theta}}(t)\right)}{\partial\hat{\boldsymbol{\eta}}(t)} &= \sum_{j=1}^{J}\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)\frac{\partial \log p\left(y_i|\hat{\theta}_j(t)\right)}{\partial\hat{\boldsymbol{\eta}}(t)}\\ &= \sum_{j=1}^{J}\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)\frac{\partial}{\partial\hat{\boldsymbol{\eta}}(t)}\left[\log\left(\frac{1}{\sqrt{2\pi}}\right)+\log\left(\frac{1}{\hat{\sigma}_j(t)}\right)-\frac{(y_i-\hat{\mu}_j(t))^2}{2\hat{\sigma}_j^2(t)}\right]\\ &= 0\end{aligned} \tag{31}$$

Equation 31 results in the following equations:

$$\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)\frac{\partial}{\partial\hat{\mu}_j(t)}\left[\log\left(\frac{1}{\sqrt{2\pi}}\right)+\log\left(\frac{1}{\hat{\sigma}_j(t)}\right)-\frac{(y_i-\hat{\mu}_j(t))^2}{2\hat{\sigma}_j^2(t)}\right]=0 \tag{32}$$

and

$$\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)\frac{\partial}{\partial\hat{\sigma}_j^2(t)}\left[\log\left(\frac{1}{\sqrt{2\pi}}\right)+\log\left(\frac{1}{\hat{\sigma}_j(t)}\right)-\frac{(y_i-\hat{\mu}_j(t))^2}{2\hat{\sigma}_j^2(t)}\right]=0 \tag{33}$$

The solutions of equations 29 and 30 can be obtained by using the results of equations 32 and 33 as follows:

Solving equation 29:

$$\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) \frac{\partial}{\partial \hat{\mu}_j(t)} \left[\log\left(\frac{1}{\sqrt{2\pi}}\right) + \log\left(\frac{1}{\hat{\sigma}_j(t)}\right) - \frac{(y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^2(t)} \right] = 0$$

$$\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) \frac{(y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^2(t)} = 0$$

$$\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) y_i - \sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) \hat{\mu}_j(t) = 0$$

$$\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) y_i = \sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) \hat{\mu}_j(t)$$

$$\hat{\mu}_j(t) = \frac{\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) y_i}{\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t)} \tag{34}$$

Solving equation 30:

$$\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) \frac{\partial}{\partial \hat{\sigma}_j^2(t)} \left[\log\left(\frac{1}{\sqrt{2\pi}}\right) + \log\left(\frac{1}{\hat{\sigma}_j(t)}\right) - \frac{(y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^2(t)} \right] = 0$$

$$\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) \left[-\frac{1}{2\hat{\sigma}_j^2(t)} + \frac{(y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^4(t)} \right] = 0$$

$$-\frac{\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t)}{2\hat{\sigma}_j^2(t)} + \frac{\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) (y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^4(t)} = 0$$

$$-\frac{\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t)}{2\hat{\sigma}_j^2(t)} = -\frac{\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) (y_i - \hat{\mu}_j(t))^2}{2\hat{\sigma}_j^4(t)}$$

$$\hat{\sigma}_j^2(t) = \frac{\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) (y_i - \hat{\mu}_j(t))^2}{\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t)} \tag{35}$$

Therefore, at time $t+1$ the component parameter updates which enable equations 34 and 35 to be used in an iterative process are defined as follows

$$\hat{\mu}_j(t+1) = \frac{\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) y_i}{\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t)}, j = 1, 2, ..., J \, and \, t \geq 0 \tag{36}$$

and

$$\hat{\sigma}_j^2(t+1) = \frac{\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t) (y_i - \hat{\mu}_j(t))^2}{\sum_{i=1}^{N_j} \hat{\gamma}_{ij}(t)}, j = 1, 2, ..., J \, and \, t \geq 0 \tag{37}$$

The CEM algorithm for a univariate K-component Gaussian mixture model is summarized in Algorithm 4 based on Li, (2005) [14].

Algorithm 4 The CEM Algorithm for the univariate Gaussian case.

1) Algorithm starts by taking initial values generated by the K-means algorithm for the unknown parameters $\hat{\mu}_j(0)$, $\hat{\sigma}_j^2(0)$ and $\hat{\omega}_j(0)$ for $j = 1, 2, ..., K$.
2) Expectation Step: Compute the responsibilities

$$\hat{\gamma_{ij}}(t) = \frac{p\left(y_i|\hat{\theta}_j(t)\right)\hat{\omega}_j(t)}{\sum_{k=1}^{K} p\left(y_i|\hat{\theta}_k(t)\right)\hat{\omega}_k(t)}$$
$$= \frac{\left[\frac{1}{\sqrt{2\pi\hat{\sigma}_j(t)}}\exp\left(-\frac{1}{2\hat{\sigma}_j^2(t)}\left(y_i - \hat{\mu}_j(t)\right)^2\right)\right]\hat{\omega}_j(t)}{\sum_{k=1}^{K}\left[\frac{1}{\sqrt{2\pi\hat{\sigma}_k(t)}}\exp\left(-\frac{1}{2\hat{\sigma}_k^2(t)}\left(y_i - \hat{\mu}_k(t)\right)^2\right)\right]\hat{\omega}_k(t)}$$

where $\theta_j = (\mu_j, \sigma_j^2)$ for $i = 1, 2, ..., N$ and $j = 1, 2, ..., K$
3) Classification Step: Design a partition $P = (P_1, P_2, ..., P_K)$ by assigning each observation to a component which maximizes γ_{ij}:

$$P_j = \left\{y_i : \gamma_{ij} = \arg\max_h \gamma_{ih}\right\}$$

if $\gamma_{ij} = \gamma_{ih}$ and $j < h$ then $y_i \epsilon P_j$ for $j = 1, 2, ..., K$
4) Maximization Step: Determine the maximum-likelihood estimators of the unknown parameters using the sub-samples P_j as follows

$$\hat{\omega}_j(t+1) = \frac{N_j}{N}, j = 1, 2, ..., K$$

with N_j denoting the total number of observations assigned to component j.
and

$$\hat{\mu}_j(t+1) = \frac{\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)y_i}{\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)}, j = 1, 2, ..., K$$

and

$$\hat{\sigma}_j^2(t) = \frac{\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)\left(y_i - \hat{\mu}_j(t)\right)^2}{\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)}, j = 1, 2, ..., K$$

5) Alternate between steps 2, 3 and 4 until convergence

2.5.3 The CEM algorithm for multivariate Gaussian mixture models

As mentioned in section 2.5.1, the multivariate K-component Gaussian mixture has a similar structure to that of the univariate case, with a component-conditional probability density function of the form

$$p\left(\boldsymbol{y}|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}\right) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma_j}|^{1/2}}\exp\left(-\frac{1}{2}\left(\boldsymbol{y} - \boldsymbol{\mu_j}\right)^T \boldsymbol{\Sigma_j^{-1}}\left(\boldsymbol{y} - \boldsymbol{\mu_j}\right)\right)$$

where $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N) \in \Re^p$ is an unlabeled sample of observations assumed to be *i.i.d.*

As with the univariate case, the CEM algorithm for a multivariate K-component Gaussian mixture model is summarized in Algorithm 5 based on Li, (2005) [14].

Algorithm 5 The CEM algorithm for the multivariate Gaussian case.

1) Algorithm starts by taking initial values generated by the K-means algorithm for the unknown parameters $\hat{\boldsymbol{\mu}}_{\boldsymbol{j}}(0)$,$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{j}}(0)$ and $\hat{\omega}_j(0)$ for $j = 1, 2, ..., K$.
2) Expectation Step: Compute the responsibilities

$$\begin{aligned}\hat{\gamma_{ij}}(t) &= \frac{p\left(\boldsymbol{y}_i|\hat{\boldsymbol{\theta}}_{\boldsymbol{j}}\right)\hat{\omega}_j(t)}{\sum_{k=1}^{K} p\left(\boldsymbol{y}_i|\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}\right)\hat{\omega}_k(t)} \\ &= \frac{\left[\frac{1}{(2\pi)^{p/2}|\hat{\boldsymbol{\Sigma}}_{\boldsymbol{j}}(t)|}\exp\left(-\frac{1}{2}\left(\boldsymbol{y_i}-\hat{\boldsymbol{\mu}}_{\boldsymbol{j}}(t)\right)^T\boldsymbol{\Sigma_j^{-1}}(t)\left(\boldsymbol{y_i}-\hat{\boldsymbol{\mu}}_{\boldsymbol{j}}(t)\right)\right)\right]\hat{\omega}_j(t)}{\sum_{k=1}^{K}\left[\frac{1}{(2\pi)^{p/2}|\hat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}(t)}\exp\left(-\frac{1}{2}\left(\boldsymbol{y_i}-\hat{\boldsymbol{\mu}}_{\boldsymbol{k}}(t)\right)^T\boldsymbol{\Sigma_k^{-1}}(t)\left(\boldsymbol{y_i}-\hat{\boldsymbol{\mu}}_{\boldsymbol{k}}(t)\right)\right)\right]\hat{\omega}_k(t)}\end{aligned}$$

with $\boldsymbol{\theta_j} = \left(\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}\right)$ for $i = 1, 2, ..., N$ and $j = 1, 2, ..., K$
3) Classification Step: Design a partition $P = (P_1, P_2, ..., P_K)$ by assigning each observation to a component which maximizes γ_{ij}:

$$P_j = \left\{\boldsymbol{y}_i : \gamma_{ij} = \underset{h}{\arg\max}\ \gamma_{ih}\right\}$$

if $\gamma_{ij} = \gamma_{ih}$ and $j < h$ then $\boldsymbol{y}_i \epsilon P_j$for $j = 1, 2, ..., K$
4) Maximization Step: Determine the maximum-likelihood estimators of the unknown parameters using the sub-samples P_j as follows

$$\hat{\omega}_j(t+1) = \frac{N_j}{N}, j = 1, 2, ..., K$$

with N_j denoting the total number of observations assigned to component j and

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{j}}(t+1) = \frac{\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)\boldsymbol{y_i}}{\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)}, j = 1, 2, ..., K$$

and

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{j}}(t+1) = \frac{\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)\left(\boldsymbol{y_i}-\hat{\boldsymbol{\mu}}_{\boldsymbol{j}}(t)\right)\left(\boldsymbol{y_i}-\hat{\boldsymbol{\mu}}_{\boldsymbol{j}}(t)\right)^T}{\sum_{i=1}^{N_j}\hat{\gamma}_{ij}(t)}, j = 1, 2, ..., K$$

5) Alternate between steps 2, 3 and 4 until convergence

3 Application

3.1 Introduction

The EM and CEM algorithm's ability to estimate the parameters of a Gaussian mixture model is demonstrated by making use of the Old Faithful data set. The Old Faithful data set measures the waiting time between each eruption as well as the duration of each eruption of the famous Old Faithful hot water geyser in Yellowstone National park. Note that due to the fact that we are dealing with the univariate case, only the duration of each eruption is considered, which consists of 272 data points, $\boldsymbol{y} = (y_1, y_2, ..., y_{272}) \in \Re$.

Before the EM and CEM algorithms can be applied to the data set, we first look at the distribution of the data, where figure 1 illustrates a single Gaussian curve over a histogram of the data, where the

mean and the variance of data is equivalent to sample statistics, i.e. $N\left(\mu, \sigma^2\right) = N\left(3.48778, 1.302726\right)$, fitted to the data. On inspection of the data, it is clear that the data has bi-modality and thus a single Gaussian density will not suffice. Therefore, we will fit a two component mixture of univariate Gaussian distributions to model the data.

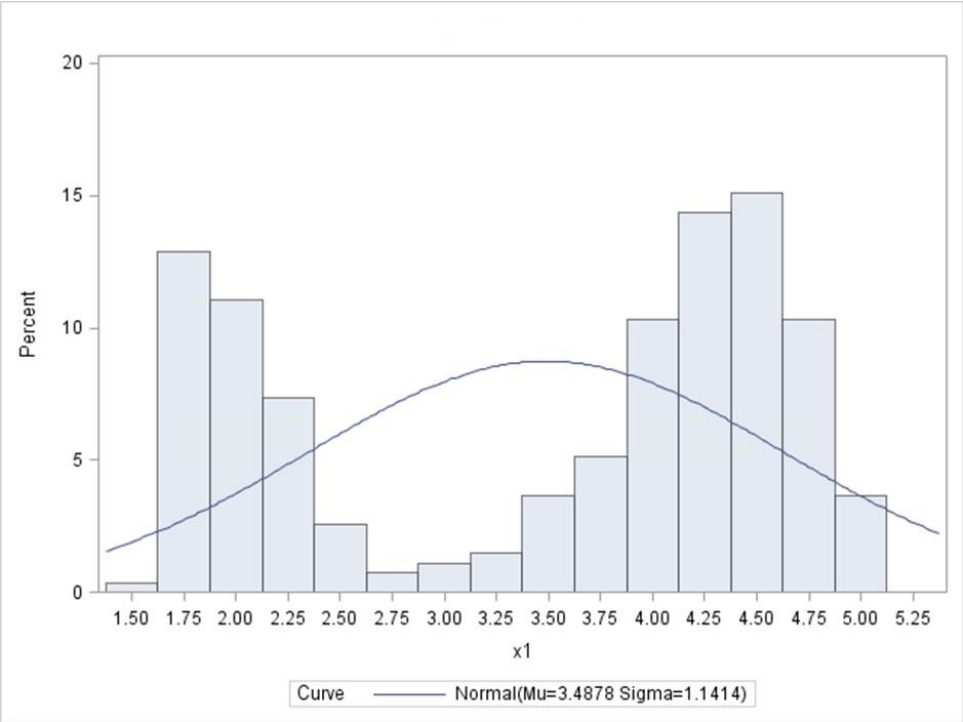


Figure 1: Histogram fitted with a single Gaussian curve.

Making use of equation 1 in section 2.1, we want to estimate the parameters of the following two component mixture model:

$$p\left(y_i|\mathbf{\Theta}\right) = \sum\nolimits_{j=1}^{2} \omega_j p\left(y_i|\theta_j\right)$$

where ω_j are the mixing coefficients of the distribution ($0 < \omega_j < 1$ and $\sum_{j=1}^{2} \omega_j = 1$),
$\mathbf{\Theta} = (\theta_1, \theta_2, \omega_1, \omega_2)$ denotes the complete set of parameters that specify the mixture with $\theta_j = \left\{\mu_j, \sigma_j^2\right\}$ for $j = 1, 2$ and $\epsilon = (\theta_1, \theta_2)$ representing the parameters of each component density.

The component-conditional density function, equation 2 from section 2.1 with all components following a univariate Gaussian distribution, i.e. $Y_i \sim N\left(\mu_j, \sigma_j^2\right)$ for $i = 1, 2, ..., 272$ and $j = 1, 2$ has the following form:

$$p\left(y_i|\theta_j\right) = p\left(y_i|\mu_j, \sigma_j^2\right)$$
$$= \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left(-\frac{\left(y_i - \mu_j\right)^2}{2\sigma_j}\right)$$

Figure 2 illustrates the distribution of the data, with a two component Gaussian curve over a histogram of the data points.

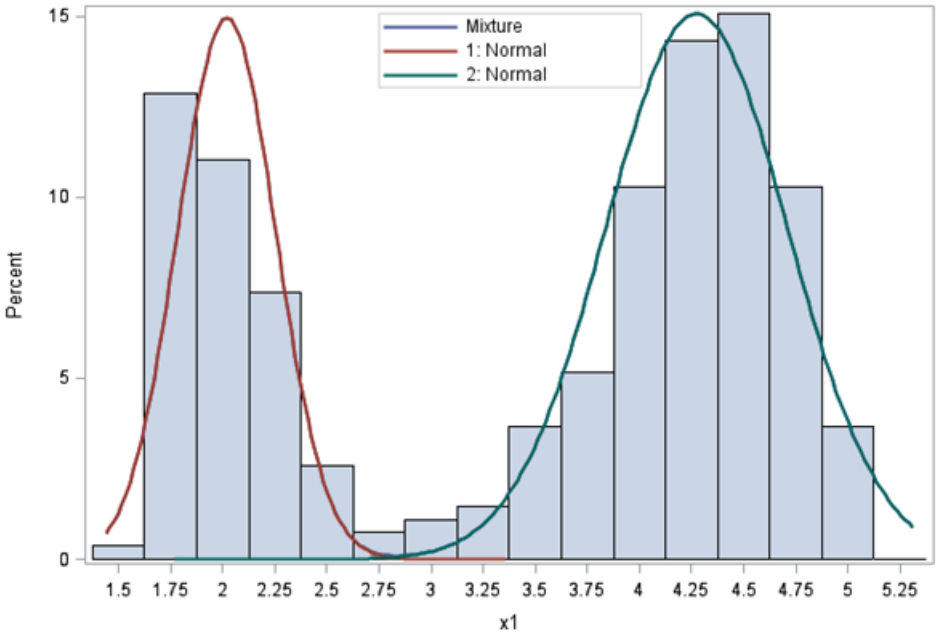


Figure 2: Histogram fitted with a two component Gaussian curve.

3.2 Application of the EM and CEM algorithms

Due to the fact that the EM algorithm as well as the classification version of the EM has problems with convergence, the K-means clustering algorithm was proposed to initialize the algorithms. The initial values for the mean,variance and mixing coefficients are presented in Table 2 below:

$\hat{\mu}_1$	$\hat{\mu}_2$
4.29834	2.04863

$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$
0.161061	0.08128

$\hat{\omega}_1$	$\hat{\omega}_2$
0.639706	0.360294

Table 1: Initial values generated by the K-means clustering algorithm.

The EM and CEM algorithms were applied using the above mentioned initial values, with the MLE's of the parameters presented in Table 2.

EM algorithm	
Component 1	Component 2
$\hat{\mu}_1 = 4.2733434$	$\hat{\mu}_2 = 2.0186078$
$\hat{\sigma}_1^2 = 0.1910242$	$\hat{\sigma}_2^2 = 0.0555176$
$\hat{\omega}_1 = 0.6515954$	$\hat{\omega}_2 = 0.3484046$

CEM algorithm	
Component 1	Component 2
$\hat{\mu}_1 = 4.2774028$	$\hat{\mu}_2 = 2.0160435$
$\hat{\sigma}_1^2 = 0.1854318$	$\hat{\sigma}_2^2 = 0.0534328$
$\hat{\omega}_1 = 0.6507353$	$\hat{\omega}_2 = 0.3492647$

Table 2: Maximum likelihood estimates generated by the EM algorithm.

Table 2, shows the difference in the estimation results from the two estimation techniques. The two algorithms should however produce fairly similar parameter estimates. Furthermore, the convergence of the parameters for both EM and CEM algorithms are illustrated in Figure 3.

EM algorithm CEM algorithm

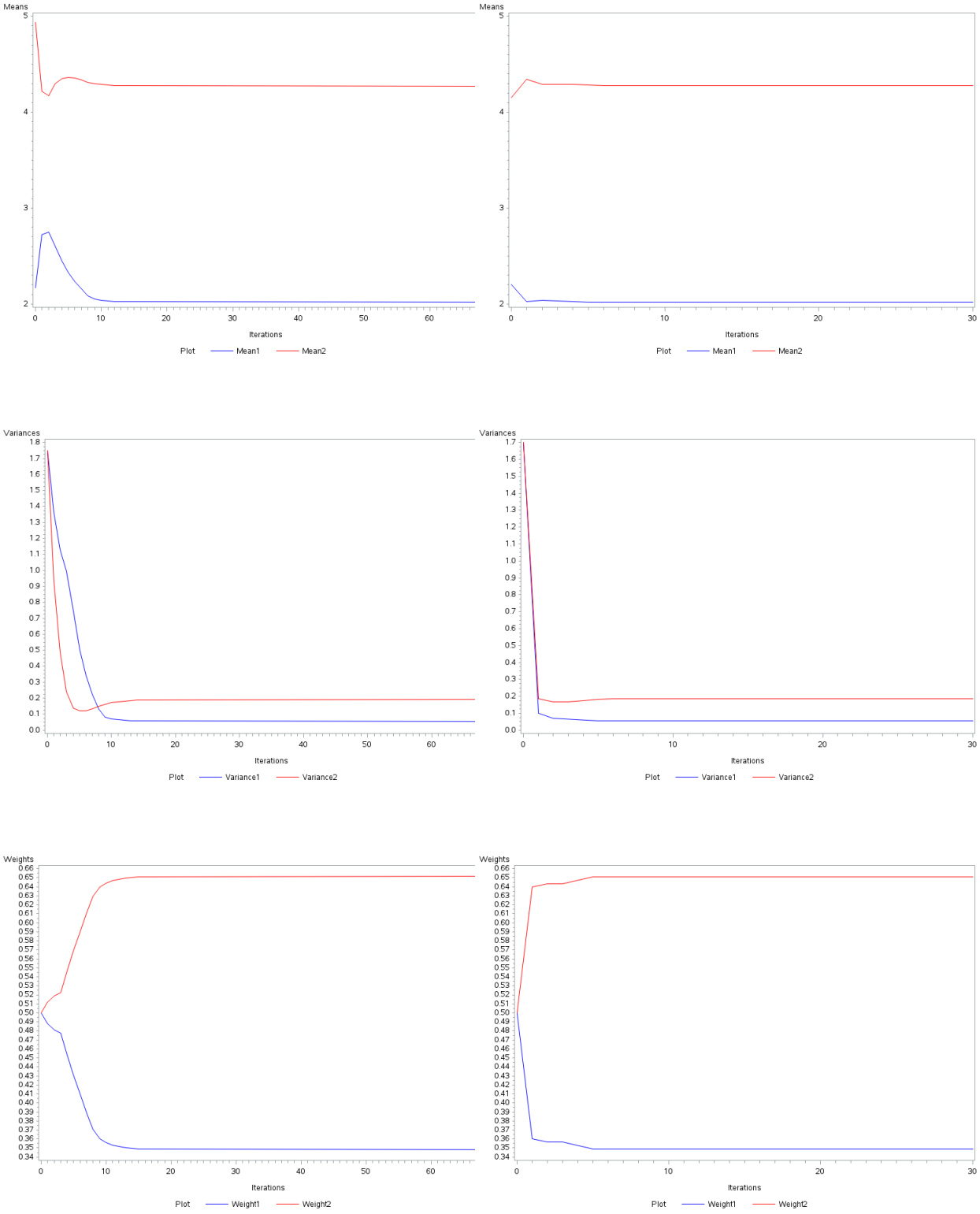


Figure 3: Convergence of parameter estimates for EM and CEM.

On closer inspection of Figure 3, it would appear that the CEM algorithm runs through fewer iterations than that of the EM algorithm. Thus, the computational time needed to reach the parameter estimates

for the CEM algorithm is far less than that of the EM algorithm. It is also important to note that the MLE's produced using the randomly selected initial values are the same as the MLE's produced using the initial values generated by the K-means algorithm for both procedures.

3.3 Simulation study

The purpose of this simulation study is to analyse the performance as well as the efficiency of estimation of the two methods by comparing the absolute bias of the estimated parameters. In keeping with the univariate case, data will be generated from two random Gaussian distributions to create bi-modal data sets.

3.3.1 Design of the study

Initial values. The initial values that were used for both algorithms were randomly selected from these data sets.

Convergence criterion. The EM and CEM algorithms were stopped when the log-likelihood difference between two successive iterations were less than 10^{-2}.

Sample sizes. The size of the samples generated from the bi-modal data sets were $n = \{15, 30, 50, 100, 200, 500\}$.

True Parameters. The true parameters that were used are as follows*:*

- $\mu_1 = 40$
- $\mu_2 = \mu_1 + \delta$
- $\sigma_1^2 = 16$
- $\sigma_2^2 = \sigma_1^2$
- $\pi_1 = 0.2$
- $\pi_2 = 1 - \pi_1$

It is important to note that δ and π_i were changed and that the variances were kept constant for simplicity.

Measure of the performance and estimation efficiency. The performance of the two estimation methods are examine using the following criteria:

1. the average number of iterations required for convergence (implying less computational time needed),
2. the absolute bias of the parameter estimates
   (a) These biases are calculated using the following formula:

$$BIAS\left(\hat{\boldsymbol{\Theta}}\right) = \left|\boldsymbol{\Theta}_{true} - \hat{\boldsymbol{\Theta}}\right|$$

Steps of the simulation study:

1. Bi-modal data set is created using two random Gaussian distributions using the true parameter values and different sample sizes, $n = \{15, 30, 50, 100, 200, 500\}$.
2. Initial values for the EM and CEM algorithms are randomly selected from these data sets.
3. EM and CEM algorithms iterate until the stopping criterion, $< 10^{-2}$ is met.
4. 500 simulations are run on each sample size, the averages as well as the absolute biases are calculated.
5. Steps 1 to 4 are repeated with π_i and δ being changed.
   (a) There are 9 different cases that are observed, namely:
      i. Case 1: $\delta = 15$ $\pi_1 = 0.2$
      ii. Case 2: $\delta = 15$ $\pi_1 = 0.4$
      iii. Case 3: $\delta = 15$ $\pi_1 = 0.5$
      iv. Case 4: $\delta = 30$ $\pi_1 = 0.2$
      v. Case 5: $\delta = 30$ $\pi_1 = 0.4$
      vi. Case 6: $\delta = 30$ $\pi_1 = 0.5$
      vii. Case 7: $\delta = 50$ $\pi_1 = 0.2$
      viii. Case 8: $\delta = 50$ $\pi_1 = 0.4$
      ix. Case 9: $\delta = 50$ $\pi_1 = 0.5$

3.3.2 Results of the simulation study

The results for both estimation methods are tabulated below and are accompanied by the relevant graphs.

The EM algorithm:

Table 3 displays the absolute biases of the parameters for cases 1 to 3 . Tables 5 and 6 illustrating the results for cases 4 to 9 can be found in the appendix.

		True values	n	Iterations	μ_1	μ_2	σ_1^2	σ_2^2	π_1
Case 1	μ_1	40	15	-	-	-	-	-	-
	μ_2	55	30	19.38	2.7611352	0.3081737	16.0150444	3.6678609	0.1000838
	σ_1^2	16	50	21.649	2.5658782	0.3087611	15.324794	3.23117275	0.0933542
	σ_2^2	16	100	24.40467	2.4320323	0.2909146	15.0809984	2.9397179	0.0863445
	π_1	0.2	200	27.7835	2.2149810	0.2720570	13.8603951	2.5902209	0.0767920
	π_2	0.8	500	31.1296	1.9549456	0.2459505	12.3201656	2.2587106	0.0668272
Case 2	μ_1	40	15	-	-	-	-	-	-
	μ_2	55	30	13.818	0.42203388	0.2438169	1.3958618	2.2176552	0.0233727
	σ_1^2	16	50	14.67	0.3445329	0.1981671	1.0999196	1.7859366	0.0189474
	σ_2^2	16	100	15.764	0.2583457	0.1594106	0.8189817	1.3867957	0.0143083
	π_1	0.4	200	16.819	0.2045108	0.1228686	0.6009611	1.1221634	0.0113877
	π_2	0.6	500	17.9532	0.1708918	0.0989171	0.5122214	0.9251890	0.0094935
Case 3	μ_1	40	15	-	-	-	-	-	-
	μ_2	55	30	11.306	0.0232700	0.0003775	0.8539629	0.5120665	0.0015787
	σ_1^2	16	50	11.447	0.0098546	0.0381091	0.5344252	0.6062985	0.0012293
	σ_2^2	16	100	11.482	0.0013686	0.0235777	0.4032293	0.4604883	0.0008819
	π_1	0.5	200	11.52	0.0040118	0.0090390	0.3347030	0.3346143	0.0003744
	π_2	0.5	500	11.5376	0.0019445	0.0082852	0.2884526	0.2864216	0.0002824

Table 3: EM : $\delta = 15$

The CEM algorithm:

Table 4 displays the absolute biases of the parameters for cases 1 to 3. Tables 7 and 8 illustrating the results for cases 4 to 9 can be found in the appendix.

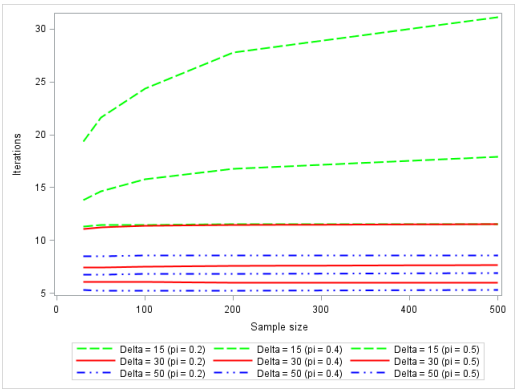
		True values	n	Iterations	μ_1	μ_2	σ_1^2	σ_2^2	π_1
Case 1	μ_1	40	15	-	-	-	-	-	-
	μ_2	55	30	7.874	2.6512848	1.5501417	5.7260239	5.4303035	0.1706
	σ_1^2	16	50	9.072	2.3921766	1.5257578	5.8396435	4.6298727	0.15196
	σ_2^2	16	100	10.944	1.4539991	1.1470505	3.5072542	2.7043497	0.12242
	π_1	0.2	200	12.516	0.4488333	0.6680815	0.6622261	1.263647	0.06396
	π_2	0.8	500	14.9	0.2745903	0.5466815	0.1407739	0.9474159	0.05436
Case 2	μ_1	40	15	-	-	-	-	-	-
	μ_2	55	30	7.37	0.7166438	0.62897	3.6406556	0.9838261	0.054
	σ_1^2	16	50	8.042	0.2830338	0.5545079	3.162949	0.6685247	0.03512
	σ_2^2	16	100	9.714	0.1626619	0.4827974	2.0118516	0.3588715	0.02844
	π_1	0.4	200	11.488	0.0082236	0.4257197	1.8214662	0.0652361	0.02557
	π_2	0.6	500	13.718	0.081324	0.3185683	0.5872703	0.0610236	0.016544
Case 3	μ_1	40	15	-	-	-	-	-	-
	μ_2	55	30	7.426	0.0454437	0.0779438	1.7853902	1.5844912	0.0030667
	σ_1^2	16	50	7.88	0.1958384	0.0382547	1.2918624	1.6414708	0.00648
	σ_2^2	16	100	9.546	0.2516107	0.214009	1.2039237	1.3864808	0.00328
	π_1	0.5	200	10.916	0.2037506	0.2654704	1.21108842	0.7467399	0.00283
	π_2	0.5	500	13.482	0.2275375	0.2172687	0.7881913	0.8810092	0.00088

Table 4: CEM : $\delta = 15$

In tables 3 to 8 the absolute bias of the parameters were calculated for both the EM and CEM algorithms,

respectively. Note that for both the EM and CEM algorithms, the simulation for sample size $n = 15$ could not be conducted, as the sample size was too small and thus proper partitioning of the sample could not occur. The average number of iterations from these tables give a clear indication that the CEM takes fewer iterations to reach convergence than the EM algorithm (less computational time) over the specified sample space. This is further strengthened by figure 4 below, which also shows that the average number of iterations increase as the sample size increases. Figures 4 to 7 compare the absolute bias of the parameters for the EM and CEM algorithms.

EM algorithm



CEM algorithm

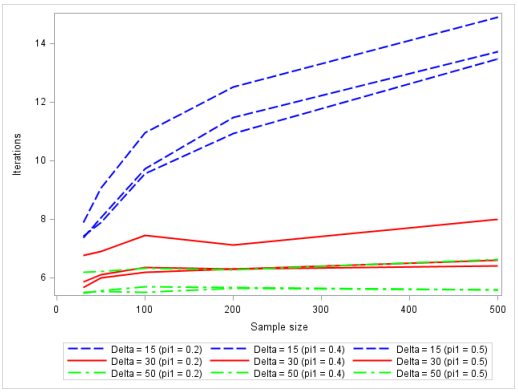
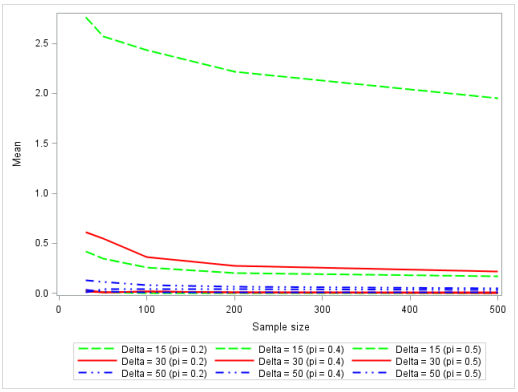
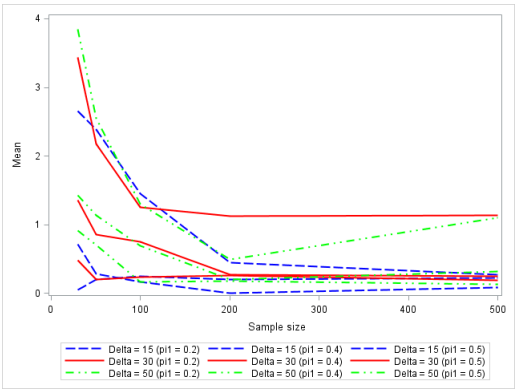
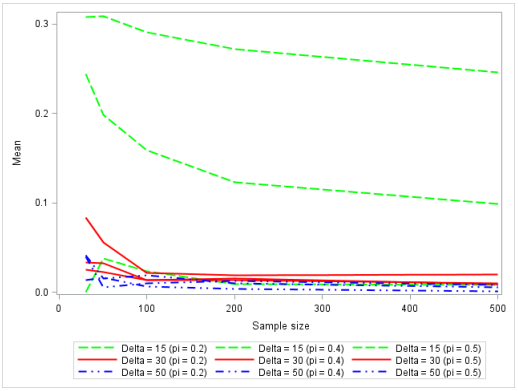


Figure 4: Absolute iteration bias over the sample space for EM and CEM.







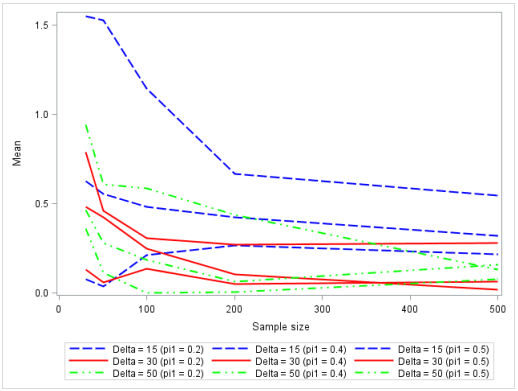


Figure 5: Absolute mean bias over the sample space for EM and CEM.

EM algorithm

CEM algorithm

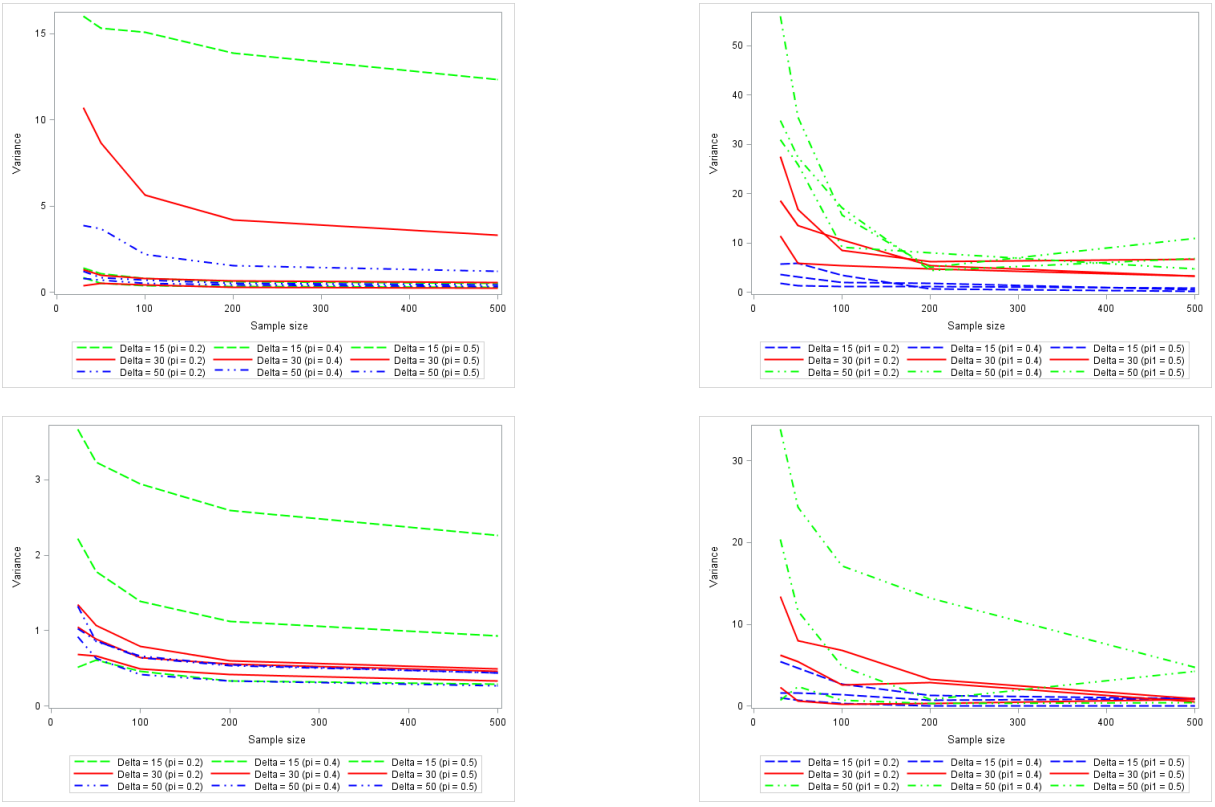


Figure 6: Absolute variance bias over the sample space for EM and CEM.

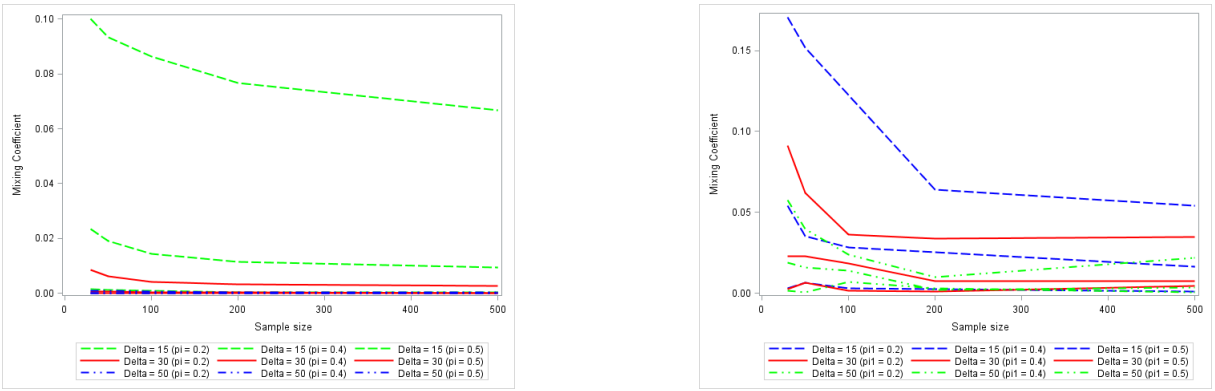


Figure 7: Absolute mixing coefficient bias over the sample space for EM and CEM.

Figures 4 to 7 illustrate the tables calculated above for the EM and CEM algorithms. It can be observed that for a large δ the absolute biases of the average number of iterations for the EM and CEM, in figure 4 remain relatively constant, with the CEM showing more of an increase than that of the EM. In figures 5 to 7 it can be seen that on average the EM estimates have a smaller absolute bias than the CEM estimates, with the EM estimates gradually converging over the specified sample space and CEM estimates converging sharply. It can also be observed that as the sample size increases the absolute bias for the EM gradually decreases, with the CEM decreasing more sharply. Therefore, the CEM estimates decrease a

lot faster as the sample size increases, than the EM algorithm.

It should also be noted that the EM algorithm produces better parameter estimates, especially the variance parameters, in every case that was observed in the simulation study. The CEM algorithm seems to be more computationally efficient than the EM algorithm, however, the EM algorithm converges better to the true parameters than that of the CEM.

4 Conclusion and recommendations

In this research report, the performance as well as the efficiency of estimation of two estimation algorithms to produce maximum likelihood estimates were examined, the EM algorithm and CEM algorithm. In the first part of our application the Old Faithful Water Geyser, was used. The results showed that the EM and CEM algorithms produced slightly different results, which is to be expected. It was also observed that the CEM algorithm took fewer iterations to reach convergence than that of the EM algorithm, thus requiring less computational time.

We went a step further and initialized both estimation methods using the K-means clustering algorithm, since literature would suggest that both algorithms struggle under poor initialization. The initialization technique showed a considerable improvement in the number of iterations taken by the EM algorithm however, showed no improvement in the CEM algorithm. This lack of improvement could come down to the fact that like the K-means clustering algorithm, the CEM algorithm is also a hard classification technique and thus proper initialization would not improve the algorithm.

In the second part of our application a simulation study was conducted, where random initial values were used, to examine the performance as well as the efficiency of estimation for both estimation methods. The results showed that the CEM algorithm (almost always) takes a fewer number of iterations to reach convergence than that of the EM algorithm, which implies that less computational time is needed. The results also showed that the EM algorithm produces better parameter estimates than that of the CEM algorithm, which leads to smaller absolute biases of the parameter estimates. Therefore, after carefully inspection of the tables and graphs we can conclude that even though the CEM is deemed to be more computationally efficient, the EM performs better than the CEM.

It is important to note that when simulating the CEM algorithm, complications were experienced. This problem occurred during the classification of the posterior probabilities (responsibilities) in the C-step of the CEM algorithm. This led to the probabilities being grouped into one of the clusters after a certain

number of iterations, thus leaving the other cluster empty. This "misclassification" resulted in the premature ending of the algorithm, which in turn led to the simulation not running its full course. Due to the fact that the data set used was random for each simulation, the number of simulations run, varied. Therefore, for the sake of results, we bypassed this issue by creating a macro in SAS, in which we entered these simulation results into a data set until 500 simulations were achieved. These results were then used to calculate the relevant averages needed as well as the graphs found in figures 4 to 7. Therefore, it is recommended that further research is conducted on this technical issue, to ensure accurate results for the simulation of the CEM algorithm can be achieved.

References

[1] C. Biernacki. Degeneracy in the maximum likelihood estimation of univariate Gaussian mixtures for grouped data and behaviour of the EM algorithm. *Scandinavian Journal of Statistics*, 34(3):569–586, 2007.

[2] J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical report, International Computer Science Institute (ICSI), 1997.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. pp. 424-435.

[4] H. Cramer. *Mathematical Methods of Statistics*. Princeton University Press, 1946.

[5] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[6] L.J. Bain & M. Engelhardt. *Introduction to Probability and Mathematical Statistics*. Brooks/Cole, 2nd edition, 2000.

[7] S. Faria and G. Soromenho. Fitting mixtures of linear regressions. *Journal of Statistical Computation and Simulation*, 80(2):201–225, 2010.

[8] J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer Series in Statistics Springer, Berlin, 2001. pp. 272-276; 460-463; 509-510.

[9] Z. Ghahramani and M.I. Jordan. Supervised learning from incomplete data via an EM approach. *Advances in Neural Information Processing Systems*, pages 120–120, 1994.

[10] P. Giudici and S. Figini. *Applied Data Mining for Business and Industry*. John Wiley & Sons Ltd, 2009. pp. 146.

[11] E.T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge Press, 2003.

[12] D. Karlis and E. Xekalaki. Choosing intial values for the EM algorithm for finite mixtures. *Computational Statistics and Data Analysis*, 2003.

[13] H. Li, K. Zhang, and T. Jiang. The regularized EM algorithm. *AAAI*, pages 807–812, 2005.

[14] J. Li. Clustering based on a multilayer mixture model. *Journal of Computational and Graphical Statistics*, 14(3):547–568, 2005.

[15] D. McLachlan, G. & Peel. Finite mixture models. *Wiley Series in Probability and Statistics*, page 419, 2000.

[16] G.J. McLachlan, T. Krishnan, and S.K. Ng. The EM algorithm. *ECONSTOR: Center for Applied Statistics and Economics (CASE)*, 2004.

[17] I. Melnykov, V. & Melnykov. Initializing the EM algorithm in Gaussian mixture models with an unknown number of components. *Computational Statistics and Data Analysis*, 2012.

[18] T.K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13.6:47–60, 1996.

[19] K.I. Rahmani, N. Pal, and K. Arora. Clustering of image data using K-means and fuzzy K-means. *International Journal of Advanced Computer Science and Applications*, 2014.

[20] R.A. Redner and H.F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):pp.195–239, 1984.

[21] A. Same, C. Ambroise, and G. Govaert. An online classification EM algorithm based on the mixture model. *Springer Science+Business Media*, 2007.

[22] J. Stewart. *Calculus*. Cengage Learning, 2011. pp. 297.

[23] S. Van Wyk. Clustering, self-organizing maps and mixtures of distributions. Master's thesis, University of Pretoria, 2016.

Appendix

Theorems

Theorem 1. *Bayes' Theorem [11]:*

Let $A_1, A_2, ..., A_n$ be a set of mutually exclusive events that together form the sample space of S. Let B by any event from the same sample space, such that $P(B) > 0$. Then,

$$\begin{aligned} P(A_k|B) &= \frac{P(A_k \cap B)}{P(A_1 \cap B) + P(A_2 \cap B) + ... + P(A_n \cap B)} \\ &= \frac{P(A_k)P(B|A_k)}{P(A_1)P(B|A_1) + P(A_2)P(B|A_2) + ... + P(A_n)P(B|A_n)} \end{aligned}$$

Theorem 2. *Lagrange Multipliers [22]:*

To find the maximum and minimum values of $f(x, y, z)$ subject to constraint $g(x, y, z) = k$ [assuming that these extreme values exist and $\nabla g \neq 0$ on the surface $g(x, y, z) = k$]:

1. Find all values of x, y, z and λ such that

$$\nabla f(x, y, z) = \lambda \nabla g(x, y, z)$$

and

$$g(x, y, z) = k$$

2. Evaluate f at all points (x, y, z) that result from step (1). The largest of these values is the maximum value of f; the smallest is the minimum value of f.

Simulation tables:

Tables 5 and 6 display cases 4 to 9 of the EM algorithm.

		True values	n	Iterations	μ_1	μ_2	σ_1^2	σ_2^2	π_1
Case 4	μ_1	40	15	-	-	-	-	-	-
	μ_2	70	30	11.072	0.6090965	0.0337769	10.7057568	1.3454859	0.0086334
	σ_1^2	16	50	11.267	0.5441448	0.0327673	8.6744264	1.0798593	0.0062770
	σ_2^2	16	100	11.39	0.3667943	0.0135992	5.6630023	0.7924117	0.0042829
	π_1	0.2	200	11.4735	0.2781510	0.0153096	4.1827910	0.5955176	0.0032280
	π_2	0.8	500	11.538	0.2202963	0.0094444	3.3093892	0.4935292	0.0025803
Case 5	μ_1	40	15	-	-	-	-	-	-
	μ_2	70	30	7.41	0.0163755	0.0833134	1.3028678	1.0423603	0.0007257
	σ_1^2	16	50	7.48	0.0062132	0.0559114	0.9682361	0.8881211	0.0005280
	σ_2^2	16	100	7.5313	0.0170804	0.0218911	0.8205931	0.6467117	0.0003311
	π_1	0.4	200	7.573	0.0085298	0.0194505	0.6607740	0.5613446	0.0002454
	π_2	0.6	500	7.6468	0.0020188	0.0196286	0.5486961	0.4649031	0.0001985
Case 6	μ_1	40	15	-	-	-	-	-	-
	μ_2	70	30	6.06	0.0341986	0.0257225	0.4024114	0.6802278	0.0001732
	σ_1^2	16	50	6.049	0.0062969	0.0223910	0.5061523	0.6643709	0.0001047
	σ_2^2	16	100	6.0487	0.0167182	0.0138383	0.4156676	0.4930508	0.0000534
	π_1	0.5	200	6.042	0.0138307	0.0133412	0.3056963	0.4193149	0.0000245
	π_2	0.5	500	6.0344	0.0107383	0.0100313	0.2358657	0.3331663	0.0000215

Table 5: EM : $\delta = 30$

		True values	n	Iterations	μ_1	μ_2	σ_1^2	σ_2^2	π_1
Case 7	μ_1	40	15	-	-	-	-	-	-
	μ_2	90	30	8.502	0.1310435	0.0136662	3.8800483	0.9137048	0.0009436
	σ_1^2	16	50	8.523	0.1159066	0.0165066	3.6662457	0.6262966	0.0008331
	σ_2^2	16	100	8.5587	0.0852495	0.0066785	2.1840906	0.4167444	0.0005554
	π_1	0.2	200	8.5645	0.0665938	0.0042451	1.5590072	0.3275412	0.0004166
	π_2	0.8	500	8.6008	0.0482681	0.0013105	1.2355744	0.2741009	0.0003333
Case 8	μ_1	40	15	-	-	-	-	-	-
	μ_2	70	30	6.742	0.0067116	0.0419181	1.2425817	1.3265644	0.0000000
	σ_1^2	16	50	6.77	0.0409195	0.0152746	0.8706070	0.8533440	0.0000000
	σ_2^2	16	100	6.816	0.0412346	0.0187128	0.7076565	0.6644133	0.0000000
	π_1	0.4	200	6.8575	0.0392615	0.0105273	0.5458495	0.5469673	0.0000000
	π_2	0.6	500	6.886	0.0324301	0.0055357	0.4409072	0.4416931	0.0000000
Case 9	μ_1	40	15	-	-	-	-	-	-
	μ_2	70	30	5.31	0.0319570	0.0400893	0.8119555	1.0235545	0.0000000
	σ_1^2	16	50	5.259	0.0140458	0.0051930	0.6957580	0.8650515	0.0000000
	σ_2^2	16	100	5.227	0.0084483	0.0103515	0.5430149	0.6408237	0.0000000
	π_1	0.5	200	5.2185	0.0095728	0.0129235	0.4318209	0.5345352	0.0000000
	π_2	0.5	500	5.2936	0.0064218	0.0087231	0.3354111	0.4357471	0.0000000

Table 6: EM : $\delta = 50$

Tables 7 and 8 display cases 4 to 9 of the CEM algorithm.

		True values	n	Iterations	μ_1	μ_2	σ_1^2	σ_2^2	π_1
Case 4	μ_1	40	15	-	-	-	-	-	-
	μ_2	70	30	6.756	3.438653	0.7874135	27.511768	2.275346	0.0910667
	σ_1^2	16	50	6.906	2.1723592	0.4593421	16.711194	0.6258108	0.06196
	σ_2^2	16	100	7.458	1.2581978	0.3058204	8.5508314	0.267037	0.03628
	π_1	0.2	200	7.116	1.1238826	0.2694495	6.1695381	0.3107791	0.03366
	π_2	0.8	500	7.992	1.133694	0.2807299	6.6415815	0.8286005	0.0348
Case 5	μ_1	40	15	-	-	-	-	-	-
	μ_2	70	30	5.866	1.3579329	0.1319289	18.626022	6.2486322	0.0228
	σ_1^2	16	50	6.118	0.8532396	0.0610227	13.498519	5.3892146	0.02276
	σ_2^2	16	100	6.362	0.754323	0.135662	10.566996	2.6235998	0.01858
	π_1	0.4	200	6.294	0.2724568	0.0526412	5.3761597	2.8515421	0.00757
	π_2	0.6	500	6.398	0.2532378	0.0634764	3.3531704	0.490197	0.007556
Case 6	μ_1	40	15	-	-	-	-	-	-
	μ_2	70	30	5.656	0.4802072	0.4830281	11.485132	13.442724	0.0024
	σ_1^2	16	50	5.986	0.2003627	0.4231389	5.9585487	7.9996704	0.00632
	σ_2^2	16	100	6.18	0.2361816	0.2482891	5.4192875	6.7654543	0.0018
	π_1	0.5	200	6.296	0.2582274	0.1053672	4.8158174	3.2262851	0.00107
	π_2	0.5	500	6.59	0.1843988	0.0169273	3.3059486	0.9336984	0.004452

Table 7: CEM : $\delta = 30$

		True values	n	Iterations	μ_1	μ_2	σ_1^2	σ_2^2	π_1
Case 7	μ_1	40	15	-	-	-	-	-	-
	μ_2	90	30	6.178	3.8496363	0.4627409	55.942163	0.6822243	0.0578
	σ_1^2	16	50	6.202	2.5546343	0.2790106	35.483875	2.3993644	0.03996
	σ_2^2	16	100	6.316	1.3033859	0.1849355	15.593832	0.7195469	0.0239
	π_1	0.2	200	6.276	0.4901582	0.0629255	5.1126492	0.2856098	0.01003
	π_2	0.8	500	6.632	1.097676	0.1573029	10.968423	0.4577337	0.021844
Case 8	μ_1	40	15	-	-	-	-	-	-
	μ_2	90	30	5.514	1.4278191	0.361165	34.830142	20.433498	0.019
	σ_1^2	16	50	5.52	1.1417952	0.1143272	27.39139	11.612193	0.01576
	σ_2^2	16	100	5.496	0.6955952	0.0000498	17.094184	4.8051579	0.01384
	π_1	0.4	200	5.63	0.2028099	0.0035006	4.3600026	0.8256249	0.00211
	π_2	0.6	500	5.582	0.3168199	0.0776246	6.7918049	4.2700591	0.003636
Case 9	μ_1	40	15	-	-	-	-	-	-
	μ_2	90	30	5.468	0.9087234	0.9426394	30.939667	33.861064	0.0018
	σ_1^2	16	50	5.558	0.6997064	0.6104903	25.863925	24.28167	0.00084
	σ_2^2	16	100	5.692	0.1703412	0.5877001	9.0677808	17.136809	0.00718
	π_1	0.5	200	5.66	0.1776498	0.4389985	8.0182226	13.227567	0.00316
	π_2	0.5	500	5.578	0.1279685	0.1336987	4.7516041	4.7390195	0.00064

Table 8: CEM : $\delta = 50$

SAS Code:

EM algorithm with random selection:

```
/*****************************************************************************/
/*Calculating the sample statistics */
data sasuser.Old2;
set Old2;
Type=1;
run;

proc summary data=Old2;
var x1;
class Type;
output out=mean_var mean(x)=ave_x var(x)=var_x;
run;

/*Graph that plots a single Gaussian curve over a histogram*/
title "Distribution of x with a single Gaussian curve";
proc univariate data=sasuser.Old2;
histogram x1 / normal (color=red mu=3.4877831 sigma=1.1413713)
midpoints = 1.50 1.75 2 2.25 2.50 2.75 3 3.25 3.5 3.75 4 4.25 4.50 4.75 5 5.25;
run;

/*Using randomly selected initial values*/
proc iml;
use sasuser.Old2;
read all into erupt;
print erupt;

/*Number of rows and columns*/
dim=ncol(erupt);
N=nrow(erupt);

/*Number of Components (Assumed to be Known)*/
K=2;
```

```
/*Random selection of initial values for the mean, variance and mixing coefficients*/
/*Mean*/
initmean=J(dim,K,.);
initmean[,1]=2.167;
initmean[,2]=4.933;

/*Variance*/
initvar=J(dim,K,.);
do j = 1 to K;
initvar[,j]=1.75;
end;

/*Mixing Coefficients*/
initweight=J(1,K,.);
do j = 1 to K;
initweight[,j]=0.5;
end;

/*Log-likelihood function*/
log_fun=-28000;

/*PI constant*/
pi=constant("pi");

/*Start the iterations*/
t=0;
do until(diff = 0);

/*E-step:*/
post = J(N,5,.);
logfun = J(N,4,.);

/*Calculating the pdfs for each component*/
```

```
do i = 1 to N;
do j = 1 to K;
post[i,j] = 1/((2#pi#initvar[,j])##(dim/2))*exp((-1/(2#initvar[,j]))#((erupt[i,]-initmean[,j])##2));
end;
end;

/*Calculating the pdfs*weights*/
post[,3]=initweight[,1]#post[,1];
post[,4]=initweight[,2]#post[,2];

/*Sum prior*pdf over K1 and K2*/
post[,5]=post[,3]+post[,4];

/*Calculating of the Responsibilities*/
Zi1=post[,3]/post[,5];
Zi2=post[,4]/post[,5];

newlogfun=log_fun;
/*Save values for the log-likelihood calculation*/
logfun[,1] = Zi1#initweight[,1];
logfun[,2] = Zi2#initweight[,2];
logfun[,3] = logfun[,1]+logfun[,2];
logfun[,4] = log(logfun[,3]);
newlogfun = logfun[+,4];

/*M-step:*/
/*Update the means, variances and mixing coefficients*/
newmean = initmean;
newweight = initweight;
newvar = initvar;

/*Update the mixing coefficients of each of the components*/
newweight[,1] = Zi1[+,]/N;
newweight[,2] = Zi2[+,]/N;
```

```
/*Update the means of each of the components*/
Zi1_x = Zi1#erupt;
newmean[,1] = Zi1_x[+,]/Zi1[+,]; /*mean1*/
Zi2_x = Zi2#erupt;
newmean[,2] = Zi2_x[+,]/Zi2[+,]; /*mean2*/

/*Update the variances of each of the components*/
var1_d = J(N,dim,.);
var2_d = J(N,dim,.);
do j = 1 to N;
var1_d[j,] = Zi1[j,]#((erupt[j,] - initmean[,1])##2);
var2_d[j,] = Zi2[j,]#((erupt[j,] - initmean[,2])##2);
end;
newvar[,1] = var1_d[+,]/Zi1[+,]; /*variance1*/
newvar[,2] = var2_d[+,]/Zi2[+,]; /*variance2*/

/*Stops iterations once the difference between the latest log-likelihood*/
/*and the previous one is equal to zero*/
diff=abs(log_fun-newlogfun);

/*Replace the previous parameters with the adjusted parameters*/
initmean = newmean;
initvar = newvar;
initweight = newweight;
log_fun = newlogfun;

/*Print the new parameters at the end of each Period*/
print "Iterations:" t;
print newmean;
    print newvar;
print newweight;
print newlogfun;
```

```
EM = EM//(t||newmean||newvar||newweight||newlogfun);

t=t+1;
end;

/*Printing the final maximum likelihood estimates*/
nm={"Iteration" "Mean 1" "Mean 2" "Variance 1" "Variance 2" "Weight 1" "Weight 2" "logfun"}
print EM[colname=nm];

/*Creating a dataset to plot the graphs*/
nm1={"Iterations" "mean1" "mean2" "var1" "var2" "Weight1" "Weight2" "logfun"}
create plot from EM[colname=nm1];
append from EM;

/*Graphs to plot the convergence of the parameters*/
/*Means*/
goptions reset=all;
symbol1 interpol=join width=1 color=blue;
symbol2 interpol=join width=1 color=red;
title;
title1 "Convergence of estimates for the Means";
proc gplot data=plot;
axis1 label=('Means');
axis2 label=('Iterations');
plot mean1*Iterations mean2*Iterations / vaxis=axis1 haxis=axis2 overlay legend=Legend1;
Legend1 label=('Plot')
  order=(1 to 2)
  value=('Mean1' 'Mean2')
  position=(bottom center outside);
run;

/*Variances*/
goptions reset=all;
symbol1 interpol=join width=1 color=blue;
```

```
symbol2 interpol=join width=1 color=red;
title;
title1 "Convergence of estimates for the Variances";
proc gplot data=plot;
axis1 label=('Variances');
axis2 label=('Iterations');
plot var1*Iterations var2*Iterations / vaxis=axis1 haxis=axis2 overlay legend=Legend1;
Legend1 label=('Plot')
   order=(1 to 2)
   value=('Variance1' 'Variance2')
   position=(bottom center outside);
run;

/*Mixing Coefficients*/
goptions reset=all;
symbol1 interpol=join width=1 color=blue;
symbol2 interpol=join width=1 color=red;
title;
title1 "Convergence of estimates for the Mixing Coefficients";
proc gplot data=plot;
axis1 label=('Weights');
axis2 label=('Iterations');
plot Weight1*Iterations Weight2*Iterations / vaxis=axis1 haxis=axis2 overlay legend=Legend1;
Legend1 label=('Plot')
   order=(1 to 2)
   value=('Weight1' 'Weight2')
   position=(bottom center outside);
run;

/*Graph to plot the observed data log-likelihood*/
goptions reset=all;
symbol1  interpol=join height=10pt value=circle CV=BLUE LINE=1 width=1;
title;
title1 "Log-likelihood function over 67 Iterations";
```

```
proc gplot data=plot;
plot logfun*Iterations /overlay legend=Legend1;
Legend1 label=('Plot')
   value=('Log-likelihood function')
   position=(bottom center outside);
run;

/*Check EM Algorithm using Proc FMM*/
proc fmm data=sasuser.Old2 plots=density(bins=17 width=0.25) gconv=0;
model x1= / dist=normal k=2;
run;

The EM algorithm initialized by the K-means algorithm:
/*************************************************************************/
/*K-means algorithm to initialize the EM algorithm*/
proc iml;
use sasuser.Old2;
read all into erupt;

/*Partition the data into two clusters*/
/*Note that for simplicity we choose k=2 to be the initial number of clusters*/
xclus1=erupt[1:136,]||J(136,1,1);
xclus2=erupt[137:272,]||J(136,1,2);

/*Append the groups*/
xclus = xclus1 // xclus2;
n = nrow(xclus);
x = xclus[,1];
class = xclus[,2];
x_class = x||class;

create x from x_class [colname = {'x' 'class'}];
append from x_class;
close x;
```

```
/*Choose the value of the centres of each of the initial clusters by*/
/*doing a visual inspection of the variables*/
/*Initial cluster values*/
cent1 = {5.1};
cent2 = {1.6};
initial_centres = cent1//cent2;
print initial_centres;

create ini_cent from initial_centres [colname ={'cx'}];
append from initial_centres;
close ini_cent;

/**********************Start the iteration process*******************/
centres_old = initial_centres;
k = nrow(initial_centres);
do t = 1 to 10 until (cent_diff = 0);
/*Calculate the squared Euclidean distance of each obs(row) from each centre in "centres_old"*/
/*n is nr of observations (standardized)*/
/*k is nr of centres*/
edij = J(n,k,.);
do i = 1 to n;
do j = 1 to k;
edij[i,j] = ((x[i,]-centres_old[j,])* (x[i,]-centres_old[j,])`);
end;
end;
/*The minimum value of each row indicates to which cluster the observation belongs.*/
/*Loop through each row of edij to identify the minimum value in each row */
/*and put it in a column vector clusmin, in order to classify each observation to a cluster*/

/*Initialize and reset the vectors on which we will append later*/
clusmin = {1};
clusmin = remove(clusmin,1:nrow(clusmin));
cl_class = {1};
```

```
cl_class = remove(cl_class,1:nrow(cl_class));
do h = 1 to nrow(edij);
cc = min(edij[h,]);
clusmin = clusmin // cc;
end;
do g = 1 to ncol(edij);
cl_class = cl_class || g * (edij[,g] = clusmin);
end;
cl_class = cl_class[,+];

/*Now put a column of cluster numbers next to the original observations */
/*and sort them according to the cluster numbers to calculate the new centres of the new clusters*/
x_class = x || cl_class;
call sort(x_class,{2});

/*Initialize and reset the cluster vectors*/
clus1 = {1};
clus1 = remove(clus1,1:nrow(clus1));
clus2 = {1};
clus2 = remove(clus2,1:nrow(clus2));

/*Append the observations of each cluster in separate vectors*/
/*Compile the 2 different classes*/
do i = 1 to nrow(x_class);
if x_class[i,2] = 1 then clus1 = clus1 // x_class[i,];
if x_class[i,2] = 2 then clus2 = clus2 // x_class[i,];
end;

/*Calculate the new centres of each cluster*/
cent1_new = sum(clus1[+,1])/nrow(clus1);
cent2_new= sum(clus2[+,1])/nrow(clus2);

/*Calculate the new variances of each cluster*/
var1_new=J(nrow(clus1),1,.);
```

```
var2_new=J(nrow(clus2),1,.);
do i = 1 to nrow(clus1);
do j = 1 to nrow(clus2);
var1_new[i,]=((clus1[i,1]-centres_old[1,])`*(clus1[i,1]-centres_old[1,]))/(nrow(clus1)-1);
var2_new[j,]=((clus2[j,1]-centres_old[2,])`*(clus2[j,1]-centres_old[2,]))/(nrow(clus2)-1);
end;
end;

var1_new=var1_new[+,];
var2_new=var2_new[+,];
var_new=var1_new//var2_new;

/*Calculate the new weights of each cluster*/
weight1=nrow(clus1)/n;
weight2=nrow(clus2)/n;
weights=weight1//weight2;

/*Fill a vector to indicate the size of each cluster*/
Clussize = J(2,2,.);
Clussize[1,1] = 1;
Clussize[1,2] = nrow(clus1);
Clussize[2,1] = 2;
Clussize[2,2] = nrow(clus2);

print "Loop number:" t;
print Clussize;
centres_new = cent1_new//cent2_new;
print  centres_old centres_new;
print  var_new; print  weights;

/*Calculating the difference between the new and old centres to stop the process*/
/*once the stop criterion has been met*/
cent_diff= abs(centres_old - centres_new);
print cent_diff;
```

```
/*Rename the new centres to initial centres and start the loop again*/
centres_old = centres_new;
end;
print "Final Loop:" t;
centrefinal = centres_old;
varfinal=var_new;
weightsfinal=weights;
print centrefinal varfinal weights;

/*The Expectation Maximization Algorithm*/
/*Number of columns and rows*/
dim=ncol(erupt);
N=nrow(erupt);

/*Number of Components (Assumed to be Known)*/
K=2;

/*Initial values generated by the K-means algorithm*/
/*Means*/
initmean=J(dim,K,.);
initmean[,1]=centrefinal[2,];
initmean[,2]=centrefinal[1,];

/*Variances*/
initvar=J(dim,K,.);
initvar[,1]=varfinal[2,];
initvar[,2]=varfinal[1,];

/*Mixing Coefficients*/
initweight=J(1,K,.);
initweight[,1]=weights[2,];
initweight[,2]=weights[1,];
```

```
/*Log-likelihood function*/
log_fun=-2800;

/*PI constant*/
pi=constant("pi");

/*Start the iterations*/
t=0;
do until(diff = 0);
/*E-step:*/
post = J(N,5,.);
logfun = J(N,4,.);

/*Calculating the pdfs for each component*/
do i = 1 to N;
do j = 1 to K;
post[i,j] = 1/((2#pi#initvar[,j])##(dim/2))*exp((-1/(2#initvar[,j]))#((erupt[i,]-initmean[,j])##2));
end;
end;

/*Calculating the pdfs*weights*/
post[,3]=initweight[,1]#post[,1];
post[,4]=initweight[,2]#post[,2];

/*Sum prior*pdf over K1 and K2*/
post[,5]=post[,3]+post[,4];

/*Calculating of the Responsibilities*/
Zi1=post[,3]/post[,5];
Zi2=post[,4]/post[,5];

newlogfun=log_fun;
/*Save values for the log-likelihood calculation*/
logfun[,1] = Zi1#initweight[,1];
```

```
logfun[,2] = Zi2#initweight[,2];
logfun[,3] = logfun[,1]+logfun[,2];
logfun[,4] = log(logfun[,3]);
newlogfun = logfun[+,4];

/*M-step:*/
/*Update the means, variances and mixing coefficients*/
newmean = initmean;
newweight = initweight;
newvar = initvar;

/*Update the mixing coefficients of each of the components*/
newweight[,1] = Zi1[+,]/N;
newweight[,2] = Zi2[+,]/N;

/*Update the means of each of the components*/
Zi1_x = Zi1#erupt;
newmean[,1] = Zi1_x[+,]/Zi1[+,]; /*mean1*/
Zi2_x = Zi2#erupt;
newmean[,2] = Zi2_x[+,]/Zi2[+,]; /*mean2*/

/*Update the variances of each of the components*/
var1_d = J(N,dim,.);
var2_d = J(N,dim,.);
do j = 1 to N;
var1_d[j,] = Zi1[j,]#((erupt[j,] - initmean[,1])##2);
var2_d[j,] = Zi2[j,]#((erupt[j,] - initmean[,2])##2);
end;
newvar[,1] = var1_d[+,]/Zi1[+,]; /*variance1*/
newvar[,2] = var2_d[+,]/Zi2[+,]; /*variance2*/

/*Stops iterations once the difference between the latest log-likelihood*/
/*and the previous one is equal to zero*/
diff=abs(log_fun-newlogfun);
```

```
/*Replace the previous parameters with the adjusted parameters*/
initmean = newmean;
initvar = newvar;
initweight = newweight;
log_fun = newlogfun;

/*Print the new parameters at the end of each Period*/
print "Iterations:" t;
print newmean;
print newvar;
print newweight;
print newlogfun;

EM = EM//(t||newmean||newvar||newweight||newlogfun);

t=t+1;
end;
nm={"Iteration" "Mean 1" "Mean 2" "Variance 1" "Variance 2" "Weight 1" "Weight 2" "logfun"}
print EM[colname=nm];

/*Creating a dataset to plot the graphs*/
nm1={"Iteration" "mean1" "mean2" "var1" "var2" "Weight1" "Weight2" "logfun"}
create plot from EM[colname=nm1];
append from EM;
close plot;

/*Graphs to plot the convergence of the parameters*/
/*Means*/
goptions reset=all;
symbol1 interpol=join width=1 color=blue;
symbol2 interpol=join width=1 color=red;
title;
title1 "Convergence of estimates for the Means";
```

```
proc gplot data=plot;
axis1 label=('Means');
axis2 label=('Iterations');
plot mean1*Iteration mean2*Iteration / vaxis=axis1 haxis=axis2 overlay legend=Legend1;
Legend1 label=('Plot')
   order=(1 to 2)
    value=('Mean1' 'Mean2')
   position=(bottom center outside);
run;

/*Variances*/
goptions reset=all;
symbol1 interpol=join width=1 color=blue;
symbol2 interpol=join width=1 color=red;
title;
title1 "Convergence of estimates for the Variances";
proc gplot data=plot;
axis1 label=('Variances');
axis2 label=('Iterations');
plot var1*Iteration var2*Iteration / vaxis=axis1 haxis=axis2 overlay legend=Legend1;
Legend1 label=('Plot')
        order=(1 to 2)
                    value=('Variance1' 'Variance2')
    position=(bottom center outside);
run;

/*Mixing coefficients*/
goptions reset=all;
symbol1 interpol=join width=1 color=blue;
symbol2 interpol=join width=1 color=red;
title;
title1 "Convergence of estimates for the Weights";
proc gplot data=plot;
axis1 label=('Weights');
```

```
axis2 label=('Iterations');
plot Weight1*Iteration Weight2*Iteration / vaxis=axis1 haxis=axis2 overlay legend=Legend1;
Legend1 label=('Plot')
    order=(1 to 2)
value=('Weight1' 'Weight2')
position=(bottom center outside);
run;

/*Graph to plot the observed data log-likelihood*/
goptions reset=all;
symbol1  interpol=join height=10pt value=circle CV=BLUE LINE=1 width=1;
title;
title1 "Log-likelihood over 58 Iterations";
proc gplot data=plot;
plot logfun*Iteration /overlay legend=Legend1;
Legend1 label=('Plot')
value=('Log-likelihood function')
position=(bottom center outside);
run;

/*Check K-means clustering using Proc Fastclus*/
proc fastclus data=sasuser.Old2 maxclusters=2 maxiter=10 ;
var x1;
run;

The CEM algorithm with random selection:
/*****************************************************************/
proc iml;
use sasuser.Old2;
read all into erupt;
print erupt;

/*Number of rows and columns*/
dim=ncol(erupt);
```

```
N=nrow(erupt);

/*Number of Components (Assumed to be Known)*/
K=2;

/*Random selection of initial values for the mean, variance and mixing coefficients*/
/*Mean*/
initmean=J(dim,K,.);
initmean[,1]=2.2;
initmean[,2]=4.15;

/*Variance*/
initvar=J(dim,K,.);
do j = 1 to K;
initvar[,j]=1.7;
end;

/*Mixing Coefficients*/
initweight=J(1,K,.);
do j = 1 to K;
initweight[,j]=0.5;
end;

/*Log-likelihood function*/
log_fun=-28000;

/*PI constant*/
pi=constant("pi");

/*Start the iterations*/
t=0;
do until(diff = 0);
/*E-step:*/
logfun = J(N,4,.);
```

```
post=J(N,5,.);

/*Calculating the pdfs for each component*/
do i = 1 to N;
do j = 1 to K;
post[i,j] = 1/((2#pi#initvar[,j])##(dim/2))*exp((-1/(2#initvar[,j]))#((erupt[i,]-initmean[,j])##2));
end;
end;

/*Calculating the pdfs*weights*/
post[,3]=initweight[,1]#post[,1];
post[,4]=initweight[,2]#post[,2];

/*Sum prior*pdf over K1 and K2*/
post[,5]=post[,3]+post[,4];

/*Calculating of the Responsibilities*/
Zi1=post[,3]/post[,5];
Zi2=post[,4]/post[,5];

newlogfun=log_fun;
/*Save values for the log-likelihood calculation*/
logfun[,1] = Zi1#initweight[,1];
logfun[,2] = Zi2#initweight[,2];
logfun[,3] = logfun[,1]+logfun[,2];
logfun[,4] = log(logfun[,3]);
newlogfun = logfun[+,4];

/*C-step:*/
P=Zi1||Zi2;
class = P[,<:>];
pos1 = loc(class=1) ;
pos2 = loc(class=2) ;
x_full = erupt||P||class;
```

```
Xi1 = x_full[pos1,] ;
Xi2 = x_full[pos2,] ;

/*Counting the number of rows of each partition*/
x1=nrow(Xi1);
x2=nrow(Xi2);

/*M-step:*/
/*Update the means, variances and mixing coefficients*/
newmean = initmean;
newweight = initweight;
newvar = initvar;

/*Update the mixing coefficients of each of the components*/
newweight[,1] = x1/N;
newweight[,2] = x2/N;

/*Update the means of each of the components*/
Xi1_x = Xi1[,1]#Xi1[,2];
newmean[,1] = Xi1_x[+,]/Xi1[+,2]; /*mean1*/
Xi2_x = Xi2[,1]#Xi2[,3];
newmean[,2] = Xi2_x[+,]/Xi2[+,3]; /*mean2*/

/*Update the variances of each of the components*/
var1_d = J(x1,dim,.);
var2_d = J(x2,dim,.);
do i = 1 to x1;
do j = 1 to x2;
var1_d[i,] = Xi1[i,2]#((Xi1[i,1] - initmean[,1])##2);
var2_d[j,] = Xi2[j,3]#((Xi2[j,1] - initmean[,2])##2);
end;
end;
newvar[,1] = var1_d[+,]/Xi1[+,2]; /*variance1*/
```

```
newvar[,2] = var2_d[+,]/Xi2[+,3]; /*variance2*/

/*Stops iterations once the difference between the latest log-likelihood*/
/*and the previous one is equal to zero*/
diff=abs(log_fun-newlogfun);

/*Replace the previous parameters with the adjusted parameters*/
initmean = newmean;
initvar = newvar;
initweight = newweight;
log_fun = newlogfun;

/*Print the new parameters at the end of each Period*/
print "Iteration:" t;
print newmean;
print newvar;
print newweight;
print newlogfun;

CEM = CEM//(t||newmean||newvar||newweight||newlogfun);

t=t+1;
end;
nm={"Iteration" "Mean 1" "Mean 2" "Variance 1" "Variance 2" "Weight 1" "Weight 2" "logfun"}
print CEM[colname=nm];

/*Creating a dataset to plot the graphs*/
nm1={"Iteration" "mean1" "mean2" "var1" "var2" "Weight1" "Weight2" "logfun"}
create plot from CEM[colname=nm1];
append from CEM;
close plot;

/*Graphs to plot the convergence of the parameters*/
/*Means*/
```

```
goptions reset=all;
symbol1 interpol=join width=1 color=blue;
symbol2 interpol=join width=1 color=red;
title;
title1 "Convergence of estimates for the Means";
proc gplot data=plot;
axis1 label=('Means');
axis2 label=('Iterations');
plot mean1*Iteration mean2*Iteration / vaxis=axis1 haxis=axis2 overlay legend=Legend1;
Legend1 label=('Plot')
order=(1 to 2)
value=('Mean1' 'Mean2')
position=(bottom center outside);
run;

/*Variances*/
goptions reset=all;
symbol1 interpol=join width=1 color=blue;
symbol2 interpol=join width=1 color=red;
title;
title1 "Convergence of estimates for the Variances";
proc gplot data=plot;
axis1 label=('Variances');
axis2 label=('Iterations');
plot var1*Iteration var2*Iteration / vaxis=axis1 haxis=axis2 overlay legend=Legend1;
Legend1 label=('Plot')
order=(1 to 2)
value=('Variance1' 'Variance2')
position=(bottom center outside);
run;

/*Mixing Coefficients*/
goptions reset=all;
symbol1 interpol=join width=1 color=blue;
```

```
symbol2 interpol=join width=1 color=red;
title;
title1 "Convergence of estimates for the Mixing Coefficients";
proc gplot data=plot;
axis1 label=('Weights');
axis2 label=('Iterations');
plot Weight1*Iteration Weight2*Iteration / vaxis=axis1 haxis=axis2 overlay legend=Legend1;
Legend1 label=('Plot')
order=(1 to 2)
value=('Weight1' 'Weight2')
position=(bottom center outside);
run;

/*Graph to plot the observed data log-likelihood*/
goptions reset=all;
symbol1  interpol=join height=10pt value=circle CV=BLUE LINE=1 width=1;
title;
title1 "Log-likelihood over 29 Iterations";
proc gplot data=plot;
plot logfun*Iteration /overlay legend=Legend1;
Legend1 label=('Plot')
value=('Log-likelihood function')
position=(bottom center outside);
run;

The CEM algorithm initialized by the K-means algorithm:
/*****************************************************************/
/*K-means algorithm to initialize the CEM algorithm*/
proc iml;
use sasuser.Old2;
read all into erupt;

/*Partition the data into two clusters*/
/*Note that for simplicity we choose k=2 to be the initial number of clusters*/
```

```
xclus1=erupt[1:136,]||J(136,1,1);
xclus2=erupt[137:272,]||J(136,1,2);

/*Append the groups*/
xclus = xclus1 // xclus2;

n1 = nrow(xclus);
x = xclus[,1];
class = xclus[,2];
x_class = x||class;

create x from x_class [colname = {'x' 'class'}];
append from x_class;
close x;

/*Choose the value of the centres of each of the initial clusters by*/
/*doing a visual inspection of the variables*/
/*Initial cluster values*/
cent1 = {5.1};
cent2 = {1.6};

initial_centres = cent1//cent2;
print initial_centres;

create ini_cent from initial_centres [colname ={'cx'}];
append from initial_centres;
close ini_cent;

/**********************Start the iteration process********************/
centres_old = initial_centres;
k = nrow(initial_centres);
do t = 1 to 10 until (cent_diff = 0);
/*Calculate the squared Euclidean distance of each obs(row) from each centre in "centres_old"*/
/*n is nr of observations (standardized)*/
```

```
/*k is nr of centres*/
edij = J(n1,k,.);
do i = 1 to n1;
do j = 1 to k;
edij[i,j] = ((x[i,]-centres_old[j,])* (x[i,]-centres_old[j,])‘);
end;
end;

/*The minimum value of each row indicates to which cluster the observation belongs.*/
/*Loop through each row of edij to identify the minimum value in each row */
/*and put it in a column vector clusmin, in order to classify each observation to a cluster*/

/*Initialize and reset the vectors on which we will append later*/
clusmin = {1};
clusmin = remove(clusmin,1:nrow(clusmin));
cl_class = {1};
cl_class = remove(cl_class,1:nrow(cl_class));
do h = 1 to nrow(edij);
cc = min(edij[h,]);
clusmin = clusmin // cc;
end;

do g = 1 to ncol(edij);
cl_class = cl_class || g * (edij[,g] = clusmin);
end;
cl_class = cl_class[,+];
/*Now put a column of cluster numbers next to the original observations*/
/*and sort them according to the cluster numbers to calculate the new centres of the new clusters*/
x_class = x || cl_class;
call sort(x_class,{2});

/*Initialize and reset the cluster vectors*/
clus1 = {1};
clus1 = remove(clus1,1:nrow(clus1));
```

```
clus2 = {1};
clus2 = remove(clus2,1:nrow(clus2));

/*Append the observations of each cluster in separate vectors*/
/*Compile the 2 different classes*/
do i = 1 to nrow(x_class);
if x_class[i,2] = 1 then clus1 = clus1 // x_class[i,];
if x_class[i,2] = 2 then clus2 = clus2 // x_class[i,];
end;

/*Calculate the new centres of each cluster*/
cent1_new = sum(clus1[+,1])/nrow(clus1);
cent2_new= sum(clus2[+,1])/nrow(clus2);

/*Calculate the new variances of each cluster*/
var1_new=J(nrow(clus1),1,.);
var2_new=J(nrow(clus2),1,.);
do i = 1 to nrow(clus1);
do j = 1 to nrow(clus2);
var1_new[i,]=((clus1[i,1]-centres_old[1,])‘*(clus1[i,1]-centres_old[1,]))/(nrow(clus1)-1);
var2_new[j,]=((clus2[j,1]-centres_old[2,])‘*(clus2[j,1]-centres_old[2,]))/(nrow(clus2)-1);
end;
end;
var1_new=var1_new[+,];
var2_new=var2_new[+,];
var_new=var1_new//var2_new;

/*Calculate the new weights of each cluster*/
weight1=nrow(clus1)/n1;
weight2=nrow(clus2)/n1;
weights=weight1//weight2;

/*Fill a vector to indicate the size of each cluster*/
Clussize = J(2,2,.);
```

```
Clussize[1,1] = 1;
Clussize[1,2] = nrow(clus1);
Clussize[2,1] = 2;
Clussize[2,2] = nrow(clus2);

print "Loop number:" t;
print Clussize;
centres_new = cent1_new//cent2_new;
print  centres_old centres_new;
print  var_new;
print  weights;

/*Calculating the difference between the new and old centres to stop*/
/*the process once the stop criterion has been met*/
cent_diff= abs(centres_old - centres_new);
print cent_diff;

/*Rename the new centres to initial centres and start the loop again*/
centres_old = centres_new;
end;
print "Final Loop:" t;
centrefinal = centres_old;
varfinal=var_new;
weightsfinal=weights;
print centrefinal varfinal weights;

/*The Classification Expectation-Maximization (CEM) Algorithm*/
/*Number of rows and columns*/
dim=ncol(erupt);
N=nrow(erupt);

/*Number of Components (Assumed to be Known)*/
K=2;
```

```
/*Initial values generated by the K-means algorithm*/
/*Means*/
initmean=J(dim,K,.);
initmean[,1]=centrefinal[2,];
initmean[,2]=centrefinal[1,];

/*Variances*/
initvar=J(dim,K,.);
initvar[,1]=varfinal[2,];
initvar[,2]=varfinal[1,];

/*Mixing Coefficients*/
initweight=J(1,K,.);
initweight[,1]=weights[2,];
initweight[,2]=weights[1,];

/*Log-likelihood function*/
log_fun=-180000;

/*PI constant*/
pi=constant("pi");

/*Start the iterations*/
t=0;
do until(diff=0);
/*E-step:*/
logfun = J(N,4,.);
post=J(N,5,.);

/*Calculating the pdfs for each component*/
do i = 1 to N;
do j = 1 to K;
post[i,j] = 1/((2#pi#initvar[,j])##(dim/2))*exp((-1/(2#initvar[,j]))#((erupt[i,]-initmean[,j])##2));
end;
```

```
end;

/*Calculating the pdfs*weights*/
post[,3]=initweight[,1]#post[,1];
post[,4]=initweight[,2]#post[,2];

/*Sum prior*pdf over K1 and K2*/
post[,5]=post[,3]+post[,4];

/*Calculating of the Responsibilities*/
Zi1=post[,3]/post[,5];
Zi2=post[,4]/post[,5];

newlogfun=log_fun;
/*Save values for the log-likelihood calculation*/
logfun[,1] = Zi1#initweight[,1];
logfun[,2] = Zi2#initweight[,2];
logfun[,3] = logfun[,1]+logfun[,2];
logfun[,4] = log(logfun[,3]);
newlogfun = logfun[+,4];

/*C-step:*/
P=Zi1||Zi2;
classC = P[,<:>];
pos1 = loc(classC=1);
pos2 = loc(classC=2);
x_full = erupt||P||classC;

Xi1 = x_full[pos1,];
Xi2 = x_full[pos2,];

/*Counting the number of rows of each partition*/
x1=nrow(Xi1);
x2=nrow(Xi2);
```

```
/*M-step:*/
/*Update the means, variances and mixing coefficients*/
newmean = initmean;
newweight = initweight;
newvar = initvar;

/*Update the mixing coefficients of each of the components*/
newweight[,1] = x1/N;
newweight[,2] = x2/N;

/*Update the means of each of the components*/
Xi1_x = Xi1[,1]#Xi1[,2];
newmean[,1] = Xi1_x[+,]/Xi1[+,2]; /*mean1*/
Xi2_x = Xi2[,1]#Xi2[,3];
newmean[,2] = Xi2_x[+,]/Xi2[+,3]; /*mean2*/

/*Update the variances of each of the components*/
var1_d = J(x1,dim,.);
var2_d = J(x2,dim,.);
do i = 1 to x1;
do j = 1 to x2;
var1_d[i,] = Xi1[i,2]#((Xi1[i,1] - initmean[,1])##2);
var2_d[j,] = Xi2[j,3]#((Xi2[j,1] - initmean[,2])##2);
end;
end;
newvar[,1] = var1_d[+,]/Xi1[+,2]; /*variance1*/
newvar[,2] = var2_d[+,]/Xi2[+,3]; /*variance2*/

/*Stops iterations once the difference between the latest log-likelihood*/
/*and the previous one is equal to zero*/
diff=abs(log_fun-newlogfun);

/*Replace the previous parameters with the adjusted parameters*/
```

```
initmean = newmean;
initvar = newvar;
initweight = newweight;
log_fun = newlogfun;

/*Print the new parameters at the end of each Period*/
print "Iteration:" t;
print newmean;
print newvar;
print newweight;
print newlogfun;

CEM = CEM//(t||newmean||newvar||newweight||newlogfun);

t=t+1;
end;
nm={"Iteration" "Mean 1" "Mean 2" "Variance 1" "Variance 2" "Weight 1" "Weight 2" "logfun"}
print CEM[colname=nm];

/*Creating a dataset to plot the graphs*/
nm1={"Iteration" "mean1" "mean2" "var1" "var2" "Weight1" "Weight2" "logfun"}
create plot from CEM[colname=];
append from CEM;
close plot;

/*Graphs to plot the convergence of the parameters*/
/*Means*/
goptions reset=all;
symbol1 interpol=join width=1 color=blue;
symbol2 interpol=join width=1 color=red;
title;
title1 "Convergence of estimates for the Means";
proc gplot data=plot;
axis1 label=('Means');
```

```
axis2 label=('Iterations');
plot mean1*Iteration mean2*Iteration / vaxis=axis1 haxis=axis2 overlay legend=Legend1;
Legend1 label=('Plot')
order=(1 to 2)
value=('Mean1' 'Mean2')
position=(bottom center outside);
run;

/*Variances*/
goptions reset=all;
symbol1 interpol=join width=1 color=blue;
symbol2 interpol=join width=1 color=red;
title;
title1 "Convergence of estimates for the Variances";
proc gplot data=plot;
axis1 label=('Variances');
axis2 label=('Iterations');
plot var1*Iteration var2*Iteration / vaxis=axis1 haxis=axis2 overlay legend=Legend1;
Legend1 label=('Plot')
order=(1 to 2)
value=('Variance1' 'Variance2')
position=(bottom center outside);
run;

/*Mixing coefficients*/
goptions reset=all;
symbol1 interpol=join width=1 color=blue;
symbol2 interpol=join width=1 color=red;
title;
title1 "Convergence of estimates for the Weights";
proc gplot data=plot;
axis1 label=('Weights');
axis2 label=('Iterations');
plot Weight1*Iteration Weight2*Iteration / vaxis=axis1 haxis=axis2 overlay legend=Legend1;
```

```
Legend1 label=('Plot')
order=(1 to 2)
value=('Weight1' 'Weight2')
position=(bottom center outside);
run;

/*Graph to plot the observed data log-likelihood*/
goptions reset=all;
symbol1  interpol=join height=10pt value=circle CV=BLUE LINE=1 width=1;
title;
title1 "Log-likelihood over 28 Iterations";
proc gplot data=plot;
plot logfun*Iteration /overlay legend=Legend1;
Legend1 label=('Plot')
value=('Log-likelihood function')
position=(bottom center outside);
run;

EM simulation:
/*****************************************************************/
proc iml ;
start GEN;
pi1 = mat[kk,7];
pi2 = mat[kk,8];
delta = mat[kk,3];
m1 = mat[kk,2];
m2 = m1+delta;
v1 = mat[kk,5];
v2 = mat[kk,6];
n = mat[kk,4];

n1 = round(n*pi1);
n2 = n-n1;
```

```
sd1 = J(n1,1,0);
sd2 = J(n2,1,0);

x1=rannor(sd1)*sqrt(v1)+m1;
x2=rannor(sd2)*sqrt(v2)+m2;
x = x1 // x2;
finish GEN;

start EM;
call GEN;
/*Sort the variable x*/
call sort(x,{1});

/*Determine the quartiles and keep only the median - initial value for pi1=0.5*/
call qntl(q,x);
qpi = q[2];

*Identify values in x below and above the median;
*essentially  splitting the data  into k=2 groups;
mx = x#(x<=qpi) || x#(x>qpi);

*Random selection of initial values for the mean, variance and mixing coefficients;
*Mean 1;
mean1 = loc(mx[,1]^=0);
mean1 = (mx[,1])[m1];
mean1 = mean1[:];

*Mean 2;
mean2 = loc(mx[,2]^=0);
mean2 = (mx[,2])[mean2];
mean2 = mean2[:];

*Variance;
variance1 = var(x);
```

```
variance2 = variance1;

*Mixing Coefficients;
pi1=0.5;
pi2=1-pi1;

*The Stopping Criterion;
StopCriterion = 0.0001;

*Initializing the log-likelihood function;
logfun = -100000000000000000000;

*Initializing the difference;
diff = StopCriterion+1;

do i = 1 to 50 while (diff>StopCriterion);
s1=sqrt(variance1);
s2=sqrt(variance2);

/*E-step:*/
/*Calculating the normal densities*/
normd1 = pdf("Normal",x,mean1,s1);
normd2 = pdf("Normal",x,mean2,s2);

/*Calculating the pdfs*weights*/
post = pi1*normd1 || pi2*normd2;
post = post / post[,+];

/*Calculating of the Responsibilities*/
Zi1=post[,1];
Zi2=post[,2];

/*M-step:*/
/*Update the means of each of the components*/
```

```
mean1 = sum(Zi1#x) / sum(Zi1);
mean2 = sum(Zi2#x) / sum(Zi2);

/*Update the variances of each of the components*/
v1 = sum(Zi1 # (x-mean1)##2)   / sum(Zi1);
v2 = sum(Zi2 # (x-mean2)##2)   / sum(Zi2);

/*Update the mixing coefficients of each of the components*/
pi1 = (Zi1)[:];
pi2 = (Zi2)[:];

/*Calculating the log-likelihood function*/
newlogfun = sum(log(pi1*normd1+pi2*normd2));

EM = EM // (i || mean1 || mean2 || variance1 || variance2 || pi1 || newlogfun);
/*Stops iterations once the difference between the latest log-likelihood*/
/*and the previous one is greater than 0.0001*/
diff = abs(logfun-newlogfun);
logfun=newlogfun;
end;
finish EM;

a=J(5,1,500);
b=J(5,1,40);
c=j(5,1,50);
d=J(5,1,16);
e=J(5,1,16);
f=J(5,1,0.5);
g=1-f;

mat = a||b||c||{30,50,100,200,500}||d||e||f||g;
do kk =  1 to nrow(mat);
call GEN;
do jj= 1 to 500;
```

```
call EM;
simulation = simulation // (jj || i-1 || mean1 || mean2 || variance1 || variance2 || pi1);
free EM;
end;
nm={"n" "Iterations" "Mean 1" "Mean 2" "Variance 1" "Variance 2" "pi1"};
average = average // (simulation[:,2:7]);
end;
nn={30,50,100,200,500};
ave=nn||average;
print  ave[colname=nm];

bmu1=40;
delta=50;
bmu2=bmu1+delta;
bv1=16;
bv2=bv1;
bmix=0.5;

Bias = Bias//(ave[,1]||ave[,2]||abs(bmu1-ave[,3])||abs(bmu2-ave[,4])||abs(bv1-ave[,5])
||abs(bv2-ave[,6])||abs(bmix-ave[,7]));
print Bias[colname={"n" "Iterations" "Mean 1" "Mean 2" "Variance 1" "Variance 2" "pi1"}];

data plot;
set plot1 plot2 plot3 plot4 plot5 plot6 plot7 plot8 plot9;
run;
quit;

/*Iterations*/
proc template;
define statgraph sgdesign;
dynamic _N _ITER1A _N2 _ITER2A _N3 _ITER3A _N4 _ITER4A _N5 _ITER5A _N6 _ITER6A _N7 _ITER7A
_N8 _ITER8A _N9 _ITER9A;
begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
```

```
layout overlay / xaxisopts=( label=('Sample size')) yaxisopts=( label=('Iterations'));
seriesplot x=_N y=_ITER1A / name='series' legendlabel='Delta = 15 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N2 y=_ITER2A / name='series2' legendlabel='Delta = 15 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N3 y=_ITER3A / name='series3' legendlabel='Delta = 15 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N4 y=_ITER4A / name='series4' legendlabel='Delta = 30 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N5 y=_ITER5A / name='series5' legendlabel='Delta = 30 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N6 y=_ITER6A / name='series6' legendlabel='Delta = 30 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N7 y=_ITER7A / name='series7' legendlabel='Delta = 50 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
seriesplot x=_N8 y=_ITER8A / name='series8' legendlabel='Delta = 50 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
seriesplot x=_N9 y=_ITER9A / name='series9' legendlabel='Delta = 50 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
discretelegend 'series' 'series2' 'series3' 'series4' 'series5' 'series6' 'series7' 'series8'
'series9'/opaque=false border=true valign=bottom displayclipped=true down=1 order=rowmajor
location=outside;
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.PLOT template=sgdesign;
dynamic _N="N" _ITER1A="ITER1" _N2="N" _ITER2A="ITER2" _N3="N" _ITER3A="ITER3"
  _N4="N" _ITER4A="ITER4" _N5="N" _ITER5A="ITER5" _N6="N"
  _ITER6A="ITER6" _N7="N" _ITER7A="ITER7" _N8="N" _ITER8A="ITER8"
  _N9="N" _ITER9A="ITER9";
run;
```

```
/*Mean 1*/
proc template;
define statgraph sgdesign;
dynamic _N _M11A _N2 _M12A _N3 _M13A _N4 _M14A _N5 _M15A _N6 _M16A _N7 _M17A
_N8 _M18A _N9 _M19A;
begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
layout overlay / xaxisopts=( label=('Sample size')) yaxisopts=( label=('Mean'));
seriesplot x=_N y=_M11A / name='series' legendlabel='Delta = 15 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N2 y=_M12A / name='series2' legendlabel='Delta = 15 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N3 y=_M13A / name='series3' legendlabel='Delta = 15 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N4 y=_M14A / name='series4' legendlabel='Delta = 30 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N5 y=_M15A / name='series5' legendlabel='Delta = 30 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N6 y=_M16A / name='series6' legendlabel='Delta = 30 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N7 y=_M17A / name='series7' legendlabel='Delta = 50 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
seriesplot x=_N8 y=_M18A / name='series8' legendlabel='Delta = 50 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
seriesplot x=_N9 y=_M19A / name='series9' legendlabel='Delta = 50 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
discretelegend 'series' 'series2' 'series3' 'series4' 'series5' 'series6' 'series7' 'series8'
'series9'/opaque=false border=true valign=bottom displayclipped=true down=1 order=rowmajor
location=outside;
endlayout;
endlayout;
endgraph;
end;
```

```
run;

proc sgrender data=WORK.PLOT template=sgdesign;
dynamic _N="N" _M11A="M11" _N2="N" _M12A="M12" _N3="N" _M13A="M13" _N4="N" _M14A="M14"
_N5="N" _M15A="M15" _N6="N" _M16A="M16" _N7="N" _M17A="M17" _N8="N"
_M18A="M18" _N9="N" _M19A="M19";
run;

/*Mean 2*/
proc template;
define statgraph sgdesign;
dynamic _N _M21A _N2 _M22A _N3 _M23A _N4 _M24A _N5 _M25A _N6 _M26A _N7 _M27A
_N8 _M28A _N9 _M29A;
begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
layout overlay / xaxisopts=( label=('Sample size')) yaxisopts=( label=('Mean'));
seriesplot x=_N y=_M21A / name='series' legendlabel='Delta = 15 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N2 y=_M22A / name='series2' legendlabel='Delta = 15 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N3 y=_M23A / name='series3' legendlabel='Delta = 15 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N4 y=_M24A / name='series4' legendlabel='Delta = 30 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N5 y=_M25A / name='series5' legendlabel='Delta = 30 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N6 y=_M26A / name='series6' legendlabel='Delta = 30 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N7 y=_M27A / name='series7' legendlabel='Delta = 50 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
seriesplot x=_N8 y=_M28A / name='series8' legendlabel='Delta = 50 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
seriesplot x=_N9 y=_M29A / name='series9' legendlabel='Delta = 50 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
```

```
discretelegend 'series' 'series2' 'series3' 'series4' 'series5' 'series6' 'series7' 'series8'
'series9'/opaque=false border=true valign=bottom displayclipped=true down=1 order=rowmajor
location=outside;
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.PLOT template=sgdesign;
dynamic _N="N" _M21A="M21" _N2="N" _M22A="M22" _N3="N" _M23A="M23" _N4="N" _M24A="M24"
_N5="N" _M25A="M25" _N6="N" _M26A="M26" _N7="N" _M27A="M27" _N8="N" _M28A="M28"
_N9="N" _M29A="M29";
run;

/*Variance 1*/
proc template;
define statgraph sgdesign;
dynamic _N _V11A _N2 _V12A _N3 _V13A _N4 _V14A _N5 _V16A _N6 _V17A _N7 _V18A
_N8 _V15A _N9 _V19A;
begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
layout overlay / xaxisopts=( label=('Sample size')) yaxisopts=( label=('Variance'));
seriesplot x=_N y=_V11A / name='series' legendlabel='Delta = 15 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N2 y=_V12A / name='series2' legendlabel='Delta = 15 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N3 y=_V13A / name='series3' legendlabel='Delta = 15 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N4 y=_V14A / name='series4' legendlabel='Delta = 30 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N5 y=_V16A / name='series5' legendlabel='Delta = 30 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N6 y=_V17A / name='series6' legendlabel='Delta = 30 (pi = 0.5)'
```

```
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
seriesplot x=_N7 y=_V18A / name='series7' legendlabel='Delta = 50 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
seriesplot x=_N8 y=_V15A / name='series8' legendlabel='Delta = 50 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N9 y=_V19A / name='series9' legendlabel='Delta = 50 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
discretelegend 'series' 'series2' 'series3' 'series4' 'series5' 'series6' 'series7' 'series8'
'series9'/opaque=false border=true valign=bottom displayclipped=true down=1 order=rowmajor
location=outside;
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.PLOT template=sgdesign;
dynamic _N="N" _V11A="V11" _N2="N" _V12A="V12" _N3="N" _V13A="V13" _N4="N" _V14A="V14"
_N5="N" _V16A="V16" _N6="N" _V17A="V17" _N7="N" _V18A="V18" _N8="N" _V15A="V15"
_N9="N" _V19A="V19";
run;

/*Variance 2*/
proc template;
define statgraph sgdesign;
dynamic _N _V21A _N2 _V22A _N3 _V23A _N4 _V24A _N5 _V25A _N6 _V26A _N7 _V27A _N8
_V28A _N9 _V29A;
begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
layout overlay / xaxisopts=( label=('Sample size')) yaxisopts=( label=('Variance'));
seriesplot x=_N y=_V21A / name='series' legendlabel='Delta = 15 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N2 y=_V22A / name='series2' legendlabel='Delta = 15 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
```

```
seriesplot x=_N3 y=_V23A / name='series3' legendlabel='Delta = 15 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N4 y=_V24A / name='series4' legendlabel='Delta = 30 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N5 y=_V25A / name='series5' legendlabel='Delta = 30 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N6 y=_V26A / name='series6' legendlabel='Delta = 30 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N7 y=_V27A / name='series7' legendlabel='Delta = 50 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
seriesplot x=_N8 y=_V28A / name='series8' legendlabel='Delta = 50 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
seriesplot x=_N9 y=_V29A / name='series9' legendlabel='Delta = 50 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
discretelegend 'series' 'series2' 'series3' 'series4' 'series5' 'series6' 'series7' 'series8'
'series9'/opaque=false border=true valign=bottom displayclipped=true down=1 order=rowmajor
location=outside;
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.PLOT template=sgdesign;
dynamic _N="N" _V21A="V21" _N2="N" _V22A="V22" _N3="N" _V23A="V23" _N4="N" _V24A="V24"
_N5="N" _V25A="V25" _N6="N" _V26A="V26" _N7="N" _V27A="V27" _N8="N" _V28A="V28"
_N9="N" _V29A="V29";
run;

/*Mixing Coefficients*/
proc template;
define statgraph sgdesign;
dynamic _N _PI11A _N2 _PI12A _N3 _PI13A _N4 _PI14A _N5 _PI15A _N6 _PI16A _N7 _PI17A
_N8 _PI18A _N9 _PI19A;
```

```
begingraph;    layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
layout overlay / xaxisopts=( label=('Sample size')) yaxisopts=( label=('Mixing Coefficient'));
seriesplot x=_N y=_PI11A / name='series' legendlabel='Delta = 15 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N2 y=_PI12A / name='series2' legendlabel='Delta = 15 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N3 y=_PI13A / name='series3' legendlabel='Delta = 15 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=MEDIUMDASH thickness=2 );
seriesplot x=_N4 y=_PI14A / name='series4' legendlabel='Delta = 30 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N5 y=_PI15A / name='series5' legendlabel='Delta = 30 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N6 y=_PI16A / name='series6' legendlabel='Delta = 30 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_N7 y=_PI17A / name='series7' legendlabel='Delta = 50 (pi = 0.2)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
seriesplot x=_N8 y=_PI18A / name='series8' legendlabel='Delta = 50 (pi = 0.4)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
seriesplot x=_N9 y=_PI19A / name='series9' legendlabel='Delta = 50 (pi = 0.5)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=DASHDOTDOT thickness=2 );
discretelegend 'series' 'series2' 'series3' 'series4' 'series5' 'series6' 'series7' 'series8'
'series9'/opaque=false border=true valign=bottom displayclipped=true down=1 order=rowmajor
location=outside;
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.PLOT template=sgdesign;
dynamic _N="N" _PI11A="PI11" _N2="N" _PI12A="PI12" _N3="N" _PI13A="PI13" _N4="N" _PI14A="PI14"
_N5="N" _PI15A="PI15" _N6="N" _PI16A="PI16" _N7="N" _PI17A="PI17" _N8="N" _PI18A="PI18"
_N9="N" _PI19A="PI19";
run;
```

CEM simulation:

```
/*************************************************************************/
data sasuser.CEMSim;
run;
%macro cem;
proc iml;
start GEN;
n=30; *Will be changed - n={15,30,50,100,200,500};
weight1=0.2; *Will be changed - pi={0.2,0.4,0.5};
weight2=1-weight1;
delta=15; *Will be changed delta={15,30,50};
m1=40;
m2=m1+delta;
var1=16;
var2=var1;
    k=ranuni(J(n,1,0));
xx1 = loc(k<weight1);
xx2 = loc(k>=weight1);
n1=ncol(xx1);
n2=ncol(xx2);

sd1 = J(n1,1,0);
sd2 = J(n2,1,0);

x1=rannor(sd1)*sqrt(var1)+m1;
x2=rannor(sd2)*sqrt(var2)+m2;
x = (x1 || J(n1,1,1)) //  (x2 || J(n2,1,2));
finish GEN;

do kk= 1 to 100;
call GEN;
call sort(x,{1});
/*Calculating the number of rows and columns*/
```

```
n=nrow(x);
dim = 1;

f = (sample(x[,1],2,"NoReplace"))‘;
f=f‘;
/*Initial values for the means*/
mean1=min(f);
mean2=max(f);

/*Initial values for the variances*/
variance1 = var(x[,1]);
variance2 = variance1;

/*Initial values for the mixing coefficients*/
pi1=0.5;
pi2=1-pi1;

logfun = -10000;
diff = 100000;

do t =1 to 50 while(diff > 0.01);
s1=sqrt(variance1);
s2=sqrt(variance2);
/*E-step:*/
/*Calculating the normal densities*/
normd1 = pdf("Normal",x[,1],mean1,s1);
normd2 = pdf("Normal",x[,1],mean2,s2);

post = pi1*normd1 || pi2*normd2;
post = post / post[,+];

/*C-step:*/
class =  post[,<:>];
p1 = loc(class=1);
```

```
p2 = loc(class=2);

x_full = x|| post || class;
Xi1 = x_full[p1,];
Xi2 = x_full[p2,];

/*M-step:*/
/*Counting the number of rows of each partition*/
x1=nrow(Xi1);
x2=nrow(Xi2);

/*Update the mixing coefficients of each of the components*/
pi1 = x1/N;
pi2 = 1-pi1;

/*Update the means of each of the components*/
mean1 = (Xi1[,1]#Xi1[,3])[+] / sum(Xi1[,3]); /*mean1*/
mean2 = (Xi2[,1]#Xi2[,4])[+] / sum(Xi2[,4]); /*mean2*/

/*Update the variances of each of the components*/
variance1 = (Xi1[,3]#((Xi1[,1]-mean1)##2))[+] / sum(Xi1[,3]); /*variance1*/
variance2 = (Xi2[,4]#((Xi2[,1]-mean2)##2))[+] / sum(Xi2[,4]); /*variance2*/

newlogfun = logfun;
/*Calculating the log-likelihood function*/
logfun = sum(log(pi1*normd1+pi2*normd2));

/*Stops iterations once the difference between the latest log-likelihood*/
/*and the previous one is greater than 0.01*/
diff =  abs(logfun - newlogfun);

nm={"i" "Mean 1"  "Mean 2"  "Variance 1"  "Variance 2"  "pi1"  "logfun"};
CEM = CEM // ( t || mean1 || mean2 || variance1 || variance2 || pi1 || logfun);
end;
```

```
simulation = simulation // ( kk ||t||mean1||mean2||variance1||variance2||pi1);
free CEM;
end;
nm2={"Simulation" "Iterations" "Mean 1"  "Mean 2"  "Variance 1"  "Variance 2"  "pi1"};
print simulation[colname=nm2];

create simulation from simulation[colname=nm2] ;
append from simulation;
close simulation;

data sasuser.CEMSim;
set sasuser.CEMSim simulation;
run;
quit;
%mend;
%cem;
quit;

data sasuser.CEMgraphs;
set plot1 plot2 plot3 plot4 plot5 plot6 plot7 plot8 plot9;
run;

/*Graphing the Iterations*/
proc template;
define statgraph sgdesign;
dynamic _SAMPLE_SIZE1A _ITERATIONS1A _SAMPLE_SIZE2A _ITERATIONS2A _SAMPLE_SIZE3A
_ITERATIONS3A _SAMPLE_SIZE4A _ITERATIONS4A _SAMPLE_SIZE4A2 _ITERATIONS4A2 _SAMPLE_SIZE5A
_ITERATIONS5A _SAMPLE_SIZE6A _ITERATIONS6A _SAMPLE_SIZE7A _ITERATIONS7A _SAMPLE_SIZE8A
_ITERATIONS8A _SAMPLE_SIZE9A _ITERATIONS9A;
begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
layout overlay / xaxisopts=( reverse=false label=('Sample size') tickvalueattrs=(color=CX000000 )
linearopts=( minorticks=OFF)) yaxisopts=( label=('Iterations'));
seriesplot x=_SAMPLE_SIZE1A y=_ITERATIONS1A / name='series'
```

```
legendlabel='Delta = 15 (pi1 = 0.2)' connectorder=xaxis lineattrs=(color=CX0000FF
pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE2A y=_ITERATIONS2A / name='series2'
legendlabel='Delta = 15 (pi1 = 0.4)' connectorder=xaxis lineattrs=(color=CX0000FF
pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE3A y=_ITERATIONS3A / name='series3'
legendlabel='Delta = 15 (pi1 = 0.5)' connectorder=xaxis lineattrs=(color=CX0000FF
pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE4A y=_ITERATIONS4A / name='series4'
legendlabel='Delta = 30 (pi1 = 0.2)' connectorder=xaxis
lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE5A y=_ITERATIONS5A / name='series5'
legendlabel='Delta = 30 (pi1 = 0.4)' connectorder=xaxis
lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE6A y=_ITERATIONS6A / name='series6'
legendlabel='Delta = 30 (pi1 = 0.5)' connectorder=xaxis
lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE7A y=_ITERATIONS7A / name='series7'
legendlabel='Delta = 50 (pi1 = 0.2)' connectorder=xaxis
lineattrs=(color=CX00FF00 pattern=MEDIUMDASHSHORTDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE8A y=_ITERATIONS8A / name='series8'
legendlabel='Delta = 50 (pi1 = 0.4)' connectorder=xaxis
lineattrs=(color=CX00FF00 pattern=MEDIUMDASHSHORTDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE9A y=_ITERATIONS9A / name='series9'
legendlabel='Delta = 50 (pi1 = 0.5)' connectorder=xaxis
lineattrs=(color=CX00FF00 pattern=MEDIUMDASHSHORTDASH thickness=2 );
discretelegend 'series' 'series1' 'series2' 'series3' 'series4' 'series5' 'series6'
'series7' 'series8' 'series9' / opaque=false border=true valign=bottom displayclipped=true
down=1 order=rowmajor location=outside;
endlayout;
endlayout;
endgraph;
end;
run;
```

```
proc sgrender data=sasuser.CEMgraphs template=sgdesign;
dynamic _SAMPLE_SIZE1A="'SAMPLE_SIZE1'n" _ITERATIONS1A="ITERATIONS1"
_SAMPLE_SIZE2A="'SAMPLE_SIZE2'n" _ITERATIONS2A="ITERATIONS2" _SAMPLE_SIZE3A="'SAMPLE_SIZE3'n"
_ITERATIONS3A="ITERATIONS3" _SAMPLE_SIZE4A="'SAMPLE_SIZE4'n" _ITERATIONS4A="ITERATIONS4"
_SAMPLE_SIZE4A2="'SAMPLE_SIZE4'n" _ITERATIONS4A2="ITERATIONS4"
_SAMPLE_SIZE5A="'SAMPLE_SIZE5'n" _ITERATIONS5A="ITERATIONS5"
_SAMPLE_SIZE6A="'SAMPLE_SIZE6'n" _ITERATIONS6A="ITERATIONS6"
_SAMPLE_SIZE7A="'SAMPLE_SIZE7'n" _ITERATIONS7A="ITERATIONS7"
_SAMPLE_SIZE8A="'SAMPLE_SIZE8'n" _ITERATIONS8A="ITERATIONS8"
_SAMPLE_SIZE9A="'SAMPLE_SIZE9'n" _ITERATIONS9A="ITERATIONS9";
run;

/*Graphing the means*/
/*Mean 1*/
proc template;
define statgraph sgdesign;
dynamic _SAMPLE_SIZE1A _MEAN11A _SAMPLE_SIZE2A _MEAN12A _SAMPLE_SIZE3A _MEAN13A
_SAMPLE_SIZE4A _MEAN14A _SAMPLE_SIZE5A _MEAN15A _SAMPLE_SIZE6A _MEAN16A _SAMPLE_SIZE7A
_MEAN17A _SAMPLE_SIZE8A _MEAN18A _SAMPLE_SIZE9A _MEAN19A;
begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
layout overlay / xaxisopts=( label=('Sample size')) yaxisopts=( label=('Mean'));
seriesplot x=_SAMPLE_SIZE1A y=_MEAN11A / name='series'  legendlabel='Delta = 15 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE2A y=_MEAN12A / name='series2' legendlabel='Delta = 15 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE3A y=_MEAN13A / name='series3' legendlabel='Delta = 15 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE4A y=_MEAN14A / name='series4' legendlabel='Delta = 30 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE5A y=_MEAN15A / name='series5' legendlabel='Delta = 30 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE6A y=_MEAN16A / name='series6' legendlabel='Delta = 30 (pi1 = 0.5)'
```

```
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE7A y=_MEAN17A / name='series7' legendlabel='Delta = 50 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
seriesplot x=_SAMPLE_SIZE8A y=_MEAN18A / name='series8' legendlabel='Delta = 50 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
seriesplot x=_SAMPLE_SIZE9A y=_MEAN19A / name='series9' legendlabel='Delta = 50 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
discretelegend 'series' 'series1' 'series2' 'series3' 'series4' 'series5' 'series6' 'series7'
'series8' 'series9' / opaque=false border=true valign=bottom displayclipped=true
down=1 order=rowmajor location=outside;
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=sasuser.CEMgraphs template=sgdesign;
dynamic _SAMPLE_SIZE1A="'SAMPLE_SIZE1'n" _MEAN11A="MEAN11"
_SAMPLE_SIZE2A="'SAMPLE_SIZE2'n"
_MEAN12A="MEAN12" _SAMPLE_SIZE3A="'SAMPLE_SIZE3'n" _MEAN13A="MEAN13"
_SAMPLE_SIZE4A="'SAMPLE_SIZE4'n"
_MEAN14A="MEAN14" _SAMPLE_SIZE5A="'SAMPLE_SIZE5'n" _MEAN15A="MEAN15"
_SAMPLE_SIZE6A="'SAMPLE_SIZE6'n"
_MEAN16A="MEAN16" _SAMPLE_SIZE7A="'SAMPLE_SIZE7'n" _MEAN17A="MEAN17"
_SAMPLE_SIZE8A="'SAMPLE_SIZE8'n"
_MEAN18A="MEAN18" _SAMPLE_SIZE9A="'SAMPLE_SIZE9'n"
_MEAN19A="MEAN19";
run;

/*Mean 2*/
proc template;
define statgraph sgdesign; dynamic _SAMPLE_SIZE1A _MEAN21A _SAMPLE_SIZE2A _MEAN22A
_SAMPLE_SIZE3A _MEAN23A _SAMPLE_SIZE4A _MEAN24A _SAMPLE_SIZE5A _MEAN25A _SAMPLE_SIZE6A
_MEAN26A _SAMPLE_SIZE7A _MEAN27A _SAMPLE_SIZE8A _MEAN28A _SAMPLE_SIZE9A _MEAN29A;
```

```
begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
layout overlay / xaxisopts=( label=('Sample size')) yaxisopts=( label=('Mean'));
seriesplot x=_SAMPLE_SIZE1A y=_MEAN21A / name='series'  legendlabel='Delta = 15 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE2A y=_MEAN22A / name='series2' legendlabel='Delta = 15 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE3A y=_MEAN23A / name='series3' legendlabel='Delta = 15 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE4A y=_MEAN24A / name='series4' legendlabel='Delta = 30 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE5A y=_MEAN25A / name='series5' legendlabel='Delta = 30 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE6A y=_MEAN26A / name='series6' legendlabel='Delta = 30 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE7A y=_MEAN27A / name='series7' legendlabel='Delta = 50 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
seriesplot x=_SAMPLE_SIZE8A y=_MEAN28A / name='series8' legendlabel='Delta = 50 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
seriesplot x=_SAMPLE_SIZE9A y=_MEAN29A / name='series9' legendlabel='Delta = 50 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
discretelegend 'series' 'series1' 'series2' 'series3' 'series4' 'series5' 'series6' 'series7'
'series8' 'series9' / opaque=false border=true valign=bottom displayclipped=true
down=1 order=rowmajor location=outside;
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=sasuser.CEMgraphs template=sgdesign;
dynamic _SAMPLE_SIZE1A="'SAMPLE_SIZE1'n" _MEAN21A="MEAN21" _SAMPLE_SIZE2A="'SAMPLE_SIZE2'n"
_MEAN22A="MEAN22" _SAMPLE_SIZE3A="'SAMPLE_SIZE3'n"
_MEAN23A="MEAN23" _SAMPLE_SIZE4A="'SAMPLE_SIZE4'n"
```

```
_MEAN24A="MEAN24" _SAMPLE_SIZE5A="'SAMPLE_SIZE5'n"
_MEAN25A="MEAN25" _SAMPLE_SIZE6A="'SAMPLE_SIZE6'n"
_MEAN26A="MEAN26" _SAMPLE_SIZE7A="'SAMPLE_SIZE7'n"
_MEAN27A="MEAN27" _SAMPLE_SIZE8A="'SAMPLE_SIZE8'n"
_MEAN28A="MEAN28" _SAMPLE_SIZE9A="'SAMPLE_SIZE9'n"
_MEAN29A="MEAN29";
run;

/*Graphing the Variances*/
/*Variance 1*/
proc template;
define statgraph sgdesign;
dynamic _SAMPLE_SIZE1A _VARIANCE11A _SAMPLE_SIZE2A _VARIANCE12A _SAMPLE_SIZE3A _VARIANCE13A
_SAMPLE_SIZE4A _VARIANCE14A _SAMPLE_SIZE5A _VARIANCE15A _SAMPLE_SIZE6A _VARIANCE16A _SAMPLE_SIZE7A
_VARIANCE17A _SAMPLE_SIZE8A _VARIANCE18A _SAMPLE_SIZE9A _VARIANCE19A;
begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
layout overlay / xaxisopts=( label=('Sample size')) yaxisopts=( label=('Variance'));
seriesplot x=_SAMPLE_SIZE1A y=_VARIANCE11A / name='series'  legendlabel='Delta = 15 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE2A y=_VARIANCE12A / name='series2' legendlabel='Delta = 15 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE3A y=_VARIANCE13A / name='series3' legendlabel='Delta = 15 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE4A y=_VARIANCE14A / name='series4' legendlabel='Delta = 30 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE5A y=_VARIANCE15A / name='series5' legendlabel='Delta = 30 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE6A y=_VARIANCE16A / name='series6' legendlabel='Delta = 30 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE7A y=_VARIANCE17A / name='series7' legendlabel='Delta = 50 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
seriesplot x=_SAMPLE_SIZE8A y=_VARIANCE18A / name='series8' legendlabel='Delta = 50 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
```

```
seriesplot x=_SAMPLE_SIZE9A y=_VARIANCE19A / name='series9' legendlabel='Delta = 50 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
discretelegend 'series' 'series1' 'series2' 'series3' 'series4' 'series5' 'series6' 'series7'
'series8' 'series9' / opaque=false border=true valign=bottom displayclipped=true
down=1 order=rowmajor location=outside;
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=sasuser.CEMgraphs template=sgdesign;
dynamic _SAMPLE_SIZE1A="'SAMPLE_SIZE1'n" _VARIANCE11A="VARIANCE11" _SAMPLE_SIZE2A="'SAMPLE_SIZE2'n"
_VARIANCE12A="VARIANCE12" _SAMPLE_SIZE3A="'SAMPLE_SIZE3'n" _VARIANCE13A="VARIANCE13"
_SAMPLE_SIZE4A="'SAMPLE_SIZE4'n" _VARIANCE14A="VARIANCE14" _SAMPLE_SIZE5A="'SAMPLE_SIZE5'n"
_VARIANCE15A="VARIANCE15" _SAMPLE_SIZE6A="'SAMPLE_SIZE6'n" _VARIANCE16A="VARIANCE16"
_SAMPLE_SIZE7A="'SAMPLE_SIZE7'n" _VARIANCE17A="VARIANCE17" _SAMPLE_SIZE8A="'SAMPLE_SIZE8'n"
_VARIANCE18A="VARIANCE18" _SAMPLE_SIZE9A="'SAMPLE_SIZE9'n" _VARIANCE19A="VARIANCE19";
run;

/*Variance 2*/
proc template;
define statgraph sgdesign;
dynamic _SAMPLE_SIZE1A _VARIANCE21A _SAMPLE_SIZE2A _VARIANCE22A _SAMPLE_SIZE3A _VARIANCE23A
_SAMPLE_SIZE4A _VARIANCE24A _SAMPLE_SIZE5A _VARIANCE25A _SAMPLE_SIZE6A _VARIANCE26A
_SAMPLE_SIZE7A _VARIANCE27A _SAMPLE_SIZE8A _VARIANCE28A _SAMPLE_SIZE9A _VARIANCE29A;
begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
layout overlay / xaxisopts=( label=('Sample size')) yaxisopts=( label=('Variance'));
seriesplot x=_SAMPLE_SIZE1A y=_VARIANCE21A / name='series'  legendlabel='Delta = 15 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE2A y=_VARIANCE22A / name='series2' legendlabel='Delta = 15 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE3A y=_VARIANCE23A / name='series3' legendlabel='Delta = 15 (pi1 = 0.5)'
```

```
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE4A y=_VARIANCE24A / name='series4' legendlabel='Delta = 30 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE5A y=_VARIANCE25A / name='series5' legendlabel='Delta = 30 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE6A y=_VARIANCE26A / name='series6' legendlabel='Delta = 30 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE7A y=_VARIANCE27A / name='series7' legendlabel='Delta = 50 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
seriesplot x=_SAMPLE_SIZE8A y=_VARIANCE28A / name='series8' legendlabel='Delta = 50 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
seriesplot x=_SAMPLE_SIZE9A y=_VARIANCE29A / name='series9' legendlabel='Delta = 50 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
discretelegend 'series' 'series1' 'series2' 'series3' 'series4' 'series5' 'series6' 'series7'
'series8' 'series9' / opaque=false border=true valign=bottom displayclipped=true
down=1 order=rowmajor location=outside;
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=sasuser.CEMgraphs template=sgdesign;
dynamic _SAMPLE_SIZE1A="'SAMPLE_SIZE1'n" _VARIANCE21A="VARIANCE21" _SAMPLE_SIZE2A="'SAMPLE_SIZE2'n"
_VARIANCE22A="VARIANCE22" _SAMPLE_SIZE3A="'SAMPLE_SIZE3'n" _VARIANCE23A="VARIANCE23"
_SAMPLE_SIZE4A="'SAMPLE_SIZE4'n" _VARIANCE24A="VARIANCE24" _SAMPLE_SIZE5A="'SAMPLE_SIZE5'n"
_VARIANCE25A="VARIANCE25" _SAMPLE_SIZE6A="'SAMPLE_SIZE6'n" _VARIANCE26A="VARIANCE26"
_SAMPLE_SIZE7A="'SAMPLE_SIZE7'n" _VARIANCE27A="VARIANCE27" _SAMPLE_SIZE8A="'SAMPLE_SIZE8'n"
_VARIANCE28A="VARIANCE28" _SAMPLE_SIZE9A="'SAMPLE_SIZE9'n" _VARIANCE29A="VARIANCE29";
run;

/*Graphing the mixing coefficients*/
proc template; define statgraph sgdesign; dynamic _SAMPLE_SIZE1A _PI11A _SAMPLE_SIZE2A _PI12A
_SAMPLE_SIZE3A _PI13A _SAMPLE_SIZE4A _PI14A _SAMPLE_SIZE5A _PI15A _SAMPLE_SIZE6A _PI16A
```

```
_SAMPLE_SIZE7A _PI17A _SAMPLE_SIZE8A _PI18A _SAMPLE_SIZE9A _PI19A;
begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
layout overlay / xaxisopts=( label=('Sample size')) yaxisopts=( label=('Mixing Coefficient'));
seriesplot x=_SAMPLE_SIZE1A y=_PI11A / name='series'  legendlabel='Delta = 15 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE2A y=_PI12A / name='series2' legendlabel='Delta = 15 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE3A y=_PI13A / name='series3' legendlabel='Delta = 15 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CX0000FF pattern=MEDIUMDASH thickness=2 );
seriesplot x=_SAMPLE_SIZE4A y=_PI14A / name='series4' legendlabel='Delta = 30 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE5A y=_PI15A / name='series5' legendlabel='Delta = 30 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE6A y=_PI16A / name='series6' legendlabel='Delta = 30 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CXFF0000 thickness=2 );
seriesplot x=_SAMPLE_SIZE7A y=_PI17A / name='series7' legendlabel='Delta = 50 (pi1 = 0.2)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
seriesplot x=_SAMPLE_SIZE8A y=_PI18A / name='series8' legendlabel='Delta = 50 (pi1 = 0.4)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
seriesplot x=_SAMPLE_SIZE9A y=_PI19A / name='series9' legendlabel='Delta = 50 (pi1 = 0.5)'
connectorder=xaxis lineattrs=(color=CX00FF00 pattern=DASHDOTDOT thickness=2 );
discretelegend 'series' 'series1' 'series2' 'series3' 'series4' 'series5' 'series6' 'series7'
'series8' 'series9' / opaque=false border=true valign=bottom displayclipped=true
down=1 order=rowmajor location=outside;
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=sasuser.CEMgraphs template=sgdesign; dynamic _SAMPLE_SIZE1A="'SAMPLE_SIZE1'n"
_PI11A="PI11" _SAMPLE_SIZE2A="'SAMPLE_SIZE2'n" _PI12A="PI12" _SAMPLE_SIZE3A="'SAMPLE_SIZE3'n"
_PI13A="PI13" _SAMPLE_SIZE4A="'SAMPLE_SIZE4'n" _PI14A="PI14" _SAMPLE_SIZE5A="'SAMPLE_SIZE5'n"
```

```
_PI15A="PI15" _SAMPLE_SIZE6A="'SAMPLE_SIZE6'n" _PI16A="PI16" _SAMPLE_SIZE7A="'SAMPLE_SIZE7'n"
_PI17A="PI17" _SAMPLE_SIZE8A="'SAMPLE_SIZE8'n" _PI18A="PI18" _SAMPLE_SIZE9A="'SAMPLE_SIZE9'n"
_PI19A="PI19";
run;
```

Simulating gamma variates when facing misbehaving parameters

Jarod Smith 14016665

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor(s): Mr. J.T. Ferreira and Dr. M.T. Loots

Department of Statistics, University of Pretoria

30 October 2017

Abstract

Simulating from a gamma distribution with small shape parameter, particularly $\alpha < 1$, poses a number of challenges. When the shape parameter approaches zero, the gamma distribution loses its infinite-divisibility property. The central limit theorem - to find a standardised normal approximation for a variable having a gamma distribution with small shape parameter - is of little help. The key to this dilemma is a computationally efficient transformation of a gamma random variable. This study evaluates acceptance-rejection Monte Carlo methods that have been developed to overcome this problem and to compare the efficiency of these methods. An extension of the transformation to the bivariate case, under independence, is also considered.

Keywords and phrases: Acceptance rate; acceptance-rejection sampling; envelope function; logarithmic concave functions; R software; SAS software.

Declaration

I, *Jarod Mark Smith,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Jarod Mark Smith

_ _

Mr. J.T. Ferreira

_ _

Dr. M.T. Loots

_ _

Date

Acknowledgements

This work is based on the research supported in part by the "National Research Foundation" of South Africa for the grant, Unique Grant No. 94108. Any opinion, finding and conclusion or recommendation expressed in this material is that of the author(s) and the NRF does not accept any liability in this regard. The author(s) would also like to thank STATOMET for the financial assistance and the Department of Statistics at the University of Pretoria for making this essay possible.

Finally I would like to thank Mr. Johan Ferreira and Dr. Theodor Loots for their unwavering support and guidance throughout the year.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Let X be a continuous random variable which is gamma distributed with probability density function (pdf) given by (1)

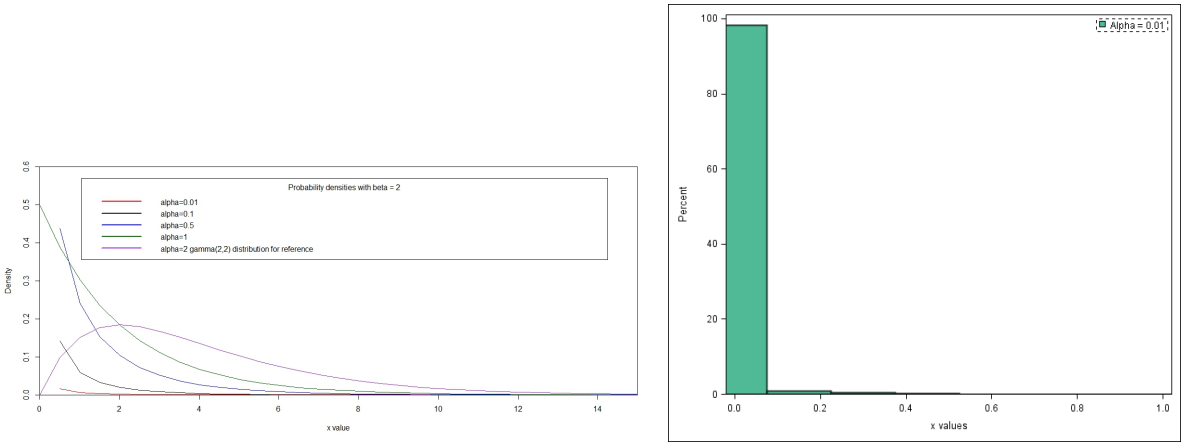
$$f(x;\alpha,\beta) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}} \; x > 0 \tag{1}$$

where $\alpha > 0$ and $\beta > 0$. The gamma distribution is a fundamental distribution in Statistics [9] and is closely related to many other distributions including the exponential, Erlang, chi-square, Nakagami-m, generalised gamma [21] and other transformations including inverse gamma, log-gamma and exponential-gamma to name a few. The focus here is to study the behavior of a simulated gamma random variable when α, the shape parameter, assumes small values. This is a significant problem because the small shape gamma distribution has applications in modelling lifetime random variables and also loss distributions in insurance science where accuracy is paramount for predictions and reserve calculations [11]. The gamma distribution with small shape parameter tends to be problematic to work with in a practical and simulation environment where it is found that general calculations become inaccurate.

Many numerical problems in Statistics can be solved through Monte Carlo methods. Monte Carlo simulation is a computer based mathematical procedure that produces distributions of possible outcomes, in essence it is a means of statistical evaluation of mathematical functions using a random sample. Monte Carlo simulation methods are useful for studying the characteristics of a population in compressed time. The acceptance-rejection method, which is a statistical simulation technique, is a form of Monte Carlo simulation. Without any doubt the power of Monte Carlo methods increases enormously when simulations are combined with analytical calculations and it is therefore useful investigating the underlying theoretical results [12]. Monte Carlo acceptance-rejection methods can be used to simulate from a gamma distribution with a small shape parameter. Limitations of simulating small shape gamma variates are evident in **R** and **SAS** when using basic functions such as **rgamma**, **rangam** and **randgen** (see Appendix and Figure 1) respectively. When α is very small, typically $\alpha \leq 0.3$, the pdf of the gamma distribution is concentrated around zero hence simulating practically non-zero values is quite difficult. Figure 2 compares the empirical pdf of (1), $\beta = 1$, to the theoretical as $\alpha \to 0$. For the empirical case it is observed that the pdf spikes and is highly concentrated around zero as $x \to 0$.

In addition when α is very small, the central limit theorem to simulate a suitably standardised normal approximation of a gamma random variable is not applicable. Simulating random gamma variates where the shape parameter tends to zero is well reported in literature, see [1, 4, 13, 17, 23], with quite a few prospective solutions including the acceptance-rejection method.

In Theorem 1 below it is demonstrated that for $X \sim gamma(\alpha, 1)$ the transformation $Z = -\alpha log X$



(a) (b)

Figure 1: Gamma pdf for decreasing α using **rgamma**(**R**) (a) and **rangam**(**SAS**) (b).

[15], converges in distribution to an $Exp(1)$ distribution which provides a compact envelope function. The transformation is used to developed an acceptance-rejection algorithm to simulate small shape gamma variates (returned on a log scale).

The transformation simplifies simulating small shape gamma variates significantly and is comparatively more efficient in terms of acceptance rates than other existing methods.

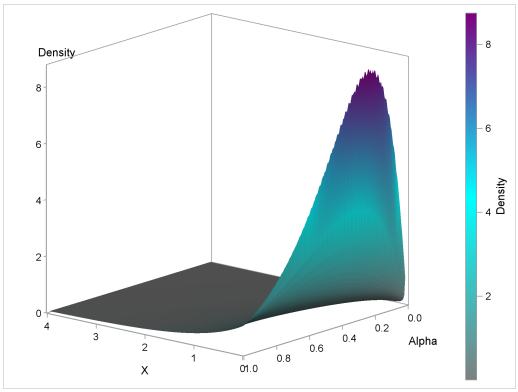
1.1 Aims and objectives

The aims and objectives of this study are described below:

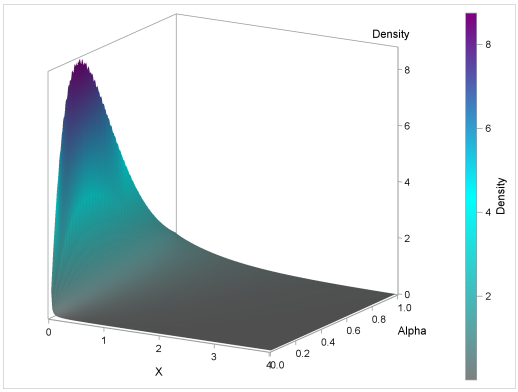
- Study the acceptance-rejection method in detail.
- Investigate the acceptance-rejection method proposed by [15] of simulating from a gamma distribution with small α values.
- Extend the methodology proposed by [15] to the bivariate case under independence.
- Compare the acceptance rates for the algorithm proposed by [15] to those of [1, 4, 13] in the univariate case.
- Compare the acceptance rates of the univariate algorithm proposed by [15] to the extended bivariate algorithm.

1.2 Outline of study

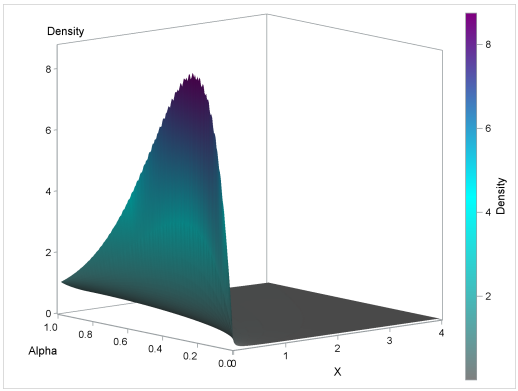
In section 2 the Monte Carlo acceptance-rejection method is introduced and an exploration into the development of the acceptance-rejection methods proposed by [1, 4, 13, 17, 23, 15] respectively is studied. In section 3 the limiting distribution result of the transformation $Z = -\alpha log(X)$ proposed by [15] is investigated. Further-more in section 4 an extension of the acceptance-rejection algorithm proposed by [15] to the bivariate case under independence is considered and finally in section 5, a comparison of the acceptance rates in the univariate and bivariate case is given.



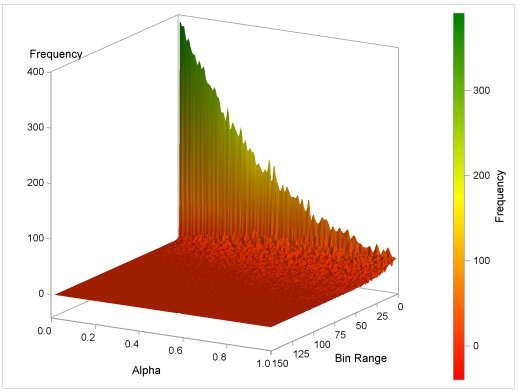
(a)



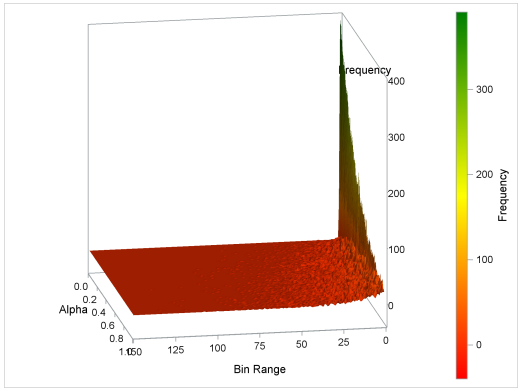
(b)



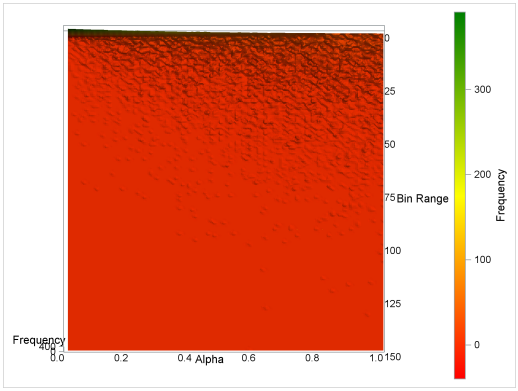
(c)



(d)



(e)



(f)

Figure 2: Theoretical (a to c, rotated plots) and empirical (d to f, rotated and empirical contour plot) gamma pdf as $\alpha \to 0$.

2 Literature review

Simulating random gamma variates where the shape parameter tends to zero is well reported in literature with quite a few prospective solutions including the acceptance-rejection method.

2.1 Acceptance-rejection method

Acceptance-rejection sampling is based on the idea that if it is difficult or impossible to sample observations, x values, at random directly from the pdf of the target distribution, say $f(x)$, then an appropriate alternative would be to generate values from an envelope function, $g(x)$, that satisfies the following:

1. Simpler to simulate from.
2. Completely bounds the target pdf as compactly as possible.
3. Reject values that are not acceptable.

The methodology can be summarised as follows:

1. Identify the target distribution $f(x)$.
2. Choose a simpler distribution, $g(x)$, that completely bounds $f(x)$ as compactly as possible such that $\frac{f(x)}{g(x)}$ is bounded for all values of x.
3. Ensure that $g(x)$ is a valid pdf. Let $h(x)$ be the valid envelope pdf.
4. Let $c = max\frac{f(x)}{h(x)}$ for all values of x where $c \geq 1$.
5. Generate x values from $h(x)$ and reject the values if they are not acceptable i.e. the value, $h(x)$, does not fall within $f(x)$.
6. The proportion of sampled variates accepted is $\frac{1}{c}$ i.e. the acceptance rate.
7. Note that $0 < \frac{f(x)}{ch(x)} \leq 1$ and this ratio is independent of $U \sim uniform(0, 1)$.
8. The number of iterations 'N' needed to successfully generate an X that comes from $f(x)$ has a geometric distribution with probability of success given by $p = P\left(U \leq \frac{f(x)}{ch(x)}\right)$.

The basic acceptance-rejection algorithm is given by Algorithm 1.

Algorithm 1 Basic acceptance-rejection algorithm.

1. Generate X from the envelope function $h(x)$.
2. Generate $U \sim uniform(0, 1)$ independently of X.
3. If $U \leq \frac{f(x)}{ch(x)}$ then return X otherwise, go to step 1.

In terms of efficiency c can be viewed as the number of variates that must be generated on average in order to end up with an acceptable value, where a smaller value of c is preferable i.e. closer to 1.

Proposition 1. *The efficiency of an acceptance-rejection algorithm is expressed in terms of the acceptance rate given by* $\frac{1}{c}$.

Proof. The probability of generating a point X from $h(x)$ that falls under $f(x)$ is given by $p = P\left(U \leq \frac{f(x)}{ch(x)}\right)$ where $U \sim uniform(0,\, 1)$.

Therefore,

$$p = P\left(U \leq \frac{f(X)}{ch(X)} | X = x\right).$$

Let A be the event such that $U \leq \frac{f(X)}{ch(X)}$ and B the event such that $X = x$ then it follows that

$$\begin{aligned} p &= P\left(A|B\right) \\ &= \frac{P\left(A \cap B\right)}{P\left(B\right)} \\ &= \int_{-\infty}^{\infty} \frac{f(x)}{ch(x)} h(x)dx \\ &= \frac{1}{c} \int_{-\infty}^{\infty} f(x)dx \\ &= \frac{1}{c} \end{aligned}$$

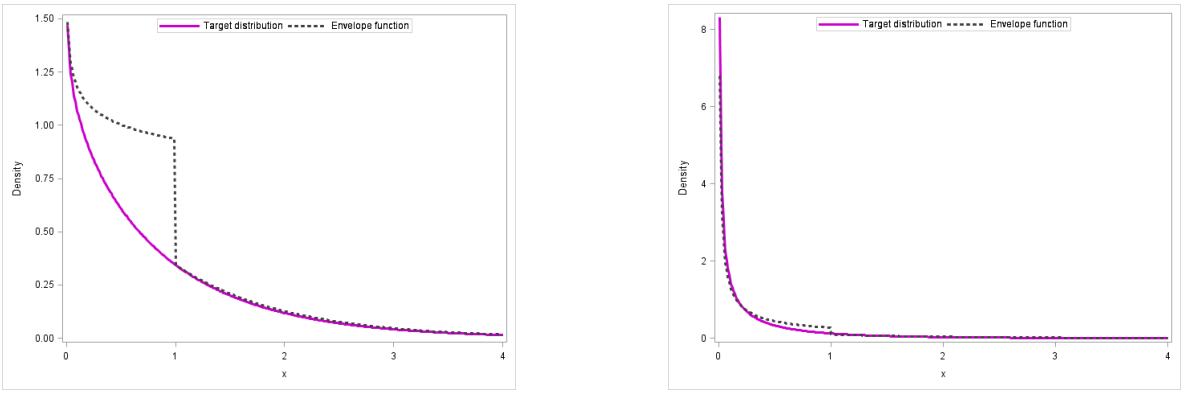
using the definition of conditional probability. □

2.2 The development of algorithm GS (gamma small shape)

[1] provides a renown acceptance-rejection algorithm, 'algorithm GS', that transforms uniformly distributed random numbers into gamma variates, specifically for simulating from a gamma distribution with shape parameter $0 < \alpha \leq 1$ and scale parameter $\beta = 1$. The gamma pdf, $f_{AD}(x; \alpha)$, is given by (1) when $\beta = 1$. The rejection method is based on the following envelope function

$$g_{AD}(x; \alpha) = \begin{cases} \frac{x^{\alpha-1}}{\Gamma(\alpha)} & 0 < x \leq 1 \\ \frac{e^{-x}}{\Gamma(\alpha)} & 1 \leq x. \end{cases} \tag{2}$$

Since



(a) (b)

Figure 3: $g_{AD}(x;\alpha)$ for $\alpha = 0.9$ (a) and $\alpha = 0.3$ (b).

$$\begin{cases} e^{-x} \leq 1 & if\ 0 < x \\ x^{\alpha-1} \leq 1 & if\ \alpha \leq 1\ and\ 1 \leq x. \end{cases}$$

See Figure 3 for a visual justification for the choice of $g_{AD}(x;\alpha)$.

However it has to be verified whether $g_{AD}(x;\alpha)$ is a valid pdf, thus from (2)

$$\begin{aligned} \int_0^\infty g_{AD}(x;\alpha)dx &= \int_0^1 \frac{x^{\alpha-1}}{\Gamma(\alpha)}dx + \int_1^\infty \frac{e^{-x}}{\Gamma(\alpha)}dx \\ &= \frac{1}{\Gamma(\alpha)\alpha} + \frac{1}{\Gamma(\alpha)e^1} \\ &= \frac{(\alpha + e^1)}{\Gamma(\alpha)\alpha e^1} \\ &= \frac{(\alpha + e^1)}{\Gamma(\alpha+1)e^1}\ since\ \Gamma(\alpha+1) = (\alpha)\Gamma(\alpha) \\ &\neq 1 \end{aligned}$$

where $e^1 = exp(1)$.

Therefore sampling takes place from the following valid pdf

$$h_{AD}(x;\alpha) = \frac{g(x;\alpha)}{\frac{(\alpha+e^1)}{\Gamma(\alpha+1)e^1}} = \begin{cases} x^{\alpha-1}\frac{e^1\alpha}{(e^1+\alpha)} & 0 < x \leq 1 \\ e^{-x}\frac{e^1\alpha}{(e^1+\alpha)} & 1 \leq x \end{cases} \tag{3}$$

which is proportional to $g_{AD}(x;\alpha)$. It follows that $f_{AD}(x;\alpha) \leq c_1 g_{AD}(x;\alpha) = h_{AD}(x;\alpha)$ and the proportion of sampled deviates accepted is $\frac{1}{c_1} = \frac{(\alpha+e^1)}{\Gamma(\alpha+1)e^1}$. The acceptance-rejection technique leads to Algorithm 2.

Algorithm 2 GS $(0 < \alpha \leq 1)$.

1. Generate $U \sim uniform(0,1)$. Set $b \leftarrow \left(e^1+\alpha\right)/e^1$ and $P \leftarrow bU$. If $P \geq 1$ go to 3.
2. Case $(x \leq 1)$. Set $X \leftarrow P^{1/\alpha}$. Generate $U^* \sim uniform(0,1)$. If $U^* > e^{-X}$ go back to 1. (U^* independent of U)
3. Case $(x > 1)$. Set $X \leftarrow -ln\left((b-P)/\alpha\right)$. Generate U^*. If $U^* > X^{\alpha-1}$ go back to 1. Otherwise deliver X.

2.3 A modification of algorithm GS

[4] proposes an algorithm, 'algorithm RGS (revised gamma small shape)' which is a modification of Algorithm 2 for generating random gamma variates for $0 < \alpha < 1$. The algorithm proves to be significantly faster (generates small shape gamma variates in shorter simulation time) and has lower rejection proportions compared to Algorithm 2, however it is slightly more complex. Specifically two modifications are made.

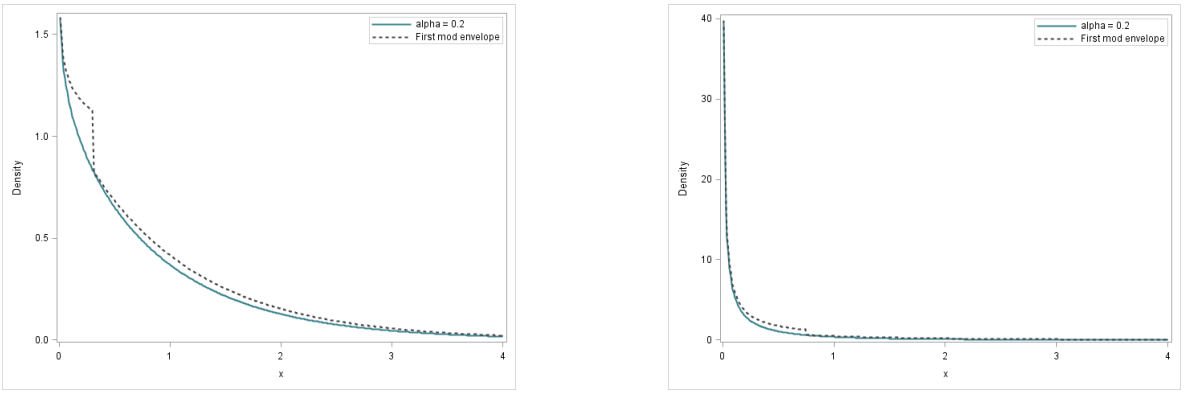
2.3.1 The first modification

The gamma pdf, $f_B(x;\alpha)$, is given by (1) when $\beta = 1$. Note that there is no difference between $f_B(x;\alpha)$ and $f_{AD}(x;\alpha)$, they are denoted differently throughout to avoid confusion. In terms of the envelope function, the restriction in (2) is changed from $x \in (0,1)$ and $1 \leq x$ to $x \in (0,z)$ and $z \leq x$ respectively, where z is a function of α i.e. $z = z(\alpha)$, z should be chosen in such a way that $\int_0^\infty g_B(x;\alpha)dx$ is a minimum. The motivation for this function is discussed in [2]. [4] proposes an approximation, $z \approx 0.07 + 0.75(1-\alpha)^{1/2}$, which leads to the following envelope function

$$g_B(x;\alpha) = \begin{cases} \frac{x^{\alpha-1}}{\Gamma(\alpha)} & 0 < x \leq z \\ \frac{z^{\alpha-1}e^{-x}}{\Gamma(\alpha)} & z \leq x. \end{cases} \tag{4}$$

See Figure 4 for a visual justification for the choice of $g_B(x;\alpha)$.

However it has to be verified whether $g_B(x;\alpha)$ is a valid pdf, thus



(a) (b)

Figure 4: $g_B(x;\alpha)$ for $\alpha = 0.9$ (a) and $\alpha = 0.2$ (b).

$$\begin{aligned}\int_0^\infty g_B(x;\alpha)dx &= \int_0^z \frac{x^{\alpha-1}}{\Gamma(\alpha)}dx + \int_z^\infty \frac{z^{\alpha-1}e^{-x}}{\Gamma(\alpha)}dx \\ &= \frac{z^\alpha}{\Gamma(\alpha)\alpha} + \frac{z^{\alpha-1}}{\Gamma(\alpha)e^z} \\ &= \frac{z^\alpha(e^z + \frac{\alpha}{z})}{\Gamma(\alpha+1)e^z} \\ &\neq 1.\end{aligned}$$

Therefore sampling gamma variates takes place from the pdf

$$h_B(x;\alpha) = \frac{g_B(x;\alpha)}{\frac{z^\alpha(e^z+\frac{\alpha}{z})}{\Gamma(\alpha+1)e^z}} = \begin{cases} \left(\frac{x}{z}\right)^{\alpha-1}\frac{\alpha}{(bz)} & 0 < x \leq z \\ e^{-x}\frac{\alpha}{(bz)} & z \leq x \end{cases} \tag{5}$$

which is proportional to $g_B(x;\alpha)$ and where $b = 1 + e^{-z}\frac{\alpha}{z}$. It follows that $f_B(x;\alpha) \leq c_2 g_B(x;\alpha) = h_B(x;\alpha)$ and the proportion of sampled deviates accepted is $\frac{1}{c_2} = \frac{z^\alpha(e^z+\frac{\alpha}{z})}{\Gamma(\alpha+1)e^z}$.

2.3.2 The second modification

The second modification aims to improve the speed of Algorithm 2 by avoiding the exponentiation in steps 2 and 3, in doing so making use of the following results given by [4]

1.
$$e^{-x} \geq \frac{(2-x)}{(2+x)} \;\; if \;\; x \geq 0.$$

2.

$$(1+x)^{-c} \geq (1+mx)^{-1} \quad if \;\; x \geq 0,\; 1 \geq m \geq 0.$$

The final algorithm is given by Algorithm 3.

Algorithm 3 RGS $(0 < \alpha < 1)$.

1. Initialise $z \leftarrow .07 + .75(1+\alpha)^{1/2}$, $b \leftarrow 1 + e^{-z}\alpha/z$.
2. Generate $U \sim uniform(0,1)$ and set $P \leftarrow bU$. If $P > 1$ go to step 4.
3. Set $X \leftarrow zP^{1/\alpha}$. Generate $U^* \sim uniform(0,1)$. If $U^* \leq (2-X)/(2+X)$, deliver X.
4. If $U^* > e^{-X}$ go to step 1, otherwise deliver X. (U^* independent of U).
5. Set $X \leftarrow -ln\left(z\left(b-P\right)/\alpha\right)$, $Y \leftarrow X/z$. Generate U^*. If $U^*\left(\alpha + Y - \alpha Y\right) < 1$, deliver X.
6. If $U^* > Y^{\alpha-1}$ go to step 1, otherwise deliver X.

where step 1 is performed once if α remains constant whilst generating the required sample.

2.4 Simulating gamma variates using normal random variates

[17] proposes a method for generating gamma variates by taking the cube of a suitably scaled normal random variate assuming there is a fast and efficient way of generating normal variates. The main focus is generating gamma variates for $\alpha \geq 1$. The procedure can be improved in terms of speed by implementing the squeeze method proposed by [16]. To generate gamma variates, γ_α, for $\alpha < 1$ [17] proposes using $\gamma_\alpha = \gamma_{1+\alpha}U^{1/\alpha}$ with U from $uniform(0,1)$ instead.

2.5 Simulating gamma variates using a generalised exponential distribution

[13] suggests a convenient acceptance-rejection method for simulating gamma variates, $0 < \alpha < 1$, using a generalised exponential distribution.

2.5.1 The first algorithm

The gamma pdf, $f_{KG}(x;\alpha)$, is given by (1) when $\beta = 1$. The following generalised exponential distribution is initially used as an envelope function

$$g_{KG}(x;\alpha;\frac{1}{2}) = \left\{\frac{\alpha}{2}\left(1-e^{-x/2}\right)^{\alpha-1}e^{-x/2}. \right. \tag{6}$$

However it has to be verified whether $g_{KG}(x;\alpha;\frac{1}{2})$ is a proper pdf, thus

$$\begin{aligned}
\int_0^\infty g_{KG}(x;\alpha;\frac{1}{2})dx &= \int_0^\infty \frac{\alpha}{2}\left(1-e^{-x/2}\right)^{\alpha-1} e^{-x/2}dx \\
&= \int_0^1 \alpha u^{\alpha-1}du \quad \left(u = 1-e^{-x/2} \Rightarrow du = \frac{e^{-x/2}dx}{2}\right) \\
&= \alpha\frac{u^\alpha}{\alpha}\Big|_0^1 \\
&= u^\alpha\Big|_0^1 \\
&= \left(1-e^{-x/2}\right)^\alpha\Big|_0^\infty \\
&= 1.
\end{aligned}$$

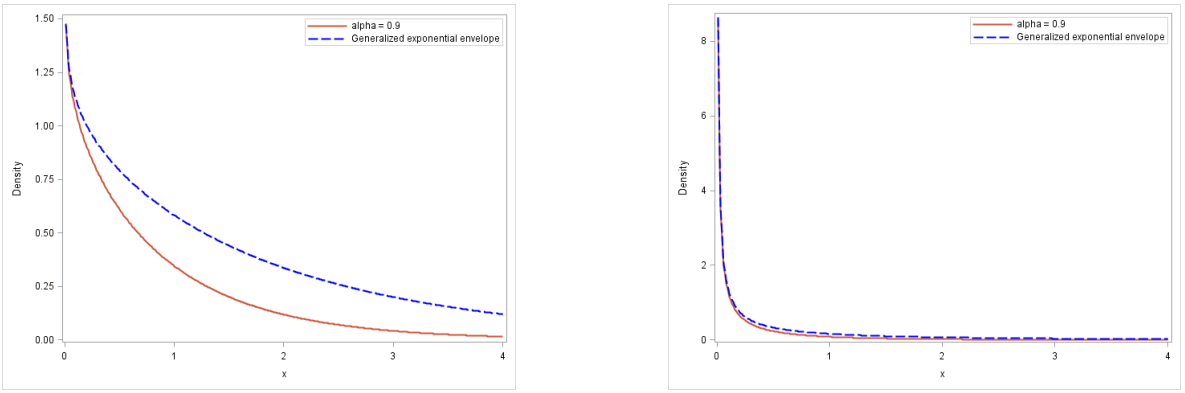
This is as expected since $g_{KG}(x;\alpha;\frac{1}{2})$ is already a valid pdf. In order to find an appropriate envelope function a suitable c_3 is needed such that $f_{KG}(x;\alpha) \leq c_3 g_{KG}(x;\alpha;\frac{1}{2})$ thus

$$\begin{aligned}
\frac{f_{KG}(x;\alpha)}{g_{KG}(x;\alpha;\lambda)} &= \frac{\frac{x^{\alpha-1}e^{-x}}{\Gamma(\alpha)}}{\frac{\alpha\left(1-e^{-x/2}\right)^{\alpha-1}e^{-x/2}}{2}} \\
&= \frac{2x^{\alpha-1}e^{-x}}{\alpha\Gamma(\alpha)\left(1-e^{-x/2}\right)^{\alpha-1}e^{-x/2}} \\
&= \frac{2x^{\alpha-1}e^{-x/2}}{\Gamma(\alpha+1)\left(1-e^{-x/2}\right)^{\alpha-1}} \\
&\leq \frac{2^\alpha}{\Gamma(\alpha+1)} \\
&= c_3.
\end{aligned}$$

It follows that

$$f_{KG}(x;\alpha) \leq h_{KG}(x;\alpha;\frac{1}{2}) = c_3 g_{KG}(x;\alpha;\frac{1}{2}) = \begin{cases} \frac{\alpha 2^\alpha}{2\Gamma(\alpha+1)}\left(1-e^{-x/2}\right)^{\alpha-1}e^{-x/2} & x>0. \end{cases} \tag{7}$$

See Figure 5 for a visual justification for the modification of $g_{KG}(x;\alpha;\frac{1}{2})$.



(a) (b)

Figure 5: Modified $g_{KG}(x;\alpha;\frac{1}{2})$ for $\alpha = 0.9$ (a) and $\alpha = 0.2$ (b).

The first algorithm (Algorithm 4) makes use of (7).

Algorithm 4 Kundu-Gupta-1.

1. Generate $U \sim uniform(0,1)$.
2. Compute $X = -2ln\left(1 - U^{1/\alpha}\right)$.
3. Generate $V \sim uniform(0,1)$ independent of U.
4. If $V \leq \frac{X^{\alpha-1}e^{-X/2}}{2^{\alpha-1}\left(1-e^{-X/2}\right)^{\alpha-1}}$ accept X, otherwise go to step 1.

2.5.2 The second algorithm

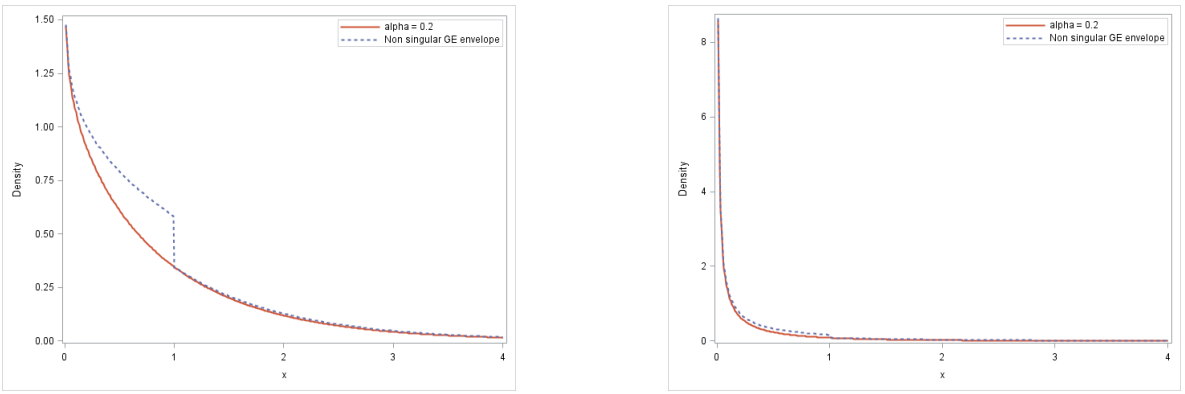
It is noted that even though (7) is true for all values of x, the upper bound provided when $0 < x < 1$ may result in a singularity, thus a new envelope function $t_{1KG}(x;\alpha)$ is given to circumvent this where

$$t_{1KD}(x;\alpha) = \begin{cases} \frac{2^{\alpha}}{\Gamma(\alpha+1)} g_{KG}(x;\alpha;\frac{1}{2}) & 0 < x < 1 \\ \frac{e^{-x}}{\Gamma(\alpha)} & x > 1. \end{cases} \tag{8}$$

It has to be verified whether $t_{1KG}(x;\alpha)$ is a proper pdf,

$$
\begin{aligned}
\int_0^\infty t_{1KD}(x;\alpha)dx &= \frac{2^\alpha}{\Gamma(\alpha+1)}\int_0^1 g_{KG}(x;\alpha;\frac{1}{2})dx + \int_1^\infty \frac{e^{-x}}{\Gamma(\alpha)}dx \\
&= \frac{2^\alpha}{\Gamma(\alpha+1)}\int_0^1 \frac{\alpha}{2}\left(1-e^{-x/2}\right)^{\alpha-1}e^{-x/2}dx + \int_1^\infty \frac{e^{-x}}{\Gamma(\alpha)}dx \\
&= \frac{2^\alpha}{\Gamma(\alpha+1)}\int_0^{\left(1-e^{-x/2}\right)} \alpha u^{\alpha-1}du + \frac{1}{\Gamma(\alpha)e^1} \quad \left(u = 1-e^{-x/2} \Rightarrow du = \frac{e^{-x/2}dx}{2}\right) \\
&= \frac{2^\alpha}{\Gamma(\alpha+1)}u^\alpha \Big|_{u=-1}^{u=\left(1-e^{-x/2}\right)} + \frac{1}{\Gamma(\alpha)e^1} \\
&= \frac{2^\alpha}{\Gamma(\alpha+1)}\left(1-e^{-x/2}\right)^\alpha \Big|_{x=0}^{x=1} + \frac{1}{\Gamma(\alpha)e^1} \\
&= \frac{1}{\alpha\Gamma(\alpha)}\left(2^\alpha\left[1-e^{-1/2}\right]^\alpha + \alpha e^{-1}\right) \\
&= c_4 \\
&\neq 1.
\end{aligned}
$$

See Figure 6 for a visual justification for the modification of $t_{1KG}(x;\alpha)$.



(a) (b)

Figure 6: Modified $g_{KG}(x;\alpha;\frac{1}{2})$ for various $\alpha = 0.9$ (a) and $\alpha = 0.2$ (b).

The normalised envelope pdf is given by

$$
t^*_{1KD}(x;\alpha) = \frac{1}{c_4}t_{1KD}(x;\alpha) = \begin{cases} \frac{2^\alpha}{c_4\Gamma(\alpha+1)}g_{KG}(x;\alpha;\frac{1}{2}) & 0 < x < 1 \\ \frac{e^{-x}}{c_4\Gamma(\alpha)} & x > 1. \end{cases} \tag{9}
$$

Algorithm 5 makes use of (9)

Algorithm 5 Kundu-Gupta-2.

Set $a = \frac{\left(1-e^{-1/2}\right)^{\alpha}}{(1-e^{-1/2})^{\alpha}+\frac{\alpha e^{-1}}{2^{\alpha}}}$ and $b = \left(1-e^{-1/2}\right)^{\alpha} + \frac{\alpha e^{-1}}{2^{\alpha}}$.

1. Generate $U \sim uniform(0,1)$.
2. If $U \leq a$, then $X = -2ln\left[1-(Ub)^{1/\alpha}\right]$, otherwise $X = -ln\left[\frac{2^{\alpha}}{\alpha}b\,(1-U)\right]$.
3. Generate $V \sim uniform(0,1)$ independent of U. If $X \leq 1$, check whether $V \leq \frac{X^{\alpha-1}e^{-X/2}}{2^{\alpha-1}\left(1-e^{-X/2}\right)^{\alpha-1}}$. If true return X, otherwise go back to step 1. If $X > 1$, check whether $V \leq X^{\alpha-1}$. If true return X, otherwise go back to step 1.

2.5.3 The third algorithm

Observing that $t_{1KD}(x;\alpha)$ is a piece wise pdf and using the same method as in [4] by altering the restriction in (8) so that the bounds depend on α i.e. $d = d(\alpha)$. [13] suggests that $d(\alpha) = 1.0334 - 0.0766e^{2.2942\alpha}$. A modified envelope function is given by

$$t_{2KD}(x;\alpha) = \begin{cases} \frac{2^{\alpha}}{\Gamma(\alpha+1)} g_{KG}(x;\alpha;\frac{1}{2}) & 0 < x < d(\alpha) \\ \frac{e^{-x}}{\Gamma(\alpha)} & x > d(\alpha). \end{cases} \tag{10}$$

However it has to be verified whether $t_{2KG}(x;\alpha)$ is a valid pdf, thus

$$\begin{aligned}
\int_0^{\infty} t_{2KD}(x;\alpha)dx &= \frac{2^{\alpha}}{\Gamma(\alpha+1)} \int_0^{d(\alpha)} g_{KG}(x;\alpha;\frac{1}{2})dx + \int_{d(\alpha)}^{\infty} \frac{e^{-x}}{\Gamma(\alpha)}dx \\
&= \frac{2^{\alpha}}{\Gamma(\alpha+1)} \int_0^{d(\alpha)} \frac{\alpha}{2}\left(1-e^{-x/2}\right)^{\alpha-1} e^{-x/2}dx + \int_{d(\alpha)}^{\infty} \frac{e^{-x}}{\Gamma(\alpha)}dx \\
&= \frac{2^{\alpha}}{\Gamma(\alpha+1)} \int_0^{\left(1-e^{-d(\alpha)/2}\right)} \alpha u^{\alpha-1}du + \frac{1}{\Gamma(\alpha)e^{d(\alpha)}} \quad \left(u = 1-e^{-x/2} \Rightarrow du = \frac{e^{-x/2}dx}{2}\right) \\
&= \frac{2^{\alpha}}{\Gamma(\alpha+1)} u^{\alpha} \Big|_{u=-1}^{u=\left(1-e^{-d(\alpha)/2}\right)} + \frac{1}{\Gamma(\alpha)e^{d(\alpha)}} \\
&= \frac{2^{\alpha}}{\Gamma(\alpha+1)} \left(1-e^{-x/2}\right)^{\alpha} \Big|_{x=0}^{x=d(\alpha)} + \frac{1}{\Gamma(\alpha)e^{d(\alpha)}} \\
&= \frac{1}{\alpha\Gamma(\alpha)} \left(2^{\alpha}\left[1-e^{-d(\alpha)/2}\right]^{\alpha} + \alpha e^{-d(\alpha)}\right) \\
&= c_5 \\
&\neq 1.
\end{aligned}$$

The normalised envelope pdf is given by

$$t^*_{2KD}(x;\alpha) = \frac{1}{c_5} t_{2KD}(x;\alpha) = \begin{cases} \frac{2^\alpha}{c_5\Gamma(\alpha+1)} g_{KG}(x;\alpha;\frac{1}{2}) & 0 < x < d(\alpha) \\ \frac{e^{-x}}{c_5\Gamma(\alpha)}. & x > d(\alpha). \end{cases} \tag{11}$$

Using (11) the final algorithm is given by Algorithm 6.

Algorithm 6 Kundu-Gupta-3.

Set $d = 1.0334 - 0.0766e^{2.2942\alpha}$, $a = 2^\alpha \left(1 - e^{-d/2}\right)^\alpha$, $b = \alpha d^{\alpha-1} e^{-d}$ and $c = a + b$.

1. Generate $U \sim uniform(0,1)$.
2. If $U \leq \frac{a}{a+b}$, then $X = -2ln\left[1 - \frac{(cU)^{1/\alpha}}{2}\right]$, otherwise $X = -ln\left[\frac{c(1-U)}{\alpha d^{\alpha-1}}\right]$.
3. Generate $V \sim uniform(0,1)$ independent of U. If $X \leq d$, check whether $V \leq \frac{X^{\alpha-1}e^{-X/2}}{2^{\alpha-1}\left(1-e^{-X/2}\right)^{\alpha-1}}$. If true return X, otherwise go back to step 1. If $X > d$, check whether $V \leq \left(\frac{d}{X}\right)^{1-\alpha}$. If true return X, otherwise go back to step 1.

Note that the normalising constant $c_5 \geq 1$ and in general the value of $'c'$ should be as small as possible to ensure that the proportion of values accepted is as large as possible. The method proposed by [13] has greater acceptance rates compared to [1] and [4]. The only difficulty being that the approximation is slightly inaccurate for $0.9 < \alpha < 1$ (see Section 5).

2.6 Simulating gamma random variables using the ratio-of-uniforms technique

[23] suggests a simple gamma random number generator with no restriction on the shape parameter that can be used to generate gamma variates. Acceptance regions for the proposed algorithm are determined by applying the ratio-of-uniforms methods (see Appendix). Various comparisons in terms of computational time are made with other proposed generators such as [4, 6, 5, 19, 17].

2.7 Simulating gamma random variables using logarithmic transformations

A non-standard acceptance-rejection algorithm that provides results more efficient than existing methods is given by [15]. It is demonstrated that for $X \sim gamma(\alpha, 1)$ where $f_{Liu}(x)$ is given by (1) when $\beta = 1$, the transformation

$$Z = -\alpha log X \tag{12}$$

converges in distribution to an $Exp(1)$ distribution. The transformation simplifies simulating small shape gamma variates significantly. By using methods suggested in [14] the envelope function is chosen such that it is as tight an upper bound for the transformed target pdf, $f_{Liu}(z)$, as possible. The target pdf is given by

$$f_{Liu}(z) = \left\{ \frac{1}{\Gamma(\alpha+1)} e^{-z-e^{-z/\alpha}} \quad \infty < z < \infty. \right. \tag{13}$$

The un-normalised envelope function is given by (see Section 3 for the derivation)

$$h_{Liu_1}(z) = \begin{cases} e^{-z} & z \geq 0 \\ w\lambda e^{\lambda z} & z < 0 \end{cases} \tag{14}$$

where $\lambda = \lambda(\alpha) = \alpha^{-1} - 1$ and $w = w(\alpha) = \alpha/e^1(1-\alpha)$ and the acceptance rate for the suggested method is $r = r(\alpha) = [1 + w(\alpha)]^{-1}$. The normalised envelope function is a mixture pdf given by

$$h_{Liu_2}(z) = \left\{ \frac{1}{1+w} e^{-z} - \frac{w}{1+w} e^{\lambda z} \right. . \tag{15}$$

The proposed algorithm (Algorithm 7) is given by,

Algorithm 7 Ryan Martin.

1. Set $\lambda = \lambda(\alpha)$, $w = w(\alpha)$ and $r = r(\alpha)$.
2. Generate $U \sim uniform(0, 1)$.
3. If $U \leq r$ then $z = -log(U/r)$, otherwise $z = -\frac{log(U)}{\lambda}$.
4. If $\frac{f_{Liu}(z)}{h_{Liu_1}(z)} > U$ then return$Z = z$.
5. Return $Y = e^{-Z/\alpha}$.

The simulated variates (Y′s) are returned on log scale for convenience.

3 Univariate gamma simulation with small shape parameter

3.1 Application of acceptance-rejection sampling

It is important to keep in mind that when developing and using an acceptance-rejection algorithm the choice of envelope function becomes crucial as it is directly related to the acceptance rate.

In [15] the overall target pdf is given by (1) when $\beta = 1$. For small values of α, the transformation $Z = -\alpha logX$ is made and the transformed target pdf is given by (13), the un-normalised envelope function

is given by (14) and the corresponding normalised envelope function by (15). The limiting distribution of the transformation, i.e. the limiting distribution of (13), is an $Exp(1)$ distribution. The reason for the transformation becomes clear due to the fact that generating exponential variates is computationally quick as a result of the simple structure of the exponential distribution's cumulative density function. The envelope function for the above transformation is determined by methods given in [14], by using exponential curves as envelope functions. Since the target pdf is log-concave (this can be verified by showing $\frac{d^2}{dz^2} ln f_{Liu}(z) \leq 0 \; \forall \; z \; \in \; (-\infty, \infty)$, see Appendix) it is ideal for acceptance-rejection sampling using a piece wise exponential distribution as an envelope function. Additionally since the mode, m, of (13) occurs on the interior of the support of the pdf it will be highly advantageous using two exponential functions orientated in opposite directions from the mode as the envelope function [14]. The proposed envelope function is chosen to be a compact upper bound by, geometrically, selecting optimal tangent points to (13). The un-normalised envelope function has the form

$$h_{Lange_1}(x) = \begin{cases} c_{left}\lambda_{left} e^{-\lambda_{left}(m-x)} & x < m \quad 'left' \; of \; the \; mode \\ c_{right}\lambda_{right} e^{-\lambda_{right}(x-m)} & x \geq m \quad 'right' \; of \; the \; mode \end{cases} \tag{16}$$

and the normalised envelope pdf

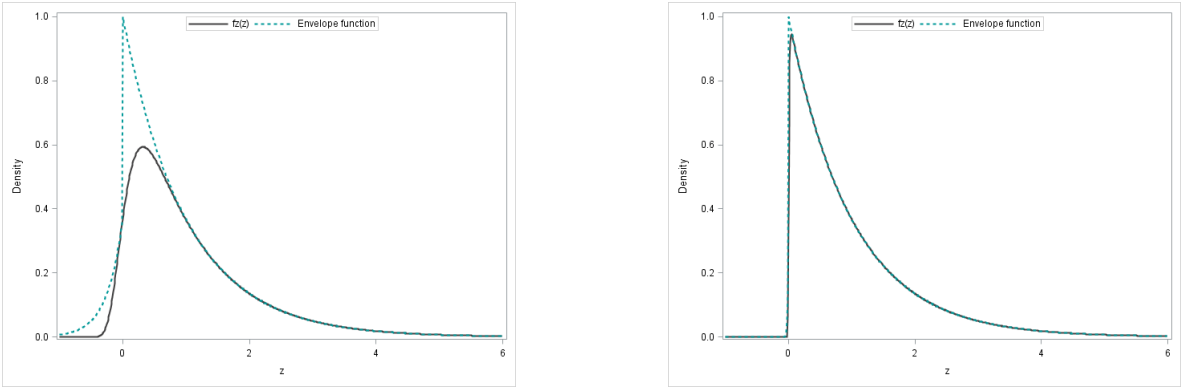
$$h_{Lange_2}(x) = \begin{cases} \frac{c_{left}}{c_{left}+c_{right}}\lambda_{left} e^{-\lambda_{left}(m-x)} & x < m \\ \frac{c_{right}}{c_{left}+c_{right}}\lambda_{right} e^{-\lambda_{right}(x-m)} & x \geq m. \end{cases} \tag{17}$$

The expressions given by [15] are derived as follows:

1. The target pdf is given by (13).
2. Solve for $\lambda = -f'_{Liu}(z)/f_{Liu}(z)$.
3. $\frac{d^2}{dz^2} ln\left[f_{Liu}(z)\right] = \frac{d^2}{dz^2} ln\left(\frac{1}{c_4} e^{-z-e^{-z/\alpha}}\right) = -\frac{e^{-\frac{z}{\alpha}}}{\alpha^2} \leq 0 \; \forall \; z \; \epsilon \quad (-\infty, \infty) \; \therefore$ log-concave.
4. Since $Z \rightarrow Exp(1)$ in distribution we have the following for $x \geq m$ where $m = 0$:
   - $h_{Lange_1}(x) = e^{-x}$.
   - $\therefore \; \lambda_{right} = 1$.
   - $\therefore \; c_{right} = 1$.
5. For $x < m$:
   - $\lambda_{left} : \frac{f'_{Liu}(z)}{f_{Liu}(z)} = \frac{\left(e^{-z-e^{-z/\alpha}}\right)\left(\frac{1}{\alpha}e^{-z/\alpha}-1\right)}{\left(e^{-z-e^{-z/\alpha}}\right)} \; |_{Z=0 \, (optimal \; tangent \; point)} = \frac{1}{\alpha} - 1.$
   - $c_{leftt} = w = \frac{\left(e^{-z-e^{-z/\alpha}}\right)}{\left(\frac{1}{\alpha}e^{-z/\alpha}-1\right)} e^{-\left(\frac{1}{\alpha}e^{-z/\alpha}-1\right)Z} \; |_{Z=0 \, (optimal \; tangent \; point)} = \frac{\alpha}{e^1(1-\alpha)}.$

6. $r = \frac{1}{c_{left}+c_{right}} = \frac{1}{1+w} = \left(1 + \frac{\alpha}{e^1(1-\alpha)}\right)^{-1}$ *(the acceptance rate)*.

Figure 7 shows plots of the envelope function $h_{Liu_1}(z)$ and the target pdf $f_{Liu}(z)$ for various values of α.



(a) (b)

Figure 7: $f_{Liu}(z)$ and $h_{Liu_1}(z)$ for $\alpha = 0.2$ (a) and $\alpha = 0.01$ (b).

3.2 Limit distribution result with transformation $Z = -\alpha logX$

For small α values the limiting distribution of the transformation $Z = -\alpha logX$ is an $Exp(1)$ distribution with pdf given by

$$f_{exp}(x) = \begin{cases} e^{-x} & x > 0 \\ 0 & otherwise. \end{cases}$$

Theorem 2. *For $Y \sim Gamma(\alpha, 1)$, the transformation: $Z = -\alpha logY$ converges in distribution to $Exp(1)$ as $\alpha \to 0$.*

Proof. Using Theorem 3.3.1 in [3],

$$\Gamma(\alpha + 1) = \alpha\Gamma(\alpha)$$
$$\therefore \Gamma(\alpha) = \frac{\Gamma(\alpha + 1)}{\alpha}.$$

Secondly, from (1)

$$
\begin{aligned}
E\left(Y^k\right) &= \int_0^\infty y^k \frac{y^{(\alpha-1)}e^{-y}}{\Gamma(\alpha)}dy \\
&= \frac{1}{\Gamma(\alpha)}\int_0^\infty y^{(k+\alpha-1)}e^{-y}dy \\
&= \frac{1}{\Gamma(\alpha)}\int_0^\infty \frac{\Gamma(\alpha+1)}{\Gamma(\alpha+1)}y^{(k+\alpha-1)}e^{-y}dy \\
&= \frac{\Gamma(\alpha+1)}{\Gamma(\alpha)}\int_0^\infty \frac{y^{(k+\alpha-1)}e^{-y}}{\Gamma(\alpha+1)}dy \\
&= \frac{\Gamma(\alpha+1)}{\Gamma(\alpha)}.
\end{aligned}
$$

Consider the following transformation

$$
Z = -\alpha log Y.
$$

The characteristic function of Z is given by

$$
\begin{aligned}
\Phi_z(t) &= E\left(e^{itz}\right) \\
&= E\left(Y^{-i\alpha t}\right) \\
&= \frac{\Gamma(\alpha - i\alpha t)}{\Gamma(\alpha)}.
\end{aligned}
$$

This expression can be rewritten as

$$
\begin{aligned}
\Phi_z(t) &= \frac{\Gamma(1+\alpha-i\alpha t)/(\alpha - i\alpha t)}{\Gamma(1+\alpha)/\alpha} \\
&= \frac{1}{1-it}\frac{\Gamma(1+0\,(\alpha))}{\Gamma(1+0\,(\alpha))}
\end{aligned}
$$

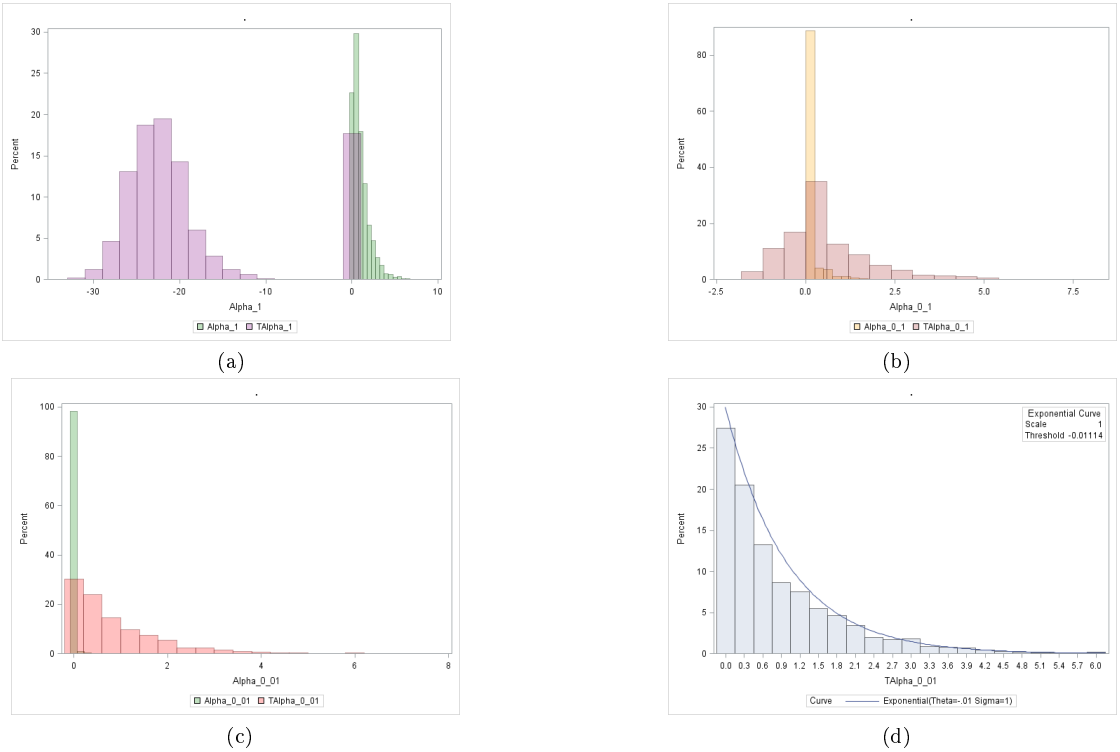
where $0\,(\alpha)$ become insignificant as $\alpha \to 0$ (see Appendix). The gamma function is continuous on $y > 0$, therefore the limit of $\Phi_z(t))$ as $\alpha \to 0$ exists and is given by $\frac{1}{1-it}$ which is the characteristic function of the $Exp(1)$ distribution.

□

Since Z is a one-to-one transformation we can obtain the pdf of Z by using techniques given in Chapter 6 of [3],

$$
\begin{aligned}
f_{Liu}(z) = f_z(z) &= f_x\left(w\left(z\right)\right)\left|J(x \to z)\right| \\
&= \left\{ \left(e^{-z/\alpha}\right)^{\alpha-1} \frac{1}{\Gamma(\alpha)} e^{-e^{-z/\alpha}} \left(-\frac{1}{\alpha} e^{-z/\alpha}\right) \right. \\
&= \left\{ \frac{1}{\Gamma(\alpha)\alpha} e^{-z-e^{-z/\alpha}} \right. \\
&= \left\{ \frac{1}{\Gamma(\alpha+1)} e^{-z-e^{-z/\alpha}} \right. \\
&= \left\{ \frac{1}{c_6} e^{-z-e^{-z/\alpha}} \quad -\infty < z < \infty. \right.
\end{aligned}
$$

Figure 8 shows how the transformation converges to an $Exp(1)$ and that the empirical pdf, using the **rangam** generator in **SAS**, of the gamma distribution is concentrated around zero as $\alpha \to 0$. Plots (a) to (c) compare the empirical pdf of (13) 'TAlpha_alpha_size' to (1) 'Alpha_alpha_size' for $\alpha = 1$, $\alpha = 0.1$ and $\alpha = 0.01$ respectively. Plot (d) overlays the theoretical pdf of an $Exp(1)$ distribution on the empirical pdf of (13) when $\alpha = 0.01$.



(a) (b) (c) (d)

Figure 8: Convergence of $Z = -\alpha log X$ to an $Exp(1)$.

4 Bivariate gamma simulation with small shape parameters

4.1 Bivariate gamma distribution

In the univariate case the gamma distribution with small shape parameter tends to be problematic to work with in a practical and simulation environment where it is found that general calculations become inaccurate and the distribution appears to be degenerate. The pdf of the product of two independent gamma variables also appears to be concentrated around zero (see Figure (9)) for α_1 and α_2 converging to zero simultaneously, hence simulating practically non-zero values remains challenging. In Figure 9 - 12 the bin values for the frequency plots are calculated using the Euclidean distance between x_1 and x_2. In the bivariate environment there are four cases to consider, namely:

1. α_1 remains constant $(\alpha_1 < 1)$ while $\alpha_2 \to 0$.
2. α_1 and α_2 converging to zero simultaneously.
3. α_1 increases from $(0 <)$ to 1 and α_2 decreases simultaneously from 1 to $(0 <)$.
4. $\alpha_2 \to 0$ and $\alpha_1 > 1$.

The premise of Section 4 is to focus on case 2.

This is a significant problem because the bivariate gamma distribution with gamma marginals has applications in the analysis of multivariate hydrological events [20].

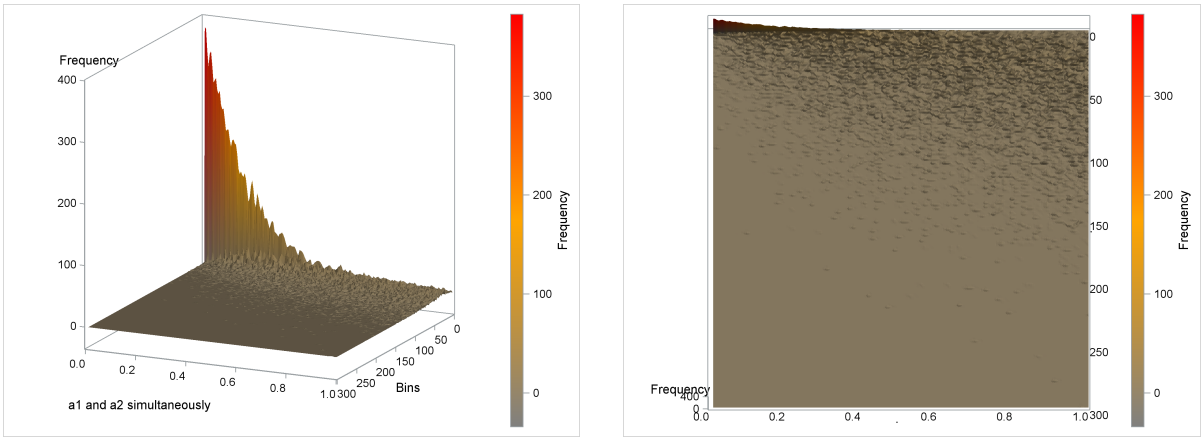


Figure 9: Empirical distribution and contour plot for α_1 and α_2 converging to zero simultaneously.

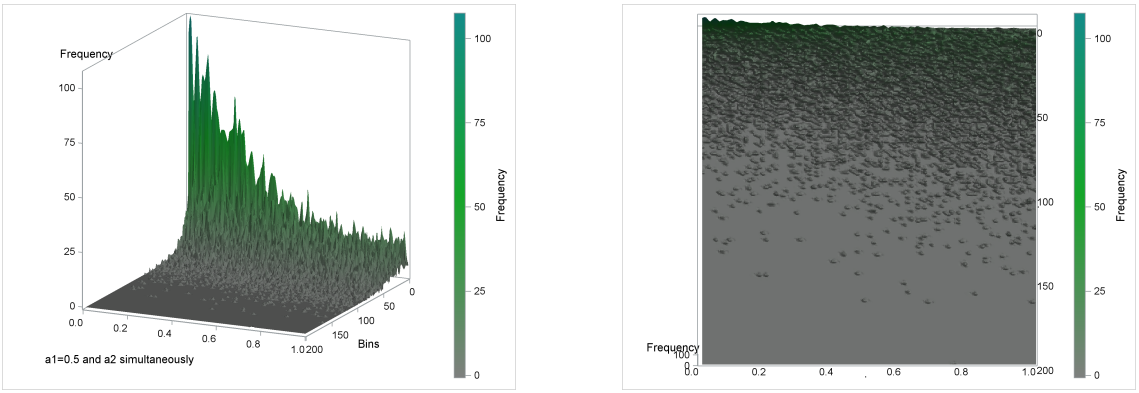


Figure 10: Empirical distribution and contour plot for α_1 remains constant ($\alpha_1 < 1$) while $\alpha_2 \to 0$.

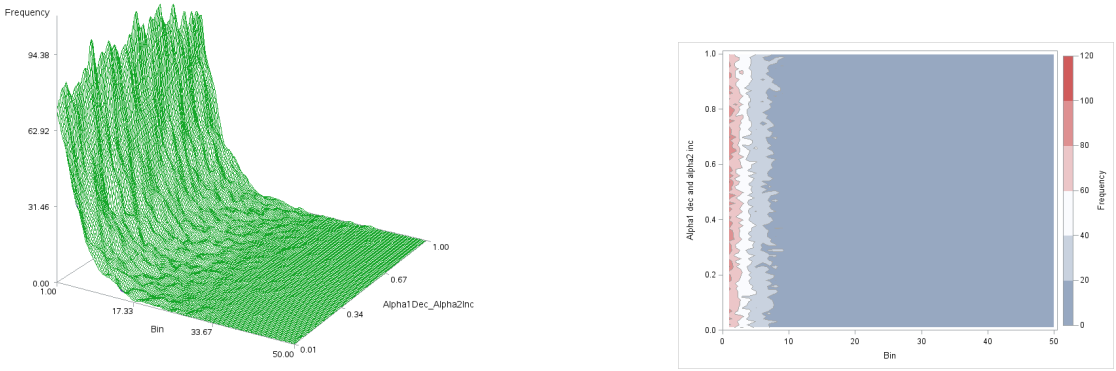
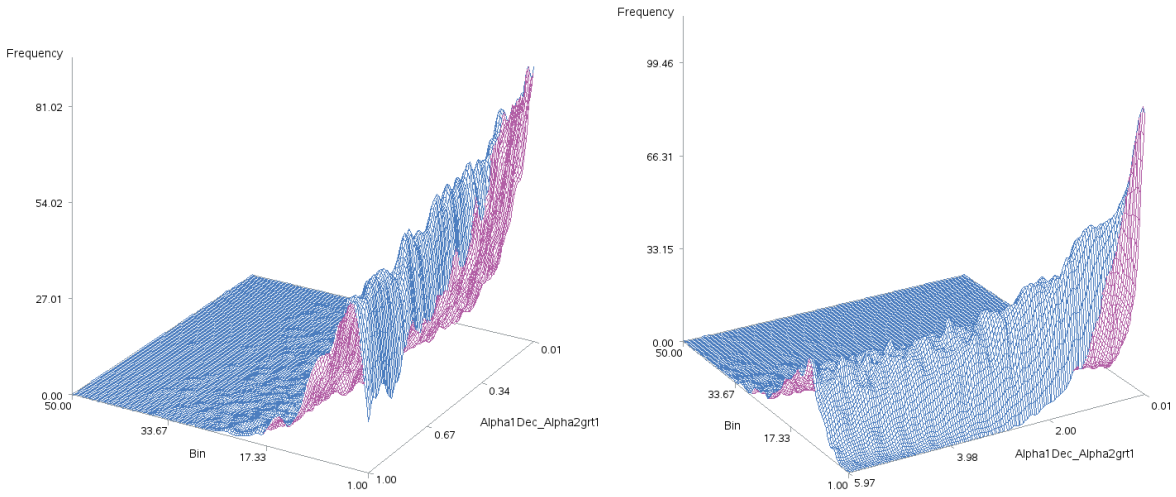


Figure 11: Empirical distribution and contour plot for α_1 increases from 0.01 to 1 and α_2 decreases simultaneously from 1 to 0.01.



(a) (b)

Figure 12: Empirical distribution for $\alpha_2 \to 0$ and $\alpha_1 = 1$ (a) and $\alpha_1 = 5$ (b).

Consider definition 3 for a joint pdf using [3].

Definition 3. The continuous random variables X and Y are said to be independent if $f_{X,Y}(x,y) = f_X(x).f_Y(y)$.

Suppose that X_1 and X_2 are independent and have pdf given by (1) with parameters $(\alpha_1,\ \beta_1 = 1)$ and $(\alpha_2,\ \beta_2 = 1)$ respectively. The joint pdf is given by

$$\begin{aligned} f_{X_1,X_2}(x_1,x_2) &= f_{X_1}(x_1).f_{X_2}(x_2) \\ &= \frac{1}{\Gamma(\alpha_1)}x_1^{\alpha_1-1}e^{-x_1}.\frac{1}{\Gamma(\alpha_2)}x_2^{\alpha_2-1}e^{-x_2} \\ &= \frac{1}{\Gamma(\alpha_1)\Gamma(\alpha_2)}e^{-(x_1+x_2)}x_1^{\alpha_1-1}x_2^{\alpha_2-1}. \end{aligned} \tag{18}$$

where $x_1,\ x_2\ > 0$.

The interest here is how the transformation $Z = -\alpha_i log Y_i$ given by [15] performs in the bivariate case for simulation purposes.

Firstly, consider Theorem 4 for the joint transformation of several random variables using [15].

Theorem 4. *Suppose that* $\boldsymbol{X}= (X_1, X_2, \ldots, X_k)$ *is a k-variate random variable with a joint pdf* $f_{X_1,X_2,\ldots,X_k}(x_1,x_2,\ldots,x_k) > 0$ *on a set A, and* $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_k)$ *is defined by the one-to-one transformation;*

$$Z_i = u_i(X_1, X_2, \ldots, X_k) \ \textit{for}\ i = 1, 2, \ldots, k.$$

On condition that the Jacobian is continuous and non-zero over the range of the transformation, then the joint pdf of $\boldsymbol{Z}$ *is*

$$f_{Z_1,Z_2,\ldots,Z_k}(z_1,z_2,\ldots,z_k) = f_{X_1,X_2,\ldots,X_k}(x_1,x_2,\ldots,x_k)\,|J|$$

where $\boldsymbol{x}= (x_1, x_2, \ldots, x_k)$ *is the solution of* $\boldsymbol{z}=\boldsymbol{u}(x)$.

The joint pdf of X_1 and X_2 is given by (18) where $A = \{(x_1,x_2)\,|0 < x_1,\ 0 < x_2\}$. The random variables $Z_1 = -\alpha_1 log X_1$ and $Z_2 = -\alpha_2 log X_2$ are independent and have unique solutions $X_1 = e^{-\frac{Z_1}{\alpha_1}}$ and $X_2 = e^{-\frac{Z_2}{\alpha_2}}$. The Jacobian is given by

$$J = \begin{vmatrix} \frac{-e^{-\frac{z_1}{\alpha_1}}}{\alpha_1} & 0 \\ 0 & \frac{-e^{-\frac{z2}{\alpha_2}}}{\alpha_2} \end{vmatrix} = \frac{e^{-\left(\frac{Z_1}{\alpha_1}+\frac{Z_2}{\alpha_2}\right)}}{\alpha_1\alpha_2}$$

Therefore

$$
\begin{aligned}
f_{Z_1,Z_2}(z_1,z_2) &= f_{X_1,X_2}\left(e^{-\frac{z_1}{\alpha_1}}, e^{-\frac{z_2}{\alpha_2}}\right)\frac{e^{-\left(\frac{z_1}{\alpha_1}+\frac{z_2}{\alpha_2}\right)}}{\alpha_1\alpha_2} \\
&= \left\{\frac{1}{\Gamma(\alpha_1)\Gamma(\alpha_2)}e^{-\left(e^{-\frac{z_1}{\alpha_1}}+e^{-\frac{z_2}{\alpha_2}}\right)}\left(e^{-\frac{z_1}{\alpha_1}}\right)^{\alpha_1-1}\left(e^{-\frac{z_2}{\alpha_2}}\right)^{\alpha_2-1}\frac{e^{-\left(\frac{z_1}{\alpha_1}+\frac{z_2}{\alpha_2}\right)}}{\alpha_1\alpha_2}\right. \\
&= \left\{\left(e^{-z_1/\alpha_1}\right)^{\alpha_1-1}\frac{1}{\Gamma(\alpha_1)}e^{-e^{-z/\alpha}}\left(-\frac{1}{\alpha_1}e^{-z_1/\alpha_1}\right)\left(e^{-z_2/\alpha_2}\right)^{\alpha_2-1}\frac{1}{\Gamma(\alpha_2)}e^{-e^{-z_2/\alpha_2}}\left(-\frac{1}{\alpha_2}e^{-z_2/\alpha_2}\right)\right. \\
&= \left\{\frac{1}{\Gamma(\alpha_1+1)}e^{-z_1-e^{-z_1/\alpha_1}}\frac{1}{\Gamma(\alpha_2+1)}e^{-z_2-e^{-z_2/\alpha_2}}\right. \qquad (19) \\
&= \left\{\frac{1}{c_6^*}e^{-z_1-e^{-z_1/\alpha_1}}\frac{1}{c_6^*}e^{-z_2-e^{-z_2/\alpha_2}} \quad z_1,\ z_2\ \epsilon\ (-\infty,\infty)\right..
\end{aligned}
$$

4.2 Limit distribution result with transformations $Z_i = -\alpha_i log X_i$

Consider the following extension of Theorem 1.

Theorem 5. *For $X_1 \sim Gamma(\alpha_1, 1)$ and $X_2 \sim Gamma(\alpha_2, 1)$ as α_1 and α_2 converge to zero simultaneously the joint pdf of Z_1 and Z_2 converges to the joint pdf of the product of two independent $Exp(1)$ random variables.*

Proof.

Firstly, note that by theorem 3.3.1 in [3],

$$
\begin{aligned}
\Gamma(\alpha_j+1) &= \alpha_j\Gamma(\alpha_j)\ for\ j=1,\,2 \\
\therefore \Gamma(\alpha_j) &= \frac{\Gamma(\alpha_j+1)}{\alpha_j}.
\end{aligned}
$$

Secondly,

$$
\begin{aligned}
E\left(X_1^kX_2^k\right) = E\left(X_1^k\right)E\left(X_2^k\right) &= \int_0^\infty x_1^k\frac{x_2^{(\alpha_1-1)}e^{-x_1}}{\Gamma(\alpha_1)}dx_1\int_0^\infty(x)\,\frac{x_2^{(\alpha_2-1)}e^{-x_2}}{\Gamma(\alpha_2)}dx_2 \qquad (independence) \\
&= \frac{\Gamma(\alpha_1+1)}{\Gamma(\alpha_1)}\frac{\Gamma(\alpha_2+1)}{\Gamma(\alpha_2)}
\end{aligned}
$$

Consider the following transformations:

$$Z_1 = -\alpha_1 log X_1\ and\ Z_2 = -\alpha_2 log X_2.$$

Since Z_1 and Z_2 are linear functions of the independent random variables it follows that Z_1 and Z_2 are also independent, the independence of Z_1 and Z_2 implies the independence of $e^{it_1Z_1}$ and $e^{it_2Z_2}$. The

characteristic function is given by

$$\begin{aligned}\Phi_{z_1,\,z_2}(t_1,\,t_2) &= E\left(e^{i(t_1Z_1+t_2Z_2)}\right)\\ &= E\left(e^{it_1Z_1}\right)E\left(e^{it_2Z_2}\right)\\ &= E\left(X_1^{-i\alpha_1t_1}\right)E\left(X_1^{-i\alpha_2t_2}\right)\\ &= \frac{\Gamma(\alpha_1-i\alpha_1t_1)}{\Gamma(\alpha_1)}\frac{\Gamma(\alpha_2-i\alpha_2t_2)}{\Gamma(\alpha_2)}.\end{aligned}$$

This expression can be rewritten as

$$\begin{aligned}\Phi_{z_1,\,z_2}(t_1,\,t_2) &= \left(\frac{\Gamma(1+\alpha_1-i\alpha_1t_1)/(\alpha_1-i\alpha_1t_1)}{\Gamma(1+\alpha_1)/\alpha_1}\right)\left(\frac{\Gamma(1+\alpha_2-i\alpha_2t_2)/(\alpha_2-i\alpha_2t_2)}{\Gamma(1+\alpha_2)/\alpha_2}\right)\\ &= \left(\frac{1}{1-it_1}\right)\left(\frac{1}{1-it_2}\right)\left(\frac{\Gamma(1+0_{\alpha_1})}{\Gamma(1+0_{\alpha_1})}\right)\left(\frac{\Gamma(1+0_{\alpha_2})}{\Gamma(1+0_{\alpha_2})}\right)\end{aligned}$$

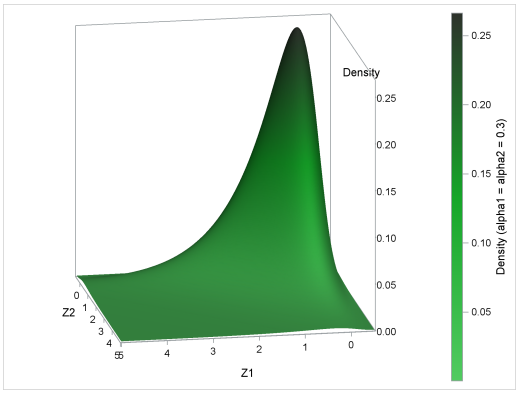
where $0(\alpha_j)$ *becomes insignificant as* $\alpha_j \to 0$. *Given that the gamma function is continuous on* $x_j > 0$ *therefore the limit of* $d_{\mathbf{z}}(\mathbf{t})$ *as* $\alpha_j \to 0$ *exists and is given by* $\left(\frac{1}{1-it_1}\right)\left(\frac{1}{1-it_2}\right)$ *which is the product of two independent* $Exp(1)$ *characteristic functions.*

Figure 13 shows how the product of two independent $Exp(1)$ distributions can be used as an envelope function for $f_{Z_1,Z_2}(z_1,z_2)$ as $\alpha_1 \to 0$ and $\alpha_2 \to 0$ simultaneously. Plots (a) and (b) are rotated plots of the bivariate target pdf for $\alpha_1 = \alpha_2 = 0.3$. Plots (c) and (d) are rotated plots of the bivariate target pdf for $\alpha_1 = \alpha_2 = 0.01$. Plots (e) and (f) are rotated plots of the product of two independent $Exp(1)$ distributions i.e. the envelope function. Plots (g) and (h) are rotated plots indicating how compactly the envelope function fits over $f_{Z_1,Z_2}(z_1,z_2)$ for $\alpha_1 = \alpha_2 = 0.01$.

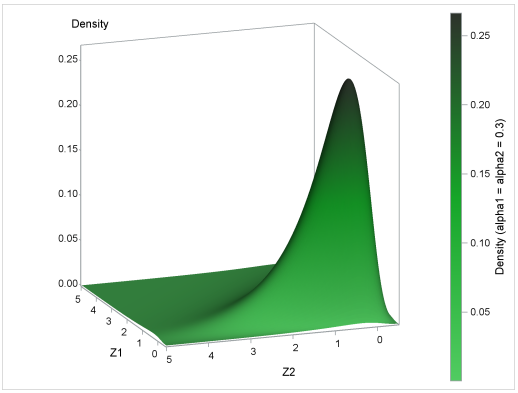
Remark 6. For $\alpha_i \to 0$ while α_j remains constant/increases for $i \neq j$ the characteristic function $d_{\mathbf{z}}(\mathbf{t})$ can be viewed as a 'mirror' function because the form of the final result remains the same i.e.

$$\begin{aligned}\Phi_{z_1,\,z_2}(t_1,\,t_2) &= \frac{\Gamma(\alpha_1-i\alpha_1t_1)}{\Gamma(\alpha_1)}\frac{\Gamma(\alpha_2-i\alpha_2t_2)}{\Gamma(\alpha_2)}\\ &= \frac{\frac{\Gamma(\alpha_1-i\alpha_1t_1+1)}{(\alpha_1-i\alpha_1t_1)}}{\frac{\Gamma(\alpha_1+1)}{\alpha_1}}\frac{\frac{\Gamma(\alpha_2-i\alpha_2t_1+1)}{(\alpha_2-i\alpha_2t_2)}}{\frac{\Gamma(\alpha_2+1)}{\alpha_2}}\\ &= \left(\frac{1}{1-it_1}\right)\left(\frac{1}{1-it_2}\right)\frac{\Gamma(\alpha_1-i\alpha_1t_1+1)}{\Gamma(\alpha_1+1)}\frac{\Gamma(\alpha_2-i\alpha_2t_1+1)}{\Gamma(\alpha_2+1)}.\end{aligned}$$

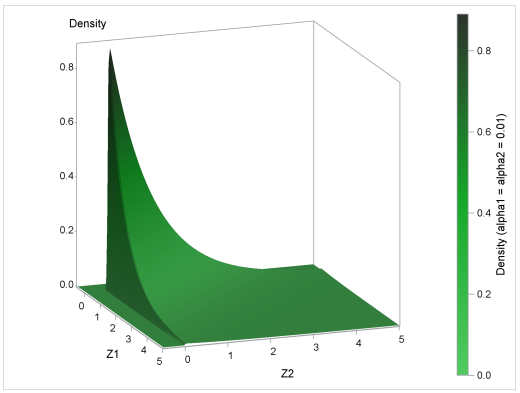
Let $\alpha_2 \to 0$



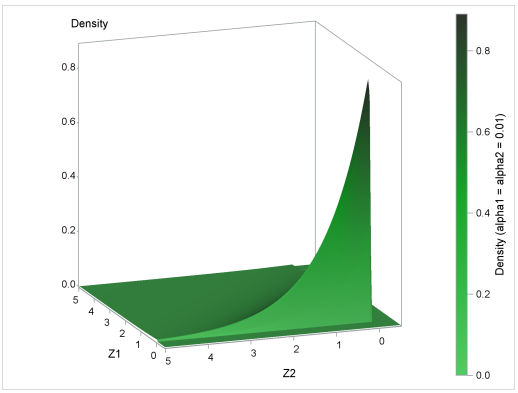
(a)



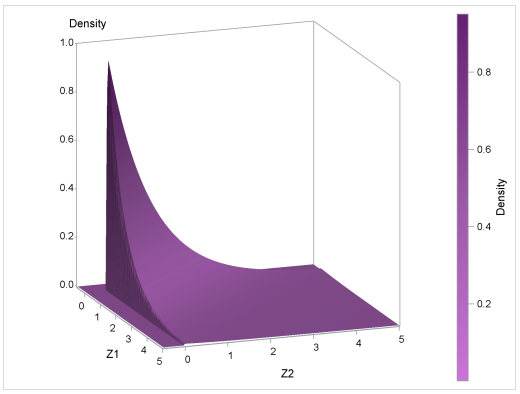
(b)



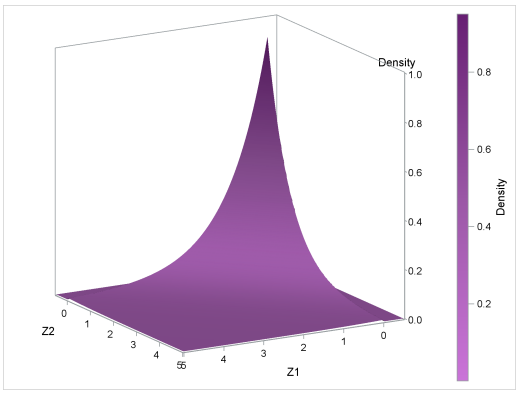
(c)



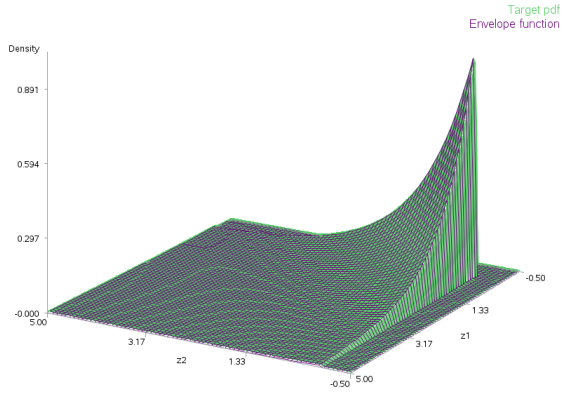
(d)



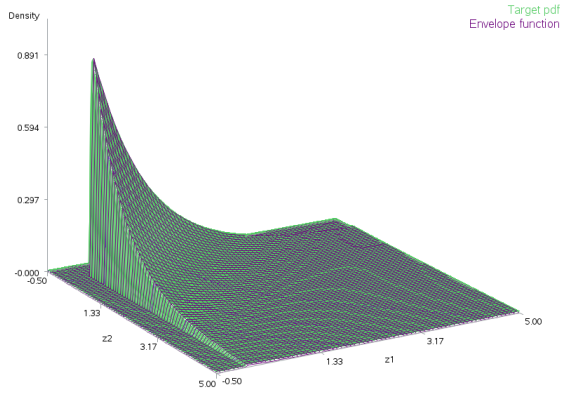
(e)



(f)



(g)



(h)

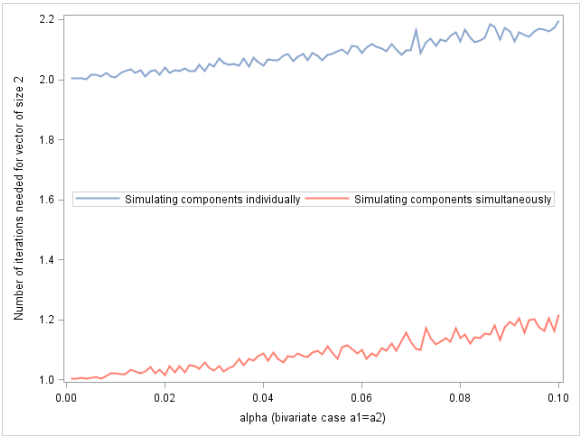
Figure 13: Target pdf, $f_{Z_1,Z_2}(z_1, z_2)$ for various α_1 and α_2 (a to d). Envelope function (e and f). Envelope and target pdf (g and h).

$$\Phi_{z_1,\,z_2}(t_1,\,t_2) = \left(\frac{1}{1-it_1}\right)\left(\frac{1}{1-it_2}\right)\frac{\Gamma(\alpha_1 - i\alpha_1 t_1 + 1)}{\Gamma(\alpha_1+1)}$$
$$= \left(\frac{1}{1-it_1}\right)\left(\frac{1}{1-it_2}\right)\frac{\Gamma(\alpha_1\,[1-it_1]+1)}{\Gamma(\alpha_1+1)}.$$

Similar results follow if $\alpha_1 \to 0$, resulting in the product of two independent characteristic functions of an $Exp(1)$ and the ratio of gamma functions.

4.3 Simulating bivariate gamma random variables using logarithmic transformations

Acceptance-rejection methods are used extensively in the univariate generation of random variables, however the extension to the multivariate case poses some significant practical difficulties. Let $\mathbf{c} = [\mathrm{c}_1, \ldots, \mathrm{c}_\mathrm{n}]$ be a vector where c_i is the i^{th} component of $\boldsymbol{c}$ and similarly for $\mathbf{x} = \mathrm{vec(x)}$. The difficulty arises in finding a suitable envelope function $\boldsymbol{c}g(\boldsymbol{x}) = h(\boldsymbol{x})$ if there is a strong dependence among the components of $\mathbf{X}$. An appropriate choice for $g(\boldsymbol{x})$ would be the pdf of the independent components of $\mathbf{X}$ whom have the same marginal distribution as $\mathbf{X}$ [10]. It should be noted though that as the dependencies among the components of $\mathbf{X}$ increase the acceptance rates tend to zero [10]. In addition the complexity of the envelope function would make the search for $\boldsymbol{c} = sup\frac{f(\boldsymbol{x})}{h(\mathbf{x})}$ where $\boldsymbol{x} \in R^p$, quite challenging. In the bivariate case the goal is to sample two components from the product of two independent gamma(α_i, 1) distributions which can be achieved by simulating each X_i component from the independent marginal distributions using Algorithm 7. It should be noted, however that components can also be sampled simultaneously, baring in mind that simultaneous simulation may increase the complexity of the program (see **SAS** code in Appendix).



(a)

Figure 14: Number of iterations required to simulate 2 random components from $f_{Z_1,Z_2}(z_1, z_2)$ for decreasing α_1 and α_2 based on 30 simulations (a).

Consider Algorithm (8) as an extension of Algorithm (7) to generate two points simultaneously,

Algorithm 8 Ryan Martin extension to bivariate case.

1. Set $\lambda_1 = \lambda_1(\alpha_1)$, $\lambda_2 = \lambda_2(\alpha_2)$, $w_1 = w_1(\alpha_1)$, $w_2 = w_2(\alpha_2)$ and $r_1 = r_1(\alpha_1)$, $r_2 = r_2(\alpha_2)$.
2. Generate $U_1 \sim uniform(0,1)$ and $U_2 \sim uniform(0,1)$.
3. If $U_1 \leq r_1$ and $U_2 \leq r_2$ then $Z_1 = -log(U_1/r_1)$ and $Z_2 = -log(U_2/r_2)$.
4. If $U_1 \leq r_1$ and $U_2 > r_2$ then $Z_1 = -log(U_1/r_1)$ and $Z_2 = -log(U_2)/\lambda_2$.
5. If $U_2 \leq r_2$ and $U_1 > r_1$ then $Z_2 = -log(U_2/r_2)$ and $Z_1 = -log(U_1)/\lambda_1$.
6. If $U_1 > r_1$ and $U_2 > r_2$ then $Z_1 = -log(U_1)/\lambda_1$ and $Z_2 = -log(U_2)/\lambda_2$.
7. If $\frac{f_{Liu}(z_1)}{h_{Liu}(z_1)} > U_1$ and $\frac{f_{Liu}(z_2)}{h_{Liu}(z_2)} > U_2$ then return Z_1 and Z_2.
8. Return $Y_1 = e^{-Z_1/\alpha_1}$ and $Y_2 = e^{-Z_2/\alpha_2}$.

An alternative method to generate a vector $\mathbf{X}$ of n components would be to generate each component, X_i, individually i.e. to repeat Algorithm 7 n times. Algorithm 9 is a simple adaption of algorithm 7.

Algorithm 9 Ryan Martin for individual components.

1. Set $\lambda_i = \lambda_i(\alpha_i)$, $w_i = w_i(\alpha_i)$ and $r_i = r_i(\alpha_i)$.
2. Generate $U \sim uniform(0,1)$.
3. If $U \leq r_i$ then $z_i = -log(U/r_i)$, otherwise $z_i = -\frac{log(U)}{\lambda_i}$.
4. If $\frac{f_{Liu}(z_i)}{h_{Liu_1}(z_i)} > U$ then return $Z_i = z_i$.
5. Return $Y_i = e^{-Z_i/\alpha_i}$.
6. After n iterations return $\mathbf{Y} = [\mathrm{Y_1}, \ldots, \mathrm{Y_n}]^{'}$.

The choice of which algorithm to use is up to the end user, however from Figure 14 it is clear that simulating the random variates simultaneously requires fewer iterations than simulating the components individually.

5 Application

5.1 Efficiency in the univariate case

All the methods considered can be used to generate samples from the target pdf given by (1) for small α using an acceptance-rejection technique. The most suitable way to compare these methods is to use the acceptance rate $\frac{1}{c}$ (proportion of sampled variates accepted). Having a high acceptance rate is preferable

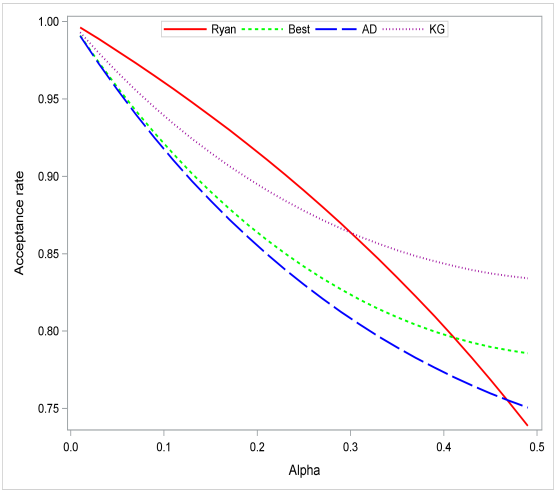
and therefore minimising c is beneficial. Table 1 provides the comparative acceptance rate expressions for [2, 4, 13, 15]. Figure 15 compares the acceptance rates for [2, 4, 13, 15] for decreasing α respectively. For all the methods discussed it is clear that the acceptance rate converges to 1 as $\alpha \to 0$, however $r(a)$ converges quicker for $\alpha \leq 0.3$ thus the method proposed by [15] is highly efficient for small α values which is clear from Theorem 1. For $\alpha \in (0.3, 0.33)$ Algorithm 7 is less efficient than that of Algorithm 6 and for $\alpha > 0.55$ Algorithm 7 becomes the most inefficient.

Ryan Martin	Ahrens-Dieter	Best	Kundu-Gupta
$\left(1+\frac{\alpha}{e(1-\alpha)}\right)^{-1}$	$\frac{\Gamma(\alpha+1)e^1}{(\alpha+e^1)}$	$\frac{\Gamma(\alpha+1)e^z}{z^\alpha(e^z+\frac{\alpha}{z})}$	$\left[\frac{1}{\alpha\Gamma(\alpha)}\left(2^\alpha\left[1-e^{-b(\alpha)/2}\right]^\alpha+\alpha b(\alpha)^{\alpha-1}e^{-b(\alpha)}\right)\right]^{-1}$

Table 1: Acceptance rates for the four indicated univariate methods.

Obs	AD	Best	KG	Ryan	Alpha
1	0.99068	0.99098	0.99306	0.99630	0.01
2	0.97281	0.97374	0.97978	0.98875	0.03
3	0.95592	0.95754	0.96729	0.98101	0.05
4	0.93995	0.94230	0.95553	0.97306	0.07
5	0.92484	0.92798	0.94447	0.96489	0.09
6	0.91055	0.91453	0.93408	0.95651	0.11
7	0.89703	0.90191	0.92433	0.94789	0.13
8	0.88425	0.89009	0.91519	0.93904	0.15
9	0.87216	0.87903	0.90663	0.92993	0.17
10	0.86072	0.86868	0.89862	0.92056	0.19
11	0.84992	0.85904	0.89115	0.91092	0.21
12	0.83971	0.85006	0.88419	0.90099	0.23
13	0.83006	0.84172	0.87773	0.89077	0.25
14	0.82096	0.83401	0.87175	0.88023	0.27
15	0.81237	0.82690	0.86623	0.86937	0.29
16	0.80428	0.82037	0.86116	0.85816	0.31

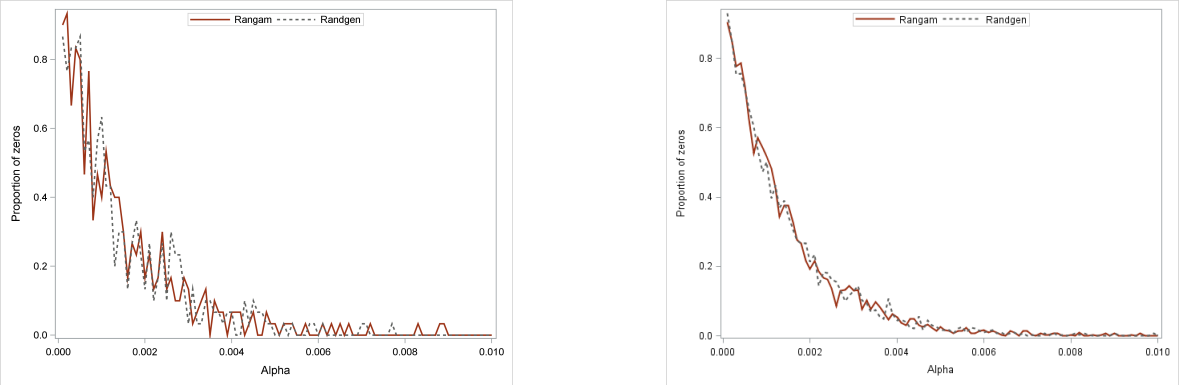
(a)



(b)

Figure 15: Acceptance rates for $\alpha < 0.5$.

Figure 16 illustrates how the standard methods used by **SAS** (**rangam** and **randgen**('GAMMA')) become highly inaccurate in terms of simulating practically non-zero values when the shape parameter becomes 'small'. In addition to being computationally efficient in terms of having higher acceptance rates in shorter simulation time, Algorithm 7 does not suffer from this problem. Therefore as suggested in [15] for $\alpha < 0.1$ Algorithm 7 should be used in simulation and when $\alpha \geq 0.1$ it is suggested that the **randgen**('GAMMA') function be used instead of the **rangam** generator (see Appendix and **SAS** help for the advantages and comparisons between the **rangam** and **randgen** generator).



(a) (b)

Figure 16: Proportion of zeros for **rangam** and **randgen** for sample size of 30 (a) and 300 (b).

5.2 Efficiency in the bivariate case

Simulating vectors from multivariate distributions where the cumulative density function does not have a closed form expression can be quite challenging and computationally strenuous. The intention of acceptance-rejection sampling is to identify an envelope function that is easier to simulate from and where the range of the target pdf is a subset of the range of the envelope function and both are defined on the same support. Identifying an envelope function in higher dimensions can become extremely difficult. As the dimension increases the acceptance rate tends to decrease exponentially [12].

Proposition 7. *The efficiency of a bivariate acceptance-rejection algorithm, assuming independence of the components, is expressed in terms of the acceptance rate given by* $\frac{1}{c^2}$.

Proof. The probability of generating points X_1 from $h_1(x)$ that falls under $f_1(x)$ and X_2 from $h_2(x)$ that falls under $f_2(x)$ simultaneously is given by

$$p = P\left(\left[U_1 \leq \frac{f_1(x)}{ch_1(x)}\right] \cap \left[U_2 \leq \frac{f_2(x)}{ch_2(x)}\right]\right) = P\left(U_1 \leq \frac{f_1(x)}{ch_1(x)}\right) P\left(U_2 \leq \frac{f_2(x)}{ch_2(x)}\right)$$

since $U_1 \sim uniform(0,\, 1)$ and $U_2 \sim uniform(0,\, 1)$ are independent of each other. Note that $\frac{f_1(x)}{h_1(x)}$ is independent of $\frac{f_2(x)}{h_2(x)}$ as well.

Therefore following directly from the proof in section 2 the acceptance rate for the bivariate case is given by $\left(\frac{1}{c}\right)^2$. This can be generalised to any multivariate dimension under the assumption of independence. □

6 Conclusion

The challenge of simulating gamma variates with small shape parameter(s) in the univariate and bivariate environment can be dealt with (to a certain degree) using an acceptance-rejection method based on a limit distribution approximation of a gamma distribution with a small shape parameter. The proposed algorithm (Algorithm 7) can be extended from the **R** software package to **SAS**, and when compared to various existing algorithms, including the **rangam** and **randgen** generators used in **SAS**, it proves to be more efficient for small shapes. In the case of the bivariate gamma distribution (under independence), specifically where both shape parameters tend to zero simultaneously, it remains challenging to simulate gamma variates that differ from zero. Algorithm 7 can be extended to the bivariate case. Two variations of the bivariate algorithm can be considered; the first simulates each component of the vector individually (Algorithm 9) and the second simulates the components simultaneously (Algorithm 8). Simultaneous simulation of the vector components require fewer iterations than the individual simulation, however the algorithms may become slightly more complex.

Adaptions to the acceptance-rejection method in the multivariate environment where the gamma variates are dependent can be investigated in future studies where a new acceptance-rejection algorithm would have to be formulated and the envelope function investigated. In addition adaptions from the univariate limiting distribution result to the remaining three cases for the bivariate distribution can be studied where consideration of the ratio of gamma functions needs to be evaluated.

References

[1] J. H. Ahrens and U. Dieter. Computer methods for sampling from gamma, beta, poisson and binomial distributions. *Computing (Arch. Elektron. Rechnen)*, 12(3):223–246, 1974.

[2] A. C. Atkinson and M. C. Pearce. The computer generation of beta, gamma and normal random variables. *Journal of the Royal Statistical Society. Series A (General)*, pages 431–461, 1976.

[3] L. J. Bain and M. Engelhardt. *Introduction to Probability and Mathematical Statistics*. The Duxbury advanced series in statistics and decision sciences. Brooks/Cole, 2nd edition, 1987.

[4] D. J. Best. A note on gamma variate generators with shape parameter less than unity. *Computing*, 30(2):185–188, 1983.

[5] R. C. H. Cheng. The generation of gamma variables with non-integral shape parameter. *Applied statistics*, 26(1):71–75, 1977.

[6] R. C. H. Cheng and G. M. Feast. Some simple gamma variate generators. *Applied Statistics*, 28(3):290–295, 1979.

[7] T. H. Cormen. *Introduction to Algorithms*. MIT press, 3rd edition, 2009.

[8] G. S. Fishman. *Principles of Discrete Event Simulation*. Wiley series on systems engineering and analysis. John Wiley & Sons, 1978.

[9] Kotz S. Johnson, N. L. and N. Balakrishnan. *Continuous Univariate Probability Distributions*, volume 1 of *Wiley series in probability and mathematical statistics: applied probability and statistics*. John Wiley & Sons, New York, 2nd edition, 1994.

[10] M. E. Johnson. *Multivariate Statistical Simulation*. Wiley series in probability and mathematical statistics. Applied probability and statistics. John Wiley & Sons, 2013.

[11] C. Kleiber and S. Kotz. *Statistical Size Distributions in Economics and Actuarial Sciences*, volume 470 of *Wiley series in probability and statistics*. John Wiley & Sons, Hoboken, 2003.

[12] Taimre T. Kroese, D. P. and Z. I. Botev. *Handbook of Monte Carlo Methods*, volume 706 of *Wiley series in probability and statistics*. John Wiley & Sons, June 2013. page 66.

[13] D. Kundu and R. D. Gupta. A convenient way of generating gamma random variables using generalized exponential distribution. *Computational Statistics & Data Analysis*, 51(6):2796–2802, 2007.

[14] K. Lange. *Numerical Analysis for Statisticians*. Springer Science & Business Media, 2010.

[15] Martin R. Liu, C. and N. Syring. Efficient simulation from a gamma distribution with small shape parameter. *Computational Statistics*, pages 1–9, 2016. doi:10.1007/s00180-016-0692-0.

[16] G. Marsaglia. The squeeze method for generating gamma variates. *Computers & Mathematics with Applications*, 3(4):321–325, 1977.

[17] G. Marsaglia and W. W. Tsang. A simple method for generating gamma variables. *Association for Computing Machinery Transactions on Mathematical Software*, 26(3):363–372, 2000.

[18] Nishimura T. Niederreiter H. Matsumoto, M. and J. Spanier. Dynamic creation of pseudorandom number generators. *Monte Carlo and Quasi-Monte Carlo Methods*, 2000:56–69, 1998.

[19] B. W. Schmeiser and R. Lal. Squeeze methods for generating gamma variates. *Journal of the American Statistical Association*, 75(371):679–682, 1980.

[20] Y. Sheng. A bivariate gamma distribution for use in multivariate flood frequency analysis. *Hydrological Processes*, 15(6):1033–1045, 2001.

[21] E. W. Stacy. A generalization of the gamma distribution. *The Annals of Mathematical Statistics*, 33(3):1187–1192, September 1962.

[22] J. Stewart. *Calculus: Early Transcendentals*. Cengage Learning, 7th edition, 2010.

[23] H. Tanizaki. A simple gamma random number generator for arbitrary shape parameters. *Economics Bulletin*, 3(7):1–10, 2008.

[24] R. Wicklin. Six reasons you should stop using the ranuni function to generate random numbers. SAS blog: The DO Loop, September 2017. http://blogs.sas.com/content/iml/2013/07/10/stop-using-ranuni.html.

Appendix

6.1 $O(h)$ functions

In mathematics $O(h)$ notation describes the behavior of a function as the parameter of interest, say h, tends towards a particular value or infinity.

Definition 8. A function $f : \Re \to \Re$ is $o(h)$ if $\lim_{h \to 0} \frac{f(h)}{h} = 0$ [7].

6.2 Logarithmic concave functions

Definition 9. A function $f(x) : \Re \to \Re$ is defined as convex if for any x and $y \in \Re$ and for every $\beta \in [0,1]$, $f[\beta x + (1-\beta) y] \leq \beta f(x) + (1-\beta) f(y)$. $f(x)$ is concave if $-f(x)$ is convex [22].

Definition 10. A function $f(x)$ $\Re^n \to \Re$ is log-concave if $logf[\beta x + (1-\beta) y] \geq \beta logf(x) + (1-\beta) logf(y)$ for every $\beta \in [0,1]$ [22].

6.3 RANGAM vs RANDGEN

The rangam generator returns a variate that is generated from a gamma distribution. For shape parameter ≤ 1 a rejection method proposed by [8] is used. The RANDGEN('GAMMA')/RAND function uses a random number generator developed by [18]. The advantages of using the RAND generator over the RANGAM generator can be summarised as follows as described by [24]:

1. **A longer period**:

   The period of a random generator refers to the number of values that can be generated before repetitions occur. The period of the RANGAM generator is 2^{31} and the corresponding RANDGEN function is 2^{19937}.

2. **Superior statistical properties:**

   The randomness obtained in simulating a random sample (stream of random numbers) is more accurate for the RANDGEN generator compared to the RANGAM generator. In essence this means that the samples generated by the RANDGEN function have the correct proportion of duplicate values in a simulated sample when compared to the RANGAM function.

3. **A simpler specification of the seed values:**

   The syntax for the RANGAM function requires the specification of a seed value each time the function is called which may lead to the misconception that changing the seed on consecutive calls may result in a different stream of random numbers, however this is not true since all random number seeds except the first one, are completely ignored. The RANDGEN function clearly indicates that

a single stream of random numbers is created. The CALL RANDSEED routine can be used to set the random seed number for the RANDGEN function.

4. **Uniform syntax:**

   The RANDGEN function uses the same syntax as other SAS functions for dealing with probability distributions such as the pdf and quantile functions for the gamma distribution.

5. **Improved handling of wider parameter space regions:**

   The RANDGEN function uses more recent algorithms that handle situations in where a distribution tends to become degenerate as the parameter(s) tend to a limiting region.

6. **Continued development and support:**

   Support for new distributions are added to the RANDGEN function in SAS, where as support for the RANXXX functions are no longer developed.

6.4 The ratio-of-uniforms method

The ratio-of uniforms method is similar to the acceptance-rejection method, having the advantage that the form of the target pdf only needs to be known up to a normalising constant [12]. The pdf from which sampling takes place has the following form

$$f(z) = ch(z)$$

where $h(z)$ is known but c may not be. Note that as before we require $c > 0$.
The methodology can be summarised as follows:

1. Generate a point (X, Y) uniformly over the set
   $\mathbb{C} = \left\{(x, y) : \ 0 \leq x \leq \sqrt{h\left(\frac{y}{x}\right)}\right\}$.
2. Return $Z = \frac{Y}{X}$.

6.5 Algorithm 7 adjusted for SAS

The following script has been adapted from the rgmass script given by [15].

```
proc iml; test=0;
Start RJgam(n,shape,scale=1);
a=shape;
e=2.71828182845905;
l=(1/a)-1;
```

```
w=a/(e*(1-a));
r=inv(1+w);
seed=1;
const=(1/gamma(a+1));
scale=1; y=J(n,1,0);
do i=1 to n;
do until(f/h > ranuni(seed));
u=ranuni(seed);
if u <= r then z=-log(u/r);
else  z=log(ranuni(seed))/l;
f=const*exp(-z-exp(-z/a));
if z>=0 then h=const*exp(-z);
else  h=const*w*l*exp(l*z);
if f/h > ranuni(seed) then q=z;
end;
y[i,1]=log(scale) - (q/a);
end;
return(y);
finish;
n=30;
alpha=0.0001;
x=RJgam(n,alpha,1);
k=J(n,1,0);
do i=1 to n;
ran=rangam(1,alpha);
if ran <= test then  k[i]= .;
else k[i]= log(ran);
end;
l=J(n,1,0);
call randgen(l, 'GAMMA', alpha);
do i=1 to n;
ran=rangam(1,alpha);
if l[i] <= test then  l[i]= .;
else l[i]= log(l[i]);
```

```
end;
d=x||k||l;
cn={"RJgamma" "rangam" "randgen"};
create Data from d[colname=cn];
append from d;
close;
print x;
```

6.6 Algorithm 8 adjusted for SAS

The following script has been adapted from the rgmass script given by [15].

```
proc iml;
test=0;
Start RJgam(n,shape1,shape2,scale=1);
a1=shape1;
a2=shape2;
e=2.71828182845905;
l1=(1/a1)-1; l2=(1/a2)-1;
w1=a1/(e*(1-a1));
w2=a2/(e*(1-a2));
r1=inv(1+w1);
r2=inv(1+w2);
seed=1;
const1=(1/gamma(a1+1));
const2=(1/gamma(a2+1));
scale=1; y=J(n,2,0);
do i=1 to n;
do until(f1/h1 > ranuni(seed) & f2/h2 > ranuni(seed) );
u1=ranuni(seed); u2=ranuni(seed);
if (u1 <= r1) & (u2 <= r2) then z1=-log(u1/r1) ;
if u1 <= r1 & u2 <= r2 then z2=-log(u2/r2) ;
if u1 <= r1 & u2 > r2 then z1=-log(u1/r1) ;
if u1 <= r1 & u2 > r2 then z2=log(ranuni(seed))/l1 ;
if u2 <= r2 & u1 > r1 then z2=-log(u2/r2) ;
```

```
if u2 <= r2 & u1 > r1 then z1=log(ranuni(seed))/l2 ;
if u1 > r1 & u2 > r2 then z1=-log(u1/r1) ;
if u1 > r1 & u2 > r2 then z2=-log(u2/r2) ;
f1=const1*exp(-z1-exp(-z1/a1));
f2=const2*exp(-z2-exp(-z2/a2));
if z1>=0 then h1=const1*exp(-z1);
else h1=const1*w1*l1*exp(l1*z1);
if z2>=0 then h2=const2*exp(-z2);
else h2=const2*w2*l2*exp(l2*z2);
if f1/h1 > ranuni(seed) then q1=z1;
if f2/h2 > ranuni(seed) then q2=z2;
end;
y[i,1]=log(scale) - (q1/a1);
y[i,2]=log(scale) - (q2/a2);
end;
return(y);
finish;
n=30;
alpha1=0.001;
alpha2=alpha1;
x=RJgam(n,alpha1,alpha2,1);
print x;
```

The following script has been adapted from the rgmass script given by [15].

```
proc iml;
test=0;
Start RJgam(n,shape1,shape2,scale=1);
a1=shape1;
a2=shape2;
e=2.71828182845905;
l1=(1/a1)-1; l2=(1/a2)-1;
w1=a1/(e*(1-a1));
w2=a2/(e*(1-a2));
r1=inv(1+w1);
r2=inv(1+w2);
```

```
seed=1;
const1=(1/gamma(a1+1));
const2=(1/gamma(a2+1));
scale=1; y=J(n,2,0);
do i=1 to n;
do until(f1/h1 > ranuni(seed) & f2/h2 > ranuni(seed) );
u1=ranuni(seed); u2=ranuni(seed);
if (u1 <= r1) & (u2 <= r2) then z1=-log(u1/r1) ;
if u1 <= r1 & u2 <= r2 then z2=-log(u2/r2) ;
if u1 <= r1 & u2 > r2 then z1=-log(u1/r1) ;
if u1 <= r1 & u2 > r2 then z2=log(ranuni(seed))/l1 ;
if u2 <= r2 & u1 > r1 then z2=-log(u2/r2) ;
if u2 <= r2 & u1 > r1 then z1=log(ranuni(seed))/l2 ;
if u1 > r1 & u2 > r2 then z1=-log(u1/r1) ;
if u1 > r1 & u2 > r2 then z2=-log(u2/r2) ;
f1=const1*exp(-z1-exp(-z1/a1));
f2=const2*exp(-z2-exp(-z2/a2));
if z1>=0 then h1=const1*exp(-z1);
else h1=const1*w1*l1*exp(l1*z1);
if z2>=0 then h2=const2*exp(-z2);
else h2=const2*w2*l2*exp(l2*z2);
if f1/h1 > ranuni(seed) then q1=z1;
if f2/h2 > ranuni(seed) then q2=z2;
end;
y[i,1]=log(scale) - (q1/a1);
y[i,2]=log(scale) - (q2/a2);
end;
return(y);
finish;
n=30;
alpha1=0.001;
alpha2=alpha1;
x=RJgam(n,alpha1,alpha2,1);
print x;
```

The following script has been adapted from the rgmass script given by [15].

```
proc iml;
test=0;
Start RJgam(n,shape1,shape2,scale=1);
a1=shape1;
a2=shape2;
e=2.71828182845905;
l1=(1/a1)-1; l2=(1/a2)-1;
w1=a1/(e*(1-a1));
w2=a2/(e*(1-a2));
r1=inv(1+w1);
r2=inv(1+w2);
seed=1;
const1=(1/gamma(a1+1));
const2=(1/gamma(a2+1));
scale=1; y=J(n,2,0);
do i=1 to n;
do until(f1/h1 > ranuni(seed) & f2/h2 > ranuni(seed) );
u1=ranuni(seed); u2=ranuni(seed);
if (u1 <= r1) & (u2 <= r2) then z1=-log(u1/r1) ;
if u1 <= r1 & u2 <= r2 then z2=-log(u2/r2) ;
if u1 <= r1 & u2 > r2 then z1=-log(u1/r1) ;
if u1 <= r1 & u2 > r2 then z2=log(ranuni(seed))/l1 ;
if u2 <= r2 & u1 > r1 then z2=-log(u2/r2) ;
if u2 <= r2 & u1 > r1 then z1=log(ranuni(seed))/l2 ;
if u1 > r1 & u2 > r2 then z1=-log(u1/r1) ;
if u1 > r1 & u2 > r2 then z2=-log(u2/r2) ;
f1=const1*exp(-z1-exp(-z1/a1));
f2=const2*exp(-z2-exp(-z2/a2));
if z1>=0 then h1=const1*exp(-z1);
else h1=const1*w1*l1*exp(l1*z1);
if z2>=0 then h2=const2*exp(-z2);
else h2=const2*w2*l2*exp(l2*z2);
if f1/h1 > ranuni(seed) then q1=z1;
```

```
if f2/h2 > ranuni(seed) then q2=z2;
end;
y[i,1]=log(scale) - (q1/a1);
y[i,2]=log(scale) - (q2/a2);
end;
return(y);
finish;
n=30;
alpha1=0.001;
alpha2=alpha1;
x=RJgam(n,alpha1,alpha2,1);
print x;
```

6.7 SAS code

Proc univariate and SG plot for transformed alpha:

```
proc iml;

/*Plotting Gamma*/
n=2000;
Data=J(n,5,0);

a1=10;
a2=1;
a3=0.1;
a4=0.01;
seed=1;

do i = 1 to 20 by 0.01;
Data[i,5]=exp(-i);
end;

do i=1 to nrow(Data);
      Data[i,1]=rangam(seed, a1);
      Data[i,2]=rangam(seed, a2);
      Data[i,3]=rangam(seed, a3);
      Data[i,4]=rangam(seed, a4);
end;

names1={"Alpha=10", "Alpha=1", "Alpha=0.1", "Alpha=0.01", "exp"};
mattrib Data colname=names1;
create GamData from Data [colname=names1];
append from data;
close GamData;

/*Plotting Transformed Gamma*/
Data2=J(n,4,0);

do i=1 to nrow(Data2);
      Data2[i,1]=-a1*log(Data[i,1]);
      Data2[i,2]=-a2*log(Data[i,2]);
      Data2[i,3]=-a3*log(Data[i,3]);
      Data2[i,4]=-a4*log(Data[i,4]);
end;

/*print Data2;*/

names2={"TAlpha=10", "TAlpha=1", "TAlpha=0.1", "TAlpha=0.01"};
mattrib Data2 colname=names2;
create TransGamData from Data2 [colname=names2];
append from Data2;
close TransGamData;

Data3=Data||Data2;

names3={"Alpha=10", "Alpha=1", "Alpha=0.1",
"Alpha=0.01","TAlpha=10","TAlpha=1", "TAlpha=0.1","TAlpha=0.01"};
mattrib Data3 colname=names3;
create AllData from Data3 [colname=names3];
append from Data3;
close AllData;
title '.';
proc univariate data = Gamdata plots noprint;
var Alpha_10 Alpha_1 Alpha_0_1 Alpha_0_01;
WHERE Alpha_0_01 between 0 and 1;
```

```
histogram Alpha_10 / gamma(alpha=10 sigma=1)  ;
inset gamma(alpha sigma);
histogram Alpha_1 /gamma(alpha=1 sigma=1);
inset gamma(alpha sigma);
histogram Alpha_0_1 /gamma(alpha=0.1 sigma=1);
inset gamma(alpha sigma);

histogram Alpha_0_01 /endpoints = 0 to 1 by 0.1 gamma(alpha=0.01 sigma=1
theta=est);
inset gamma(alpha sigma);

run;

proc univariate data = TransGamData plots noprint;;
var TAlpha_10 TAlpha_1 TAlpha_0_1 TAlpha_0_01;

histogram TAlpha_10 /  odstitle = "."
exp(sigma=1 theta = est) ;
inset exp(sigma theta);
histogram TAlpha_1 / odstitle = "." exp(sigma=1 theta = est) ;
inset exp(sigma theta);
histogram TAlpha_0_1 / odstitle = "." exp(sigma=1 theta = est) ;
inset exp(sigma theta);
histogram TAlpha_0_01 / odstitle = "." exp(sigma=1 theta = est) ;
inset exp(sigma theta);

run;


proc sgplot data=AllData;
       histogram Alpha_10 / transparency=0.75 fillattrs=(color=red);
       histogram TAlpha_10 / transparency=0.75 fillattrs=(color=blue);
         keylegend / location=outside position=bottom;
       xaxis label="Gamma Curves";
     run;

proc sgplot data=AllData;

         histogram Alpha_1 / transparency=0.75 fillattrs=(color=green);
       histogram TAlpha_1 / transparency=0.75 fillattrs=(color=purple);
         keylegend / location=outside position=bottom;
       xaxis label="Gamma Curves";
       run;

proc sgplot data=AllData;
         histogram Alpha_0_1 / transparency=0.75 fillattrs=(color=orange);
       histogram TAlpha_0_1 / transparency=0.75 fillattrs=(color=brown);
         keylegend / location=outside position=bottom;
       xaxis label="Gamma Curves";
       run;

proc sgplot data=AllData;

         histogram Alpha_0_01/ transparency=0.75 fillattrs=(color=green);
       histogram TAlpha_0_01 / transparency=0.75 fillattrs=(color=red);
         keylegend / location=outside position=bottom;
       xaxis label="Gamma Curves";
     run;
proc template;
define statgraph sgdesign;
dynamic _ALPHA_0_01A;
```

```
begingraph / designwidth=640 designheight=546;
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
      layout overlay / xaxisopts=( label=('x values') linearopts=(
viewmin=0.0 viewmax=1.0));
         histogram _ALPHA_0_01A / name='histogram' legendlabel='Alpha =
0.01' datatransparency=0.31 binaxis=false scale=Percent
fillattrs=(color=CX009966 ) outlineattrs=(pattern=SOLID thickness=2 );
         discretelegend 'histogram' / opaque=false border=true halign=right
valign=top displayclipped=true across=1 order=rowmajor location=inside
titleattrs=(color=CXFFFFFF );
      endlayout;
   endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.ALLDATA template=sgdesign;
dynamic _ALPHA_0_01A="'ALPHA_0_01'n";
run;
```

Ahrens-Dieter Algorithm:

```
proc iml;

minX1=0.01;
maxX1=4;
step=0.01;
Data=J((maxX1/step),3,0);
alpha1=0.9;

i=1;
do x1=minX1 to maxX1 by step;
      Data[i,1]=x1;
      Data[i,2]=(1/gamma(alpha1))*exp(-x1)*x1**(alpha1-1);
      if x1 >= 1 then
      Data[i,3]=(1/gamma(alpha1))*exp(-x1);
      if x1 < 1 then
      Data[i,3]=(1/gamma(alpha1))*(x1**(alpha1-1));
      i=i+1;
end;

cn={"x","g","p1","p2"};
create gam from Data[colname=cn];
append from Data;
close gam;
proc template;
define statgraph sgdesign;
dynamic _X _G _X2 _P1A;
begingraph;
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
      layout overlay / yaxisopts=( label=('Density'));
         seriesplot x=_X y=_G / name='series' legendlabel='Target
distribution' connectorder=xaxis lineattrs=(color=CXCC00CC pattern=SOLID
thickness=3 ) markerattrs=(color=CXC6C3C6 );
         seriesplot x=_X2 y=_P1A / name='series2' legendlabel='Envelope
function' connectorder=xaxis lineattrs=(color=CX424142 pattern=SHORTDASH
thickness=3 );
```

```
         discretelegend 'series' 'series2' / opaque=false border=true
halign=center valign=top displayclipped=true down=1 order=columnmajor
location=inside;
         discretelegend 'series' 'series2' / opaque=false border=true
halign=center valign=top displayclipped=true down=1 order=columnmajor
location=inside;
         discretelegend 'series' 'series2' / opaque=false border=true
halign=center valign=top displayclipped=true down=1 order=columnmajor
location=inside;
        *  discretelegend 'series' / opaque=false border=false halign=right
valign=top displayclipped=true across=1 order=rowmajor location=inside
titleattrs=(color=CX0000FF );
      endlayout;
   endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.GAM template=sgdesign;
dynamic _X="X" _G="G" _X2="X" _P1A="P1";
run;
```

Best Algorithm:

```
proc iml;

minX1=0.01;
maxX1=4;
step=0.01;
Data=J((maxX1/step),3,0);
alpha1=0.9;

i=1;
z=0.07+0.75*(1-alpha1)**0.5;
do x1=minX1 to maxX1 by step;
      Data[i,1]=x1;
      Data[i,2]=exp(-x1)*x1**(alpha1-1);
      if x1 >= z then
      Data[i,3]=exp(-x1)*z**(alpha1-1);
      if x1 <z then
      Data[i,3]=x1**(alpha1-1);
      i=i+1;
end;

cn={"x","g","p1","p2"};
create gam from Data[colname=cn];
append from Data;
close gam;
proc template;
define statgraph Graph;
dynamic _X _G _X2 _P1A;
begingraph;
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
      layout overlay / yaxisopts=( label=('Density'));
         seriesplot x=_X y=_G / name='series' legendlabel='alpha = 0.2'
connectorder=xaxis lineattrs=(color=CX39828C thickness=2 );
         seriesplot x=_X2 y=_P1A / name='series2' legendlabel='First mod
envelope' connectorder=xaxis lineattrs=(color=CX424142 pattern=SHORTDASH
thickness=2 );
```

```
         discretelegend 'series' 'series2' / opaque=false border=true
halign=right valign=top displayclipped=true across=1 order=rowmajor
location=inside;
        * discretelegend 'series' / opaque=false border=true halign=left
valign=top displayclipped=true across=1 order=rowmajor location=inside;
      endlayout;
   endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.GAM template=Graph;
dynamic _X="X" _G="G" _X2="X" _P1A="P1";
run;
```

Kundu-Gupta Algorithm 1:

```
proc iml;

minX1=0.01;
maxX1=4;
step=0.01;
Data=J((maxX1/step),3,0);
alpha1=0.2;

i=1;
z=0.07+0.75*(1-alpha1)**0.5;
do x1=minX1 to maxX1 by step;
      Data[i,1]=x1;
      Data[i,2]=(1/gamma(alpha1))*exp(-x1)*x1**(alpha1-1);
      Data[i,3]=((2**alpha1)/gamma(alpha1+1))*(alpha1/2)*((1-exp(-
x1/2))**(alpha1-1))*exp(-x1/2);

      i=i+1;
end;

cn={"x","g","p1","p2"};
create gam from Data[colname=cn];
append from Data;
close gam;
proc template;
define statgraph Graph;
dynamic _X _G _X2 _P1A;
begingraph;
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
      layout overlay / yaxisopts=( label=('Density'));
         seriesplot x=_X y=_G / name='series' legendlabel='alpha = 0.9'
connectorder=xaxis lineattrs=(color=CXCE5539 thickness=2 );
         seriesplot x=_X2 y=_P1A / name='series2' legendlabel='Generalized
exponential envelope' connectorder=xaxis lineattrs=(color=CX0000FF
pattern=MEDIUMDASH thickness=2 );
         discretelegend 'series' 'series2' / opaque=false border=true
halign=right valign=top displayclipped=true across=1 order=rowmajor
location=inside;
        * discretelegend 'series' / opaque=false border=true halign=left
valign=top displayclipped=true across=1 order=rowmajor location=inside;
      endlayout;
   endlayout;
endgraph;
end;
```

```
run;

proc sgrender data=WORK.GAM template=Graph;
dynamic _X="X" _G="G" _X2="X" _P1A="P1";
run;
```

Kundu-Gupta algorithm 2:

```
proc iml;

minX1=0.01;
maxX1=4;
step=0.01;
Data=J((maxX1/step),3,0);
alpha1=0.2;

i=1;
z=0.07+0.75*(1-alpha1)**0.5;
do x1=minX1 to maxX1 by step;
      Data[i,1]=x1;
      Data[i,2]=(1/gamma(alpha1))*exp(-x1)*x1**(alpha1-1);
      Data[i,3]=((2**alpha1)/gamma(alpha1+1))*(alpha1/2)*((1-exp(-
x1/2))**(alpha1-1))*exp(-x1/2);

      i=i+1;
end;

cn={"x","g","p1","p2"};
create gam from Data[colname=cn];
append from Data;
close gam;
proc template;
define statgraph Graph;
dynamic _X _G _X2 _P1A;
begingraph;
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
      layout overlay / yaxisopts=( label=('Density'));
         seriesplot x=_X y=_G / name='series' legendlabel='alpha = 0.9'
connectorder=xaxis lineattrs=(color=CXCE5539 thickness=2 );
         seriesplot x=_X2 y=_P1A / name='series2' legendlabel='Generalized
exponential envelope' connectorder=xaxis lineattrs=(color=CX0000FF
pattern=MEDIUMDASH thickness=2 );
         discretelegend 'series' 'series2' / opaque=false border=true
halign=right valign=top displayclipped=true across=1 order=rowmajor
location=inside;
        * discretelegend 'series' / opaque=false border=true halign=left
valign=top displayclipped=true across=1 order=rowmajor location=inside;
      endlayout;
   endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.GAM template=Graph;
dynamic _X="X" _G="G" _X2="X" _P1A="P1";
run;
```

Kundu-Gupta algorithm 3:

```
proc iml;

minX1=0.01;
maxX1=4;
step=0.01;
Data=J((maxX1/step),3,0);
alpha1=0.9;

i=1;
z=1.0334-0.0766*exp(2.2942*alpha1);
do x1=minX1 to maxX1 by step;
      Data[i,1]=x1;
      Data[i,2]=(1/gamma(alpha1))*exp(-x1)*x1**(alpha1-1);
      if x1 < z then
      Data[i,3]=((2**alpha1)/gamma(alpha1+1))*(alpha1/2)*((1-exp(-
x1/2))**(alpha1-1))*exp(-x1/2);
      else;
      Data[i,3]=exp(-x1)/gamma(alpha1);
      i=i+1;
end;

cn={"x","g","p1","p2"};
create gam from Data[colname=cn];
append from Data;
close gam;
proc template;
define statgraph Graph;
dynamic _X _G _X2 _P1A;
begingraph;
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
      layout overlay / yaxisopts=( label=('Density'));
         seriesplot x=_X y=_G / name='series' legendlabel='alpha = 0.2'
connectorder=xaxis lineattrs=(color=CXCE5539 thickness=2 );
         seriesplot x=_X2 y=_P1A / name='series2' legendlabel='Non singular
GE envelope' connectorder=xaxis lineattrs=(color=CX6371AD pattern=SHORTDASH
thickness=2 );
         discretelegend 'series' 'series2' / opaque=false border=true
halign=right valign=top displayclipped=true across=1 order=rowmajor
location=inside;
         discretelegend 'series' / opaque=false border=true halign=left
valign=top displayclipped=true across=1 order=rowmajor location=inside;
      endlayout;
   endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.GAM template=Graph;
dynamic _X="X" _G="G" _X2="X" _P1A="P1";
run;
```

Ryan Martin algorithm in SAS (SAS replication of rgmass):

```
proc iml;
test=0;

Start RJgam(n,shape,scale=1);
```

```
a=shape;
e=2.71828182845905;
l=(1/a)-1;
w=a/(e*(1-a));
r=inv(1+w);
seed=1;
const=(1/gamma(a+1));
scale=1;
y=J(n,1,0);

do i=1 to n;
      do until(f/h > ranuni(seed));
            u=ranuni(seed);
                  if u <= r then z=-log(u/r); else z=log(ranuni(seed))/l;
                        f=const*exp(-z-exp(-z/a));
                              if z>=0 then h=const*exp(-z); else
h=const*w*l*exp(l*z);
                                    if f/h > ranuni(seed) then q=z;
 end;
y[i,1]=log(scale) - (q/a);
end;

return(y);
finish;

n=30;
alpha=0.0001;
x=RJgam(n,alpha,1);

k=J(n,1,0);
      do i=1 to n;
            ran=rangam(1,alpha);
                  if ran <= test then
                  k[i]= .; else k[i]= log(ran);
      end;

l=J(n,1,0);
call randgen(l, 'GAMMA', alpha);

      do i=1 to n;
            ran=rangam(1,alpha);
                  if l[i] <= test then
                  l[i]= .; else l[i]= log(l[i]);
      end;

d=x||k||l;

cn={"RJgamma" "rangam" "randgen"};

create Data from d[colname=cn];
append from d;
close;

print x;
```

Ryan Martin envelope and target density:

```
proc iml;

D1=J((7/0.01),3,0);
alpha1=0.2;
```

```
l=(1/alpha1)-1;
w=alpha1/(exp(1)*(1-alpha1));

i=1;
do z=-1 to 6 by 0.01;
      D1[i,1]=z;
      D1[i,2]=exp(-z-exp(-z/alpha1));
            if z < 0 then D1[i,3]=w*l*exp(l*z);else
            D1[i,3]=exp(-z);
      i=i+1;
end;

print D1;
cn={"z","d1","d2"};
create gam from D1[colname=cn];
append from D1;
close gam;

proc template;
define statgraph Graph;
dynamic _Z _D1A _Z2 _D2A;
begingraph;
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
      layout overlay / yaxisopts=( label=('Density'));
         seriesplot x=_Z y=_D1A / name='series' legendlabel='fz(z)'
connectorder=xaxis lineattrs=(color=CX424142 thickness=2 );
         seriesplot x=_Z2 y=_D2A / name='series2' legendlabel='Envelope
function' connectorder=xaxis lineattrs=(color=CX009999 pattern=SHORTDASH
thickness=2 );
         discretelegend 'series' 'series2' / opaque=false border=true
halign=center valign=top displayclipped=true down=1 order=columnmajor
location=inside;
        * entry halign=center 'Alpha = 0.01 ' / valign=top
location=outside;
      endlayout;
   endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.GAM template=Graph;
dynamic _Z="Z" _D1A="D1" _Z2="Z" _D2A="D2";
run;
```

3D univariate theoretical gamma pdf for decreasing alpha:

```
proc iml;

minA=0.01;
maxA=0.99;
step=minA;
xl=minA;
xr=4;
xstep=step;

DataT=J(maxA/step,xr/xstep,0);
```

```
k=1;
do i=minA to maxA by step;
l=1;
      do j=xl to xr by xstep;
            DataT[k,l]=(1/gamma(i))*(j**(i-1))*exp(-j);
            l=l+1;
      end;
      k=k+1;
end;

DataT2=J(nrow(DataT)*ncol(DataT),3,0);

k=1;
do i=1 to nrow(DataT2);
      do j=1 to ncol(DataT);
            DataT2[k,1]=minA+(i-1)*step;
            DataT2[k,2]=xl+(j-1)*xstep;
            DataT2[k,3]=DataT[i,j];
      k=k+1;
      end;
end;

names={"Alpha" "X" "Density"};
create Gam3D_Data from DataT2[colname=names];
append from DataT2;
close;

goptions reset=all border;
proc g3grid data=Gam3D_Data out=Gam3DG;
   grid Alpha*X=Density /naxis1=200 naxis2=200;
run;

goptions reset=all;
ods html image_dpi=300;
ods graphics / ANTIALIAS=on ANTIALIASMAX=10000 SUBPIXEL=on;
run;
proc template;
  define statgraph surfaceplotparm;
    begingraph;
      layout overlay3d/
            rotate=50 tilt=15 cube=false;
            surfaceplotparm x=X y=Alpha z= Density /surfacetype=fillgrid
                        name="surface"
          reversecolormodel=true
              surfacetype=fill
          SURFACECOLORGRADIENT=Density
          colormodel=(purple cyan grey );;
              continuouslegend "surface" / title='Density';
      endlayout;
      endgraph;
  end;
proc sgrender data=Gam3DG template=surfaceplotparm;
run;
```

3D univariate empirical gamma pdf for decreasing alpha:

```
proc iml;

minA=0.01;
maxA=1;
step=0.01;
```

```
n=400;
seed=0;

Data1=J(maxA/step,n,0);

do i=1 to n;
      c=1;
      do j=minA to maxA by step;
            Data1[c,i]=rangam(seed,j);
            c=c+1;
      end;
end;



max=max(Data1);
min=min(Data1);

/*create bins*/

interval=max-min;
numBins=150;
intW=interval/numBins;
Bins=J(maxA/step,numBins,0);


do i = 1 to nrow(Data1);
      do j = 1 to ncol(Data1);
            do d = 1 to numBins;
                  if (min + (d-1)*intW) <= Data1[i,j] && Data1[i,j] <= (min
+ d*intW) then Bins[i,d]=Bins[i,d]+1;
            end;
      end;
end;

alphaM=j(maxA/step,1,0);
do i = 1 to nrow(alphaM);
      alphaM[i]=minA+(i-1)*step;
end;

DataAB=alphaM||Bins;

/*Create data matrix*/

DataF=J(nrow(DataAB)*ncol(Bins),3,0);

k=1;
do i = 1 to nrow(DataF);
      do j = 1 to ncol(Bins);
            DataF[k,1]=DataAB[i,1];
            DataF[k,2]=j;
            Dataf[k,3]=Bins[i,j];
            k=k+1;
      end;
end;


names={"Alpha" "Bin" "Frequency"};
create Gam3D_Data from DataF[colname=names];
append from DataF;
close;
```

```
proc g3grid data=Gam3D_Data out=Gam3DG;
   grid Alpha*Bin=Frequency /naxis1=200 naxis2=200;
run;

goptions reset=all;
ods html image_dpi=300;
ods graphics / ANTIALIAS=on ANTIALIASMAX=10000 SUBPIXEL=on;
run;
proc template;
  define statgraph surfaceplotparm;
    begingraph;
      layout overlay3d/
                  xaxisopts=(label="    Bin Range")
            yaxisopts=(label="Alpha")
            rotate=90 tilt=89 cube=false;
            surfaceplotparm x=Bin y=Alpha z= Frequency
/surfacetype=fillgrid
                          name="surface"
          reversecolormodel=true
              surfacetype=fill
          SURFACECOLORGRADIENT=Frequency
          colormodel=(green yellow red);
              continuouslegend "surface" / title='Frequency';
      endlayout;
      endgraph;
  end;
proc sgrender data=Gam3DG template=surfaceplotparm;
run;
```

Bivariate gamma case 1:

```
proc iml;

minA=0.01;
maxA=1;
step=0.01;
n=400;
seed=0;

Data1=J(ceil(maxA/step),n,0);

do i=1 to n;
      c=1;
      do j=minA to maxA by step;
            Data1[c,i]=sqrt((rangam(seed,j))**2+(rangam(seed,0.5))**2);
            c=c+1;
      end;
end;

max=max(Data1);
min=min(Data1);

/*create bins*/

interval=max-min;
numBins=200;
intW=interval/numBins;
Bins=J(maxA/step,numBins,0);
```

```
do i = 1 to nrow(Data1);
      do j = 1 to ncol(Data1);
            do d = 1 to numBins;
                  if (min + (d-1)*intW) <= Data1[i,j] && Data1[i,j] <= (min
+ d*intW) then Bins[i,d]=Bins[i,d]+1;
            end;
      end;
end;

alphaM=j(maxA/step,1,0);
do i = 1 to nrow(alphaM);
      alphaM[i]=minA+(i-1)*step;
end;

DataAB=alphaM||Bins;

/*Create data matrix*/

DataF=J(nrow(DataAB)*ncol(Bins),3,0);

k=1;
do i = 1 to nrow(DataF);
      do j = 1 to ncol(Bins);
            DataF[k,1]=DataAB[i,1];
            DataF[k,2]=j;
            Dataf[k,3]=Bins[i,j];
            k=k+1;
      end;
end;


names={"Alpha" "Bin" "Frequency"};
create Gam3D_Data from DataF[colname=names];
append from DataF;
close;

goptions reset=all;
proc g3grid data=Gam3D_Data out=Gam3DG;
   grid Alpha*Bin=Frequency /naxis1=250 naxis2=250;
run;
goptions reset=all;
ods html image_dpi=300;
ods graphics / ANTIALIAS=on ANTIALIASMAX=10000;
run;
proc template;
  define statgraph surfaceplotparm;
    begingraph;
      layout overlay3d/
            xaxisopts=(label=".")
            yaxisopts=(label=".")
            zaxisopts=(label="Frequency")
            rotate=90 tilt=88 cube=false;
            surfaceplotparm x=Bin y=Alpha z= Frequency /
surfacetype=fillgrid
            name="surface"
          reversecolormodel=true
              surfacetype=fill
          SURFACECOLORGRADIENT=Frequency
           colormodel=(VIBG VIYG grey);
              continuouslegend "surface" / title='Frequency';
```

```
      endlayout;
      endgraph;
  end;
proc sgrender data=Gam3DG template=surfaceplotparm;
run;
```

Bivariate gamma case 2:

```
proc iml;

minA=0.01;
maxA=1;
step=minA;
n=400;
seed=0;

Data1=J(ceil(maxA/step),n,0);

do i=1 to n;
      c=1;
      do j=minA to maxA by step;
            Data1[c,i]=sqrt((rangam(seed,j))**2+(rangam(seed,j))**2);
            c=c+1;
      end;
end;

max=max(Data1);
min=min(Data1);

/*create bins*/

interval=max-min;
numBins=300;
intW=interval/numBins;
Bins=J(maxA/step,numBins,0);


do i = 1 to nrow(Data1);
      do j = 1 to ncol(Data1);
            do d = 1 to numBins;
                  if (min + (d-1)*intW) <= Data1[i,j] && Data1[i,j] <= (min
+ d*intW) then Bins[i,d]=Bins[i,d]+1;
            end;
      end;
end;

alphaM=j(maxA/step,1,0);
do i = 1 to nrow(alphaM);
      alphaM[i]=minA+(i-1)*step;
end;

DataAB=alphaM||Bins;

/*Create data matrix*/

DataF=J(nrow(DataAB)*ncol(Bins),3,0);

k=1;
do i = 1 to nrow(DataF);
```

```
if k<=40000 then do;
      do j = 1 to ncol(Bins);
            DataF[k,1]=DataAB[i,1];
            DataF[k,2]=j;
            DataF[k,3]=Bins[i,j];
            k=k+1;
      end;
end;
end;


names={"Alpha1Dec_Alpha2Dec" "Bin" "Frequency"};
create Gam3D_Data from DataF[colname=names];
append from DataF;
close;


proc g3grid data=Gam3D_Data out=Gam3DG;
   grid Alpha1Dec_Alpha2Dec*Bin=Frequency /naxis1=200 naxis2=200;
run;
goptions reset=all;
ods html image_dpi=300;
ods graphics / ANTIALIAS=on ANTIALIASMAX=10000;
run;
proc template;
  define statgraph surfaceplotparm;
    begingraph;
      layout overlay3d/
            xaxisopts=(label=".")
            yaxisopts=(label=".")
            zaxisopts=(label="Frequency")
            rotate=90 tilt=88 cube=false;
            surfaceplotparm x=Bin y=Alpha1Dec_Alpha2Dec z= Frequency
/surfacetype=fillgrid
            name="surface"
          reversecolormodel=true
              surfacetype=fill
          SURFACECOLORGRADIENT=Frequency
           colormodel=(red orange grey);
              continuouslegend "surface" / title='Frequency';
      endlayout;
      endgraph;
  end;
proc sgrender data=Gam3DG template=surfaceplotparm;
run;
```

Bivariate gamma case 3:

```
proc iml;

minA=0.01;
maxA=1;
step=0.01;
n=400;
seed=0;

Data1=J(ceil(maxA/step),n,0);

do i=1 to n;
      c=1;
```

```
      k=maxA;
      do j=minA to maxA by step;
            Data1[c,i]=sqrt((rangam(seed,j))**2+(rangam(seed,k))**2);
            c=c+1;
      k=k-step;
      end;
end;

max=max(Data1);
min=min(Data1);

/*create bins*/

interval=max-min;
numBins=50;
intW=interval/numBins;
Bins=J(maxA/step,numBins,0);


do i = 1 to nrow(Data1);
      do j = 1 to ncol(Data1);
            do d = 1 to numBins;
                  if (min + (d-1)*intW) <= Data1[i,j] && Data1[i,j] <= (min
+ d*intW) then Bins[i,d]=Bins[i,d]+1;
            end;
      end;
end;

alphaM=j(maxA/step,1,0);
do i = 1 to nrow(alphaM);
      alphaM[i]=minA+(i-1)*step;
end;

DataAB=alphaM||Bins;

/*Create data matrix*/

DataF=J(nrow(DataAB)*ncol(Bins),3,0);

k=1;
do i = 1 to nrow(DataF);
      do j = 1 to ncol(Bins);
            DataF[k,1]=DataAB[i,1];
            DataF[k,2]=j;
            DataF[k,3]=Bins[i,j];
            k=k+1;
      end;
end;


names={"Alpha1Dec_Alpha2Inc" "Bin" "Frequency"};
create Gam3D_Data from DataF[colname=names];
append from DataF;
close;

proc g3grid data=Gam3D_Data out=Gam3DG;
   grid Alpha1Dec_Alpha2Inc*Bin=Frequency /naxis1=150 naxis2=150
                                                        spline
                                                 smooth=0.001
;
run;
```

```
goptions reset=all;
ods html image_dpi=100;
proc g3d data=Gam3DG;
plot Alpha1Dec_Alpha2Inc*Bin=Frequency/
                  rotate = -30 to 270 by 30
                  ctop=VIYG  cbottom=BIPB
                  zmin=0;
run;

proc template;
define statgraph sgdesign;
dynamic _BIN _ALPHA1DEC_ALPHA2INC _FREQUENCY;
begingraph;
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
      layout overlay / yaxisopts=( label=('Alpha1 dec and alpha2 inc'));
         contourplotparm x=_BIN y=_ALPHA1DEC_ALPHA2INC z=_FREQUENCY /
name='contour' contourtype=LINEFILL colormodel=ThreeColorRamp gridded=false
lineattrs=(color=CXA4A5A4 );
         continuouslegend 'contour' / halign=right valign=center
title='Frequency' location=outside;
      endlayout;
   endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.GAM3D_DATA template=sgdesign;
dynamic _BIN="BIN" _ALPHA1DEC_ALPHA2INC="'ALPHA1DEC_ALPHA2INC'n"
_FREQUENCY="FREQUENCY";
run;
```

Bivariate gamma case 4:

```
proc iml;

minA=0.01;
maxA=1;
step=0.01;
n=400;
seed=0;

Data1=J(ceil(maxA/step),n,0);

do i=1 to n;
      c=1;
      do j=minA to maxA by step;
            Data1[c,i]=sqrt((rangam(seed,j))**2+(rangam(seed,1))**2);
            c=c+1;
      end;
end;

max=max(Data1);
min=min(Data1);

/*create bins*/

interval=max-min;
```

```
numBins=50;
intW=interval/numBins;
Bins=J(maxA/step,numBins,0);


do i = 1 to nrow(Data1);
      do j = 1 to ncol(Data1);
            do d = 1 to numBins;
                  if (min + (d-1)*intW) <= Data1[i,j] && Data1[i,j] <= (min
+ d*intW) then Bins[i,d]=Bins[i,d]+1;
            end;
      end;
end;

alphaM=j(maxA/step,1,0);
do i = 1 to nrow(alphaM);
      alphaM[i]=minA+(i-1)*step;
end;

DataAB=alphaM||Bins;

/*Create data matrix*/

DataF=J(nrow(DataAB)*ncol(Bins),3,0);

k=1;
do i = 1 to nrow(DataF);
      do j = 1 to ncol(Bins);
            DataF[k,1]=DataAB[i,1];
            DataF[k,2]=j;
            DataF[k,3]=Bins[i,j];
            k=k+1;
      end;
end;


names={"Alpha1Dec_Alpha2grt1" "Bin" "Frequency"};
create Gam3D_Data from DataF[colname=names];
append from DataF;
close;

proc g3grid data=Gam3D_Data out=Gam3DG;
   grid Alpha1Dec_Alpha2grt1*Bin=Frequency /naxis1=150 naxis2=150
                                                              spline
                                                  smooth=0.001;
run;


goptions reset=all border cback=white htitle=12pt;
ods html image_dpi=100;
title1 'Alpha1=5';
proc g3d data=Gam3DG;
plot Alpha1Dec_Alpha2grt1*Bin=Frequency/
rotate = -30 to 270 by 30
                  ctop=BIGB  cbottom=MOPPK
                  zmin=0;
run;

proc template;
define statgraph sgdesign;
dynamic _BIN _ALPHA1DEC_ALPHA2GRT1A _FREQUENCY;
```

```
begingraph;
   entrytitle halign=center 'Alpha2 = 1';
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
      layout overlay / yaxisopts=( label=('Alpha1 dec and Alpha2 greater
than 1'));
         contourplotparm x=_BIN y=_ALPHA1DEC_ALPHA2GRT1A z=_FREQUENCY /
name='contour' contourtype=GRADIENT colormodel=ThreeColorRamp
gridded=false;
         continuouslegend 'contour' / halign=right valign=center
title='Frequency' location=outside;
      endlayout;
   endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.GAM3D_DATA template=sgdesign;
dynamic _BIN="BIN" _ALPHA1DEC_ALPHA2GRT1A="'ALPHA1DEC_ALPHA2GRT1'n"
_FREQUENCY="FREQUENCY";
run;
```

Empirical bivariate gamma for both alpha decreasing simultaneously:

```
proc iml;

alpha1=0.01;
alpha2=0.01;
minX1=0.01;
maxX1=1;
step=0.01;
seed=1;
minX2=minX1;
maxX2=maxX1;
Data3=J((maxX1/step)**2,2,0);
Data31=J((maxX1/step)**2,2,0);

i=1;
j=1;
n=(maxX1/step)**2;
do i=1 to n;
      Data3[i,1]=(rangam(seed,alpha1));
      if Data3[i,1]=0 then Data3[i,1]=(rangam(seed,alpha1));
      Data3[i,2]=(rangam(seed,alpha2));
      if Data3[i,2]=0 then Data3[i,2]=(rangam(seed,alpha1));
end;

do i=1 to nrow(Data3);
      Data31[i,1]=-alpha1*log(Data3[i,1]);
      Data31[i,2]=-alpha2*log(Data3[i,2]);
end;


/*create bins*/

numBins=100;
Bins=J(numBins,2,0);

b1=bin(Data31[,1],numBins);
```

```
b2=bin(Data31[,2],numBins);

do i=1 to nrow(b1);
      do j=1 to nrow(b1);
            if b1[j]=i then Bins[i,1]=Bins[i,1]+1;
            if b2[j]=i then Bins[i,2]=Bins[i,2]+1;
      end;
end;

Data4=J(numBins,3,0);

do i=1 to nrow(Bins);
      Data4[i,1]=i;
end;
Data4[,2]=Data4[,1];
Data4[,3]=Bins[,1]+Bins[,2];

names3={"Bin1" "Bin2" "Frequency"};
create Gam3D_Data2 from Data4[colname=names3];
append from Data4;
close;

proc g3grid data=Gam3D_Data2 out=Gam3DG2;
   grid Bin1*Bin2=Frequency /naxis1=200 naxis2=200;
run;

goptions reset=all;
ods html image_dpi=300;
ods graphics / ANTIALIAS=on;
run;
proc template;
  define statgraph surfaceplotparm2;
    begingraph;
      entrytitle "Emperical Transformed Bivariate Gamma a1=0.01 a2=0.01";
      layout overlay3d/
                  xaxisopts=(label="Bin1")
            yaxisopts=(label="Bin2")
            zaxisopts=(label="Frequency")
            rotate=170 tilt=15 cube=false;
            surfaceplotparm x=Bin1 y=Bin2 z= Frequency
/surfacetype=fillgrid
             name="surface"
          reversecolormodel=true
              surfacetype=fill
          SURFACECOLORGRADIENT=Frequency
           colormodel=(VLIV MOPR DAGRR);
              continuouslegend "surface" / title='Frequency';
      endlayout;
      endgraph;
  end;
proc sgrender data=Gam3DG2 template=surfaceplotparm2;
run;
```

Bivariate target and envelope densities:

```
proc iml;


Data1=J((6.5/0.01)**2,4,0);
```

```
alpha1=0.01;
alpha2=0.01;

j=1;
l1=(1/alpha1)-1;
l2=(1/alpha2)-1;
w1=alpha1/(exp(1)*(1-alpha1));
w2=alpha2/(exp(1)*(1-alpha2));
do z1=-0.5 to 5 by 0.01;
       do z2=-0.5 to 5 by 0.01;
                  Data1[j,1]=z1;
                  Data1[j,2]=z2;
                  Data1[j,3]=exp(-z1-exp(-z1/alpha1))*exp(-z2-exp(-
z2/alpha2));
                  Data1[j,4]=exp(-z1)*exp(-z2);
                  if z1 < 0 then Data1[j,4] = 0;
                  if z2 < 0 then Data1[j,4] = 0;
                   j=j+1;
       end;
end;


names1={"z1" "z2" "tar" "Density"};
create Gam3D_Data from Data1[colname=names1];
append from Data1;
close;

proc g3grid data=Gam3D_Data out=Gam3DG;
   grid z1*z2=tar /naxis1=300 naxis2=300;
run;
proc g3d data=Gam3DG;
plot z1*z2=tar/
CAXIS=white
CTEXT=white
rotate = 300
ctop=BIYG  cbottom=BIPB name="plot7" ;
run;
proc g3grid data=Gam3D_Data out=Gam3DG;
   grid z1*z2=Density /naxis1=70 naxis2=70;
run;
proc g3d data=Gam3DG;
plot z1*z2=Density/
rotate = 300
ctop=VIP cbottom=BIOY name="plot8";
   note;
   note j=r  h=1.5  c=BIYG "Target pdf "
        j=r  h=1.5 c=VIP "Envelope function " ;

run;
proc greplay tc=tempcat nofs igout=work.gseg;
tdef WHOLE des="my template"
        1/llx=0   lly=0
          ulx=0   uly=100
          urx=100 ury=100
          lrx=100  lry=0
          ;
template = whole;
treplay 1:plot7 1:plot8;
run;

proc g3grid data=Gam3D_Data out=Gam3DG;
```

```
   grid z1*z2=Density /naxis1=200 naxis2=200;
run;

goptions reset=all;
ods html image_dpi=300;
ods graphics / ANTIALIAS=on ANTIALIASMAX=10000 SUBPIXEL=on;
run;
proc template;
  define statgraph surfaceplotparm;
    begingraph;
      layout overlay3d/
                  xaxisopts=(label="Z1")
            yaxisopts=(label="Z2")
            zaxisopts=(label="Density")
            rotate=150 tilt=15 cube=false;
            surfaceplotparm x=z1 y=z2 z= Density /surfacetype=fillgrid
                        name="surface"
          reversecolormodel=true
              surfacetype=fill
          SURFACECOLORGRADIENT=Density
          colormodel=(STP LIP VLIP);
              continuouslegend "surface" / title='Density';
      endlayout;
      endgraph;
  end;
proc sgrender data=Gam3DG template=surfaceplotparm;
run;
```

Bivariate target density IML studio:

```
minX1=0.00001;
maxX1=4;
step=0.01;
minX2=minX1;
maxX2=maxX1;

Data1=J((maxX1/step)**2,3,0);
alpha1=0.2;
alpha2=0.2;

i=1;
j=1;
l1=(1/alpha1)-1;
l2=(1/alpha2)-1;
w1=alpha1/(exp(1)*(1-alpha1));
w2=alpha2/(exp(1)*(1-alpha2));
do x1=minX1 to maxX1 by step;
       do x2=minX2 to maxX2 by step;
            z1=-alpha1*log(x1);
            z2=-alpha2*log(x2);
                  Data1[j,1]=z1;
                  Data1[j,2]=z2;
                  Data1[j,3]=exp(-z1-exp(-z1/alpha1))*exp(-z2-exp(-
z2/alpha2));
                   j=j+1;
            end;
      i=i+1;
end;
print Data1;
RotatingPlot.Create("plot", Data1[,1],Data1[,2],Data1[,3]);
```

Ryan Martin rgmass algorithm adapted to bivariate case for simultaneous simulation:

```
proc iml;
test=0;

Start RJgam(n,shape1,shape2,scale=1);
a1=shape1;
a2=shape2;
e=2.71828182845905;
l1=(1/a1)-1;
l2=(1/a2)-1;
w1=a1/(e*(1-a1));
w2=a2/(e*(1-a2));
r1=inv(1+w1);
r2=inv(1+w2);
seed=1;
const1=(1/gamma(a1+1));
const2=(1/gamma(a2+1));
scale=1;
y=J(n,2,0);

do i=1 to n;
      do until(f1/h1 > ranuni(seed) & f2/h2 > ranuni(seed) );
            u1=ranuni(seed);
            u2=ranuni(seed);
                  if (u1 <= r1) & (u2 <= r2) then z1=-log(u1/r1) ;
                  if u1 <= r1 & u2 <= r2  then z2=-log(u2/r2) ;
                  if u1 <= r1 & u2 > r2  then z1=-log(u1/r1) ;
                  if u1 <= r1 & u2 > r2  then z2=log(ranuni(seed))/l1 ;
                  if u2 <= r2 & u1 > r1  then z2=-log(u2/r2) ;
                  if u2 <= r2 & u1 > r1  then z1=log(ranuni(seed))/l2 ;
                  if u1 > r1 & u2 > r2  then z1=-log(u1/r1) ;
                  if u1 > r1 & u2 > r2  then z2=-log(u2/r2) ;


                        f1=const1*exp(-z1-exp(-z1/a1));
                        f2=const2*exp(-z2-exp(-z2/a2));
                              if z1>=0 then h1=const1*exp(-z1); else
h1=const1*w1*l1*exp(l1*z1);
                              if z2>=0 then h2=const2*exp(-z2); else
h2=const2*w2*l2*exp(l2*z2);
                              if f1/h1 > ranuni(seed) then q1=z1;
                              if f2/h2 > ranuni(seed) then q2=z2;
 end;
y[i,1]=log(scale) - (q1/a1);
y[i,2]=log(scale) - (q2/a2);
end;

return(y);
finish;

n=30;
alpha1=0.001;
alpha2=alpha1;
x=RJgam(n,alpha1,alpha2,1);

print x;
```

Comparing acceptance rates in the univariate case:

```
proc iml;
```

```
/* acceptance rates*/

/*Liu.et.al*/

minA=0.01;
maxA=0.5;
step=0.02;
L=J(maxA/step,2,0);
AD=J(maxA/step,1,0);
Best=J(maxA/step,1,0);
KG=J(maxA/step,1,0);
e=2.71828182845905;

k=1;
do i=minA to maxA by step;
zb=0.07+0.75*(1-i)**(0.5);
d=1.0334-0.0766*exp(2.2942*i);
a=(2**i)*(1-exp(-d/2))**i;
b=i*d**(i-1);
c=a+b;
      L[k,1]=(1+(i)/(e*(1-i)))**(-1);
      AD[k,1]=gamma(i+1)*e/(i+e);
      Best[k,1]=(gamma(i+1)*exp(zb))/((zb**i)*(exp(zb)+i/zb));
      KG[k,1]=((1/gamma(i+1))*((2**i)*(1-exp(-d/2))**i + (i*d**(i-1))*exp(-
d)))**-1;
      L[k,2]=i;
      k=k+1;
end;

Data=AD||Best||KG||L;

names={"AD" "Best" "KG" "Ryan" "Alpha"};
create AccData from Data[colname=names];
append from Data;
close;

proc template;
define statgraph Graph;
dynamic _ALPHA _RYAN _ALPHA2 _BEST _ALPHA3 _AD _ALPHA4 _KG;
begingraph;
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
      layout overlay / yaxisopts=( label=('Acceptance rate'));
         seriesplot x=_ALPHA y=_RYAN / name='series' connectorder=xaxis
lineattrs=(color=CXFF0000 thickness=2 );
         seriesplot x=_ALPHA2 y=_BEST / name='series2' connectorder=xaxis
lineattrs=(color=CX00FF00 pattern=SHORTDASH thickness=2 );
         seriesplot x=_ALPHA3 y=_AD / name='series3' connectorder=xaxis
lineattrs=(color=CX0000FF pattern=LONGDASH thickness=2 );
         seriesplot x=_ALPHA4 y=_KG / name='series4' connectorder=xaxis
lineattrs=(color=CX990099 pattern=DOT thickness=2 );
         discretelegend 'series' 'series2' 'series3' 'series4' /
opaque=false border=true halign=center valign=top displayclipped=true
down=1 order=columnmajor location=inside;
      endlayout;
   endlayout;
endgraph;
end;
run;
```

```
proc sgrender data=WORK.ACCDATA template=Graph;
dynamic _ALPHA="ALPHA" _RYAN="RYAN" _ALPHA2="ALPHA" _BEST="BEST"
_ALPHA3="ALPHA" _AD="AD" _ALPHA4="ALPHA" _KG="KG";
run;
```

Comparing the efficiency of the adapted bivariate algorithm in terms of simultaneous and individual component simulation:

```
proc iml;

a=0.001;
e=2.71828182845905;
l=(1/a)-1;
w=a/(e*(1-a));
r=inv(1+w);
seed=1;
const=(1/gamma(a+1));
scale=1;
n= 20;
y=J(n,1,0);

k=500;
num=J(k,1,0);
avg=J(0.1/0.001,3,0);

idx=1;
do a=0.001 to 0.1 by 0.001;
      do j=1 to k;
            count =0;
            do i=1 to n;
                  ch=ranuni(seed);
                  do until(f/h > ch);
                        u=ranuni(seed);
                              if u <= r then z=-log(u/r); else
z=log(ranuni(seed))/l;
                                    f=const*exp(-z-exp(-z/a));
                                          if z>=0 then h=const*exp(-z);
else h=const*w*l*exp(l*z);
                                                if f/h > ch then q=z;
                                                count = count +1;
             end;
            y[i,1]=log(scale) - (q/a);
            end;
      num[j]=count;


      end;
av1=num[:];
avg[idx,1]=a;
avg[idx,2]=av1;
idx=idx+1;
end;

a1=0.001;
a2=a1;
e=2.71828182845905;
l1=(1/a1)-1;
l2=(1/a2)-1;
w1=a1/(e*(1-a1));
```

```
w2=a2/(e*(1-a2));
r1=inv(1+w1);
r2=inv(1+w2);
seed=1;
const1=(1/gamma(a1+1));
const2=(1/gamma(a2+1));
scale=1;
n=10;
y=J(n,2,0);
num1=J(k,1,0);
idx=1;

do a1 = 0.001 to 0.1 by 0.001;
a2=a1;
do j = 1 to k;
count = 0;

      do i=1 to n;
      ch1=ranuni(seed);
      ch2=ranuni(seed);
            do until(f1/h1 > ch1 & f2/h2 > ch2 );
                  u1=ranuni(seed);
                  u2=ranuni(seed);
                        if u1 <= r1 & u2 <= r2 then z1=-log(u1/r1) ;
                        if u1 <= r1 & u2 <= r2  then z2=-log(u2/r2) ;
                        if u1 <= r1 & u2 > r2  then z1=-log(u1/r1) ;
                        if u1 <= r1 & u2 > r2  then z2=log(ranuni(seed))/l1
;
                        if u2 <= r2 & u1 > r1  then z2=-log(u2/r2) ;
                        if u2 <= r2 & u1 > r1  then z1=log(ranuni(seed))/l2
;
                        if u1 > r1 & u2 > r2  then z1=-log(u1/r1) ;
                        if u1 > r1 & u2 > r2  then z2=-log(u2/r2) ;


                        f1=const1*exp(-z1-exp(-z1/a1));
                        f2=const2*exp(-z2-exp(-z2/a2));
                              if z1>=0 then h1=const1*exp(-z1); else
h1=const1*w1*l1*exp(l1*z1);
                              if z2>=0 then h2=const2*exp(-z2); else
h2=const2*w2*l2*exp(l2*z2);
                              if f1/h1 > ch1 then q1=z1;
                              if f2/h2 > ch2 then q2=z2;
                              count = count +1;
      end;
      y[i,1]=log(scale) - (q1/a1);
      y[i,2]=log(scale) - (q2/a2);
      end;
num1[j]=count;

end;
av2=num1[:];
avg[idx,3]=av2;
idx=idx+1;
end;



cn={"alpha" "N1" "N2"};
create res from avg[colname=cn];
```

```
append from avg;
close;

proc template;
define statgraph Graph;
dynamic _ALPHA _N1A _ALPHA2 _N2A;
begingraph;
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
      layout overlay / xaxisopts=( label=('alpha (bivariate case a1=a2)'))
yaxisopts=( label=('Number of iterations needed for vector of 10
components'));
         seriesplot x=_ALPHA y=_N1A / name='series' legendlabel='Simulating
components individually' connectorder=xaxis lineattrs=(color=CX8CA6CE
thickness=2 );
         seriesplot x=_ALPHA2 y=_N2A / name='series2'
legendlabel='Simulating components simultaneously' connectorder=xaxis
lineattrs=(color=CXFF8273 thickness=2 );
         discretelegend 'series' 'series2' / opaque=false border=true
halign=center valign=center displayclipped=true down=1 order=columnmajor
location=inside;
      endlayout;
   endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.RES template=Graph;
dynamic _ALPHA="ALPHA" _N1A="N1" _ALPHA2="ALPHA" _N2A="N2";
run;
```

Proportion of zeros generated in univariate case using SAS generators:

```
proc iml;

n=30;

d=J(0.01/0.0001,3,0);
c=0;
c1=0;
j=1;
test=0;

do alpha=0.0001 to 0.01 by 0.0001;

      k=J(n,1,0);
                  do i=1 to n;
                        ran=rangam(1,alpha);
                              if ran <= test then
                              k[i]= .; else k[i]= log(ran);
                  end;

      l=J(n,1,0);
            call randgen(l, 'GAMMA', alpha);

                  do i=1 to n;
                        ran=rangam(1,alpha);
                              if l[i] <= test then
                                    l[i]= .; else l[i]= log(l[i]);
                  end;
```

```
                    do i=1 to n;
                          if k[i]=. then c=c+1;
                          if l[i]=. then c1=c1+1;
                    end;
d[j,1]=alpha;
d[j,2]=c/n;
d[j,3]=c1/n;
c=0;
c1=0;
j=j+1;

end;

cn={"a" "c" "c1"};

create Data from d[colname=cn];
append from d;
close;

proc template;
define statgraph Graph;
dynamic _A _C _A2 _C1A;
begingraph;
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
      layout overlay / xaxisopts=( label=('Alpha')) yaxisopts=(
label=('Proportion of zeros'));
         seriesplot x=_A y=_C / name='series' legendlabel='Rangam'
connectorder=xaxis lineattrs=(color=CX9C3418 pattern=SOLID thickness=2 );
         seriesplot x=_A2 y=_C1A / name='series2' legendlabel='Randgen'
connectorder=xaxis lineattrs=(color=CX606260 pattern=SHORTDASH thickness=2
);
         discretelegend 'series' 'series2' / opaque=false border=true
halign=center valign=top displayclipped=true down=1 order=columnmajor
location=inside;
      endlayout;
   endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.DATA template=Graph;
dynamic _A="A" _C="C" _A2="A" _C1A="C1";
run;
```

An overview of kernel density estimation

Hester Louretha Stoop 14034027

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr MT Loots

Department of Statistics, University of Pretoria

30 October 2017

Abstract

Density estimation is to construct an estimate of the density function using observed data and can be classified to be parametric or nonparametric. Kernel density estimation is a popular nonparametric density estimation technique, evolved from the histogram and naive estimator. The univariate and multivariate case will be explored. A popular way of measuring the global accuracy of the kernel estimator is the Mean Integrated Square Error (MISE), which are used throughout to compare performances between different estimates. Asymptotic approximations of the MISE are used to determine the optimal bandwidth and the optimal kernel function. For the univariate case there has been good progress towards bandwidth estimation methods but for the multivariate case, the progress has been relatively slow. Different bandwidth methods will be studied and examples on simulated and real datasets.

Declaration

I, *Hester Louretha Stoop,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Hester Louretha Stoop

_ _

Dr MT Loots

_ _

30/10/2017

Acknowledgements

I acknowledge the financial support from the Center for AI Research, Meraka Institute, CSIR and STATOMET.

Contents

List of Figures

List of Tables

1 Introduction

One of the key concepts in statistics is the probability density function. The purpose of this function is to describe a random variable with its probabilities. Specifying the probabilities of a random variable gives a natural description of the distribution of the random variable and through the distribution of the random variable most, if not all the required information can be gathered.

Density estimation is to construct an estimate of the density function using observed data. Density estimates can be used to assess the multimodality, skewness and also give other valuable information [1]. It plays a crucial part in machine learning, classification, and clustering [21].

A graphical representation of a density estimate is of great value when presenting data back to a client. Therefore, the density estimate is fairly easy to understand thus an explanation and illustration can be provided for non-mathematicians [21]. Although the presentation of the obtained results is a vital aspect of Statistics it is often overlooked.

Furthermore, density estimation can be classified to be parametric or nonparametric. The predominant distinction between the parametric and the nonparametric approach towards density estimation lies in the assumptions made of the observed data. The parametric approach assumes that the observed data's underlying distribution is one of the known parametric family of distributions. In the nonparametric case, there are no such strong assumptions made about the observed data, therefore this approach is more robust. For the parametric approach, there has to be some degree of prior knowledge of the observed data before assumptions are made. When a parametric model is specified incorrectly, the subsequent statistical analysis may lead to inconsistent estimators and tests [21]. The nonparametric approach is far more flexible in modeling a dataset and it is also not affected by the specification bias [1]. This approach allows the data to "speak" for itself in determining the estimate of the density more than the parametric approach allows where the estimate is constrained to fall in a given parametric family of distributions [21]. Many nonparametric density estimators evolved from the classical histogram [21]. According to Silverman [21], many statisticians are moving from parametric models towards non-parametric models in search for increased flexibility as needed for data exploration.

The disadvantages of the histogram and the naive estimator led to the motivation for the kernel estimator [21]. The simplest explanation of the kernel estimator would be to consider the sum of 'bumps' placed at each observation. The kernel function (a known density function) determines the shape of the 'bump'. The width of the 'bump' is controlled by the bandwidth, otherwise known as the smoothing parameter. Note that the variance is a function of the bandwidth. Only fixed bandwidth will be considered

here i.e. the bandwidth is held constant across the domain of the data.

Kernel density estimation is one of the most commonly used nonparametric techniques in data analysis [21]. According to Silverman [21], this is not necessarily the best method to use in all cases, but there are more than a few reasons to consider the kernel method as a primary option. The kernel method is extensively relevant, particularly in the univariate case. It has strong intuitive appeal, conceptual simplicity, and its current computing standards make it inexpensive to implement [21]. Silverman also mentioned that it is worthwhile understanding the kernel method's behavior before considering other non-parametric methods [21]. Kernel density estimation is separated into the univariate case and multivariate case. Analysis of multivariate data is very important since the outcome of a situation usually depends on more than one variable. The presentation of multivariate densities is difficult. It is easy to understand the graphics of a contour plot or a two-dimensional density function but it will take experts with sophisticated graphics facilities to understand a presentation of a three-dimensional density function [21]. Therefore, it is usually not the graphics that are required but the function so that it can be used in some other statistical technique to give the required results. The multivariate kernel estimator is a generalization of the univariate kernel estimator [21].

One-sided or two-sided bounded data can be problematic for kernel density estimation. Since the kernel function used in this method is not bounded it will result in treating this observed bounded data as if it is not bounded. Kernel density estimates will often go beyond the bounds of the data and then the estimate is considerable bias at and near the bounds of the data [11]. As stated in [9], Jones provided a variety of boundary correction methods for kernel density estimation using a "generalised jackknifing" approach for many of the straightforward methods. A disadvantage of all generalised jackknife boundary corrections is that they have a natural tendency to assign negative values to the density estimate near the boundaries [11]. Jones and Foster [11] use a simple "non-negativisation" device which can be applied to any boundary corrected density estimate to show that a non-negative version of each and every generalized jackknife boundary corrected method can be obtained.

Kernel density estimation using a fixed bandwidth will result in a poor estimate when the observed data are from a long-tailed distribution or the data exhibit multimodality [16].Working with data from a long-tailed distribution and choosing a fixed bandwidth which adequately smooths near the mode of the distribution will leave the tail severely undersmoothed [21]. In other words, there will appear spurious (false) noise in the tails of the estimate because the bandwidth is fixed across the entire sample. If a larger fixed bandwidth value is chosen, the tails will be sufficiently smoothed but it will remove important features of the mode. For data from a multimodal distribution, it is difficult to find a single bandwidth

which will adequately differentiate between distinct peaks and valleys between the peaks [16]. Choosing a bandwidth which is too large can erase modes which were significant, and choosing a bandwidth which is too small may introduce spurious peaks by undersmoothing.

Multivariate fixed bandwidth estimation will only operate with a large sample [16]. Sain [16] introduced adaptive kernel density estimation which includes variable kernel estimators but also estimators that attempt to identify and utilise the local structure and other features in the underlying density through the sample data.

Some examples were kernel density estimation was applied to real datasets include the study of [10] were Jones applied univariate kernel estimation to the dataset "Lean Body Mass" in the Australian Institute of Sport which can be found in exercise 2.4 of [3] and the Old Faithful geyser dataset which consists of 107 eruption durations of the Old Faithful geyser given by [21] was used in the studies for [12]. Different inference procedures like pattern recognition, computer vision, machine learning and data mining use kernel estimation techniques extensively to produce the required results [15].

2 Background theory

The disadvantages of the histogram and naive estimator led to the motivation for the kernel estimator [21]. This section will refer to [21] concerning the background theory of the kernel estimator.

The two main components of the histogram are the origin x_0 and the bin width h. The bins of the histogram fall in the intervals $[x_0 + mh,\, x_0 + (m+1)h)$ for positive and negative integers m. Hence the definition of the histogram follows as

$$\hat{f}(x) = \frac{1}{nh}.$$

This value $\hat{f}(x)$ can be seen as the number of X_i in the same bin as x. Choosing different origins can lead to a non-statistician drawing different conclusions. The bin width controls the amount of smoothing in the histogram. The histogram is discontinuous at various points and this makes it mathematically difficult to calculate the derivative of the estimate. There is some degree of inefficiency when investigating the data using a histogram. A lack of accuracy in the study of measurements, properties, and relationships may be present. There can be improved upon the histogram to attain more valuable information.

The naive estimator $\hat{f}(x)$ is an estimator of the density that follows naturally and contains a weight function $w(x)$, defined as

$$w(x) = \begin{cases} \frac{1}{2} & if\ |x| < 1 \\ 0 & otherwise \end{cases}$$

then, the definition of the naive estimator follows as

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} w\left(\frac{x - X_i}{h}\right).$$

The simplest explanation of how the naive estimator is constructed would be to consider the sum of 'boxes' placed at each observation. The height of the 'box' is $(2nh)^{-1}$ and the width of the 'box' is $2h$. The height and width of the 'box' are controlled by the bin width h, also known as the smoothing parameter. A weight function is included in the definition of the naive estimator. The purpose of the weight function is to estimate the proportion of the sample contained in the interval $(x - h,\ x + h)$ for any given h. The naive estimator is similar to the histogram if every observation sampled, is the center of a sampling interval. If this is the case, then the influence of the choice of the origin is limited but the bin width is still controlling the amount by which the data is smoothed. The naive estimator has some mathematical properties that are better than the histogram. It is not fully successful in presenting the estimate of the density and it can lead to a misinterpretation by an untrained observer.

Some of the difficulties of the naive estimator are overcome by changing the weight function in the definition of the naive estimator to a kernel function and this leads to the kernel density estimator. The kernel estimator and the naive estimator are constructed similarly. In the case of the naive estimator, the sum of 'boxes' placed at each observation was considered. For the kernel estimator, the sum of 'bumps' placed at each observation will be considered. The kernel function determines the shape of the 'bump' and the width of the 'bump' is controlled by the bandwidth h, also known as the smoothing parameter. Deciding what the value of the bandwidth should be, takes a lot of time and effort. Since the bandwidth plays a crucial role in the performance of the estimator. When the bandwidth is too small it will result in a density estimate which is too 'spiky'. A false structure of the density estimate (spurious features) may become visible. When the bandwidth is too large important information regarding the underlying structure of the density estimate may be lost.

One-sided or two-sided bounded data can be problematic for kernel density estimation. Since the kernel function used in the method is not bounded it will result in treating the observed bounded data as if it is not bounded. The kernel density estimation has also a minor disadvantage when the observed data are from a long-tailed distribution. There will appear spurious (false) noise in the tails of the estimate because the bandwidth is fixed across the entire sample.

3 Univariate kernel density estimation

3.1 Definition of the kernel estimator

The kernel estimate $\hat{f}(x)$ of a continuous univariate density function $f(x)$ is defined as

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - Xi),\ where\ K_h(u) = \frac{1}{h}K\left(\frac{u}{h}\right).$$

The bandwidth or the smoothing parameter is represented by h and the kernel function is $K(x)$. This kernel function is usually a symmetric probability density function like the normal density but it can be some other type of function although it needs to be a valid density function [21]. When considering a fixed bandwidth the variance of the kernel function is also referred to as the bandwidth. The density estimate also inherits all the continuity and differentiability properties of the kernel function [21].

3.2 Optimal bandwidth and kernel theory

3.2.1 Optimal bandwidth

The choice of the bandwidth is crucial to the performance of the kernel estimator [13, 21]. Measuring the performance of a kernel estimator is to measure how well the kernel estimate $\hat{f}(x)$ fits the data. In other words the closeness of a kernel density estimate, $\hat{f}(x)$ to the true density function, $f(x)$ [4]. There are a few types of measures which can be used, they are also referred to as criterions. These measures can be categorized into two main classes namely goodness-of-fit for known distributions (where the true density function is known or assumed) and goodness-of-fit measures for unknown distributions (where the true density function is not known or assumed) [22]. Goodness-of-fit measures for known distributions are typically based on least squares approaches that attempt to minimise the squared distance between the kernel estimate $\hat{f}(x)$ and the true density function. Some of these popular measures are the mean-squared error, mean integrated squared error, and the asymptotic mean integrated squared error [22]. Goodness-of-fit measures for unknown distributions are typically based on maximum likelihood approaches that attempt to maximise the product of the likelihood of each data point belonging to the estimated distribution (without knowing or assuming the true density function) [22]. The focus will be on goodness-of-fit measures for known distributions for more information regarding the goodness-of-fit measures for unknown distributions refer to Van der Wald in [22].

Expected value and variance of the kernel estimator

For the following derivations assume that the random sample $X_1, X_2, \ldots X_n$ is independent and identically distributed. The assumptions of independence and identically distributed are not common in

real life situations but it is necessary to build the standard framework for density estimation [21]. The expected value of the kernel estimate is defined to be

$$\begin{aligned} E[\hat{f}(x)] &= \frac{1}{nh}\sum_{i=1}^{n} E\left[K\left(\frac{x-X_i}{h}\right)\right] \\ &= \frac{1}{h}E\left[K\left(\frac{x-X}{h}\right)\right]. \end{aligned}$$

For a bounded measurable function $g(y)$, the result

$$E[g(X_i)] = \int g(y)f(y)dy$$

can be used to set $g(y) = K\left(\frac{x-y}{h}\right)$ since K is a bounded measureable function. Hence the expected value of the kernel estimator will be

$$\begin{aligned} E[\hat{f}(x)] = \frac{1}{h}E\left[K\left(\frac{x-X}{h}\right)\right] &= \frac{1}{h}\int K\left(\frac{x-y}{h}\right)f(y)dy \\ &= \int K_h(x-y)f(y)dy \\ &= (K_h * f)(x) \end{aligned}$$

(where * is the convolution operator). The variance of the kernel estimator will be

$$\begin{aligned} var\{\hat{f}(x)\} &= E[\hat{f}(x)^2] - E[\hat{f}(x)]^2 \\ &= E\left[\frac{1}{n^2}\sum_{i=1}^{n} K_h(x-X_i)^2\right] - (K_h * f)^2(x) \\ &= \frac{1}{n^2}\sum_{i=j} E[K_h(x-X_i)K_h(x-X_j)] + \frac{1}{n^2}\sum_{i\neq j} E[K_h(x-Xi)K_h(x-X_j)] - (K_h * f)^2(x) \\ &= \frac{1}{n}E[K_h(x-X)^2] + \frac{1}{n^2}\sum_{i\neq j} E[K_h(x-X_i)]E[K_h(x-X_j)] - (K_h * f)^2(x) \\ &= \frac{1}{n}(K_h^2 * f)(x) + \left(\frac{n(n-1)}{n^2} - 1\right)(K_h * f)^2(x) \\ &= \frac{1}{n}\left\{(K_h^2 * f)(x) - (K_h * f)^2(x)\right\}. \end{aligned}$$

Expressions for MSE and MISE

A natural measure of the error between the estimator and the true density at a single point x is the mean square error and it is defined as

$$MSE(\hat{f}(x)) = E\left[\left\{\hat{f}(x) - f(x)\right\}^2\right].$$

By this basic result $E[X^2] = var(X) + (E[X])^2$we can rewrite MSE as

$$
\begin{aligned}
MSE(\hat{f}(x)) &= var\{\hat{f}(x) - f(x)\} + \left\{E\left[\hat{f}(x) - f(x)\right]\right\}^2 \\
&= var\{\hat{f}(x)\} + \left\{E\left[\hat{f}(x) - f(x)\right]\right\}^2 \\
&= var\{\hat{f}(x)\} + \left\{E\left[\hat{f}(x)\right] - f(x)\right\}^2 \\
&= \frac{1}{n}\left\{(K_h^2 * f)(x) - (K_h * f)^2(x)\right\} + \left\{(K_h * f)(x) - f(x)\right\}^2.
\end{aligned}
$$

In the second step it is shown that the MSE is the sum of the squared bias defined as $\left[bias(\hat{f}(x))\right]^2 = \left\{E\left[\hat{f}(x) - f(x)\right]\right\}^2$ and the variance of the estimator. There is a tradeoff between the bias and the variance

$$
\begin{aligned}
bias(\hat{f}(x)) &= E\left[\hat{f}(x) - f(x)\right] \\
&= E\left[\hat{f}(x)\right] - f(x) \\
&= (K_h * f)(x) - f(x).
\end{aligned}
$$

The bias depends on the bandwidth h and if h is chosen to be a function of the sample size n then the bias will also indirectly depend on n [21]. From the expression $var\{\hat{f}(x)\} = \frac{1}{n}\left\{(K_h^2 * f)(x) - (K_h * f)^2(x)\right\}$ it follows that the variance of the kernel estimator directly depends on h and n. Hence the variance can be reduced by adjusting the bandwidth but then the bias will increase and vice versa [21]. A popular way of measuring the global accuracy of the kernel estimator is the Mean Integrated Square Error since it is the most mathematically tractable criterion and is the most commonly used in practice [4]. Mathematically tractable refers to that the MISE can be solved in terms of a closed-form expression. The MISE is also simple and allows for deep analysis [21]. Defined by the following expression

$$
MISE(\hat{f}(x)) = E\left[\int \{\hat{f}(x) - f(x)\}^2 dx\right]
$$

with $\int$ being the definite integral over the real line. The MISE can also be rewritten as the sum of the integrated square bias and the integrated variance.

$$
\begin{aligned}
MISE(\hat{f}(x)) &= \int E[\{\hat{f}(x) - f(x)\}^2] dx \\
&= \int MSE(\hat{f}(x)) dx \\
&= \int \left\{E\left[\hat{f}(x)\right] - f(x)\right\}^2 dx + \int var\{\hat{f}(x)\} dx \\
&= \int \left[bias(\hat{f}(x))\right]^2 dx + \int var\{\hat{f}(x)\} dx \\
&= \int \left\{(K_h * f)(x) - f(x)\right\}^2 dx + \int \frac{1}{n}\left\{(K_h^2 * f)(x) - (K_h * f)^2(x)\right\} dx
\end{aligned}
$$

$$\begin{aligned}\int (K_h^2 * f)(x)dx &= \int\int K_h^2(x-y)f(x)dy\,dx \\ &= \int\int \frac{1}{h^2}K\left(\frac{x-y}{h}\right)^2 f(x)dy\,dx \\ &= \int\int \frac{1}{h}K(z)^2 f(x-hz)dz\,dx \\ &= \int \frac{1}{h}K(z)^2 \int f(x-hz)dx\,dz \\ &= \frac{1}{h}\int K(x)^2 dx\end{aligned}$$

The second set of equations is to show a simpler expression for $\int (K_h^2 * f)(x)dx$ which leads to a simpler expression for the MISE. The reason for the last step is $f(x-hz)$ will integrate to one over the definite integral since $f(x)$ is a valid density function. Hence a simpler expression for the MISE

$$\begin{aligned}MISE(\hat{f}(x)) &= \int (K_h * f)^2(x)dx - 2\int (K_h * f)(x)f(x)dx + \int f(x)^2 dx - \frac{1}{nh}\int K(x)^2 dx - \frac{1}{n}\int (K_h * f)^2(x)dx \\ &= \left(1-\frac{1}{n}\right)\int (K_h * f)^2(x)dx - 2\int (K_h * f)(x)f(x)dx - \frac{1}{nh}\int K(x)^2 dx + \int f(x)^2 dx.\end{aligned}$$

Approximations for the MSE and the asymptotic MISE

Asymptotic results will provide intuition on how the bandwidth operates as a smoothing parameter [10]. For approximations of the MSE and AMISE, the following assumptions are needed. The second derivative of $f(x)$ is a continuous, quadratically integrable (measurable function for which the integral of the square of the absolute value is finite) and monotonic function (either a completely increasing or completely decreasing function). The kernel function is symmetric about the origin and satisfies the following

$$\int K(z)dz = 1$$

$$\int zK(z)dz = 0$$

$$\int z^2 K(z)dz = \mu_2(K) = \sigma_k^2 \neq 0.$$

Typically K is set to be a symmetric probability density function like the normal density and σ_k^2 is a constant which is the variance of this density function [21]. Note the following notation $R(g) = \int g(x)^2 dx$ then let the bandwidth be $h_n$which is a non-random sequence of positve numbers such that $\lim_{n\to\infty} h_n = 0$ and $\lim_{n\to\infty} nh_n = \infty$.

Consider the following change of variable $z = \frac{x-y}{h}$ in the expressions for the expected value and

variance of the kernel estimator

$$\begin{aligned} E\left[\hat{f}(x)\right] &= (K_h * f)(x) \\ &= \frac{1}{h}\int K\left(\frac{x-y}{h}\right) f(y)dy \\ &= \int K(z)f(x-hz)dz \end{aligned}$$

$$\begin{aligned} var\{\hat{f}(x)\} &= \frac{1}{n}\left\{(K_h^2 * f)(x) - (K_h * f)^2(x)\right\} \\ &= \frac{1}{n}\left\{\frac{1}{h}\int K\left(\frac{x-y}{h}\right)^2 f(y)dy - \left(\frac{1}{h}\int K\left(\frac{x-y}{h}\right) f(y)dy\right)^2\right\} \\ &= \frac{1}{nh}\int K(z)^2 f(x-hz)dz - \frac{1}{n}\left\{\int K(z)f(x-hz)dz\right\}^2. \end{aligned}$$

Theorem 1. *Taylor's theorem: Let f be a real-valued function defined on the whole real space and x a element of the whole real space. Assume f has q continuous derivatives in the interval $(x-\delta,\, x+\delta)$ for some $\delta > 0$. Then for any sequence α_n converging to zero,*

$$f(x+\alpha_n) = \sum_{j=0}^{q} \frac{\alpha_n^j}{j!} f^{(j)}(x) + o(\alpha_n^q).$$

A Taylor series expansion of $f(x-hz)$ about x gives

$$f(x-hz) = f(x) - hzf'(x) + \frac{1}{2}h^2z^2f''(x) + o(h^2)$$

therefore,

$$\begin{aligned} bias(\hat{f}(x)) &= E\left[\hat{f}(x)\right] - f(x) \\ &= f(x)\int K(z)dz - hf'(x)\int zK(z)dz + \frac{1}{2}h^2f''(x)\int z^2K(z)dz + o(h^2) - f(x) \\ &= f(x) + \frac{1}{2}h^2f''(x)\sigma_k^2 + o(h^2) - f(x) \\ &= \frac{1}{2}h^2f''(x)\sigma_k^2 + o(h^2) \\ &\approx \frac{1}{2}h^2f''(x)\sigma_k^2. \end{aligned}$$

Similarly,

$$\begin{aligned} var\{\hat{f}(x)\} &= \frac{1}{nh}\int K(z)^2 f(x-hz)dz - \frac{1}{n}\int K(z)f(x-hz)dz \\ &= \frac{1}{nh}\int K(z)^2 f(x-hz)dz - \frac{1}{n}\left\{f(x) + bias(\hat{f}(x))\right\}^2 \\ &= \frac{1}{nh}\int K(z)^2 f(x-hz)dz - \frac{1}{n}\left\{f(x) + o(h^2)\right\}^2 \\ &= \frac{1}{nh}\int K(z)^2 \left\{f(x) - hzf'(x) + \dots\right\} dz + o(n^{-1}) \\ &= \frac{1}{nh} f(x)\int K(z)^2 dz + o(n^{-1}) \\ &\approx \frac{1}{nh} R(K) f(x) \end{aligned}$$

where $R(K) = \int K(x)^2 dx$. Using the above it follows that

$$\begin{aligned} MSE(\hat{f}(x)) &= var\{\hat{f}(x)\} + [bias(\hat{f}(x))]^2 \\ &\approx \frac{1}{nh} R(K) f(x) + \frac{1}{4} h^4 \sigma_k^4 f''(x)^2 \end{aligned}$$

$$\begin{aligned} AMISE(\hat{f}(x)) &= \int bias(\hat{f}(x))^2 dx + \int var\{\hat{f}(x)\}dx \\ &\approx \frac{1}{4} h^4 \sigma_k^4 R(f'') + \int \frac{1}{nh} R(K) f(x) dx \\ &\approx \frac{1}{4} h^4 \sigma_k^4 R(f'') + \frac{1}{nh} R(K) \end{aligned}$$

where $R(f'') = \int f''(x)^2 dx$. Therefore the bandwidth h which minimizes the AMISE is the optimal bandwidth given by

$$h_{AMISE} = \left[\frac{R(K)}{n\sigma_k^4 R(f'')}\right]^{\frac{1}{5}}.$$

The following notation is also sometimes encountered [24]

$$h_{AMISE} = \left[\frac{R(K)}{n[\mu_2(K)]^2 R(f'')}\right]^{\frac{1}{5}}.$$

3.2.2 Optimal kernel theory

Cline [2] showed that for a kernel estimator to be acceptable and valid the kernel function has to be symmetric and unimodal. Substituting h_{AMISE} back into the approximation for AMISE will result in

$$\frac{5}{4} C(K) R(f'')^{1/5} n^{-4/5}$$

where $C(K)$ is a constant given by

$$C(K) = \sigma^{2/5} R(K)^{4/5}.$$

Kernel function	$K(z)$	Efficiency (to 4 d.p.)
Epanechnikov	$\begin{cases} \frac{3}{4\sqrt{5}}\left(1-\frac{1}{5}z^2\right) & for\ -\sqrt{5} \leq z \leq \sqrt{5} \\ 0 & otherwise \end{cases}$	1
Biweight	$\begin{cases} \frac{15}{16}(1-z^2)^2 & for\ \lvert z\rvert < 1 \\ 0 & otherwise \end{cases}$	0.9939
Triangular	$\begin{cases} 1-\lvert z\rvert & for\ \lvert z\rvert < 1 \\ 0 & otherwise \end{cases}$	0.9859
Gaussian	$\frac{1}{\sqrt{2\pi}} exp\{-\frac{1}{2}z^2\}$	0.9512
Rectangular	$\begin{cases} \frac{1}{2} & for\ \lvert z\rvert < 1 \\ 0 & otherwise \end{cases}$	0.9295

Table 1: Some kernel functions and their efficiencies

If the bandwidth is chosen correctly then the kernel function, K should be chosen such that $C(K)$ will be a small value [21]. The Epanechnikov kernel

$$K_e(z) = \begin{cases} \frac{3}{4\sqrt{5}}\left(1-\frac{1}{5}z^2\right) & -\sqrt{5} \leq z \leq \sqrt{5} \\ 0 & otherwise \end{cases}$$

solves this problem and makes it theoretically possible to obtain a small value of the MISE [21]. The efficiency of any symmetric kernel function is defined by Silverman [21] as

$$\begin{aligned} eff(K) &= \{C(K_e)/C(K)\}^{4/5} \\ &= \frac{3}{5\sqrt{5}}\sigma_k^{-1/2}R(K)^{-1}. \end{aligned}$$

Hence the efficiency of a kernel function as defined by Silverman is a relative efficiency to the Epanechnikov kernel. Table 1 is found in [21] and Silverman describes the purpose of this table as to see how close the efficiency of different kernel functions are to one another. Therefore the choice of the kernel function should be based on other considerations as well, such as the degree of differentiability required or the computational effort involved.

A practical way of finding the optimal kernel function is to compare plots of different kernel estimates for the dataset of interest but these comparisons become meaningless when identical bandwidths are used [13]. Since for the obvious, it would be just to change the kernel function in the kernel estimate each time leaving the bandwidth unchanged. The problem is that the comparisons are not only based on different kernel functions but also based on different amounts of smoothing since local averaging drives the density estimator [13]. Marron and Nolan's [13] approach to solving this problem involved a so-called canonical representation of the kernels which is a rescaling of the kernel function. They found that each kernel function has exactly one rescaling that allows for functional comparison. This canonical rescaling of the

kernel separates the optimal bandwidth from the kernel function for a complete discussion see [13].

3.3 Bandwidth estimation methods

The most suitable corresponding bandwidth will be selected after the appropriate kernel function has been selected [22]. As discussed above it is very important to choose the correct amount of smoothing. According to Silverman [21], the purpose of the density estimate will give an indication of the amount of smoothing. For example, there will be relatively more smoothing present when the density estimate is used to analyse the underlying structure of the data than when the density estimate is used to present results back to the client. Since it is possible for the reader to do more smoothing 'by eye'. Pointed out by Jones [10] it is often important that a software package choose the amount of smoothing for the density estimate automatically for various reasons such as software packages need a default, people that are not experts in the field and will save time for the experts to give them a good functional starting point. Loader [12] emphasized on the opposite, how important it is to not just reply blindly on a bandwidth estimation method that will automatically give you the right bandwidth. Loader's reason for this is that if only the kernel estimate is plotted that fits, a very one-sided view of the bias-variance trade-off is obtained, seeing the variance, but not the bias.

Bandwidth estimation methods are typically derived by optimizing an objective function with respect to the bandwidth of the kernel estimator and then finding an optimal solution [22]. An objective function measures the performance of the estimator i.e. the closeness of a kernel density estimate $\hat{f}(x)$ to the true density function $f(x)$ [4]. Objective functions refer to types of criterions discussed above, for example, the MSE, MISE and the AMISE.

Jones [10] classified bandwidth estimation methods that were developed before the 1990's as "first generation" methods. Loader [12] refers to these methods as classical estimation methods. The bandwidth methods developed after the 1990's was classified by Jones as "second generation" methods. Also known as plug-in methods. There has been a massive improvement upon bandwidth estimation methods' performances. They are far more superior and ready to be used as defaults in some software packages [10]. Sheather and Jones [20] developed the so-called "solve-the-equation plug-in" method. According to Jones [10], it is the best method in terms of the overall performance and should become the benchmark for good performance.

Silverman's rule of thumb

Silverman's rule of thumb consists of substituting the unknown part of h_{AMISE} which is $R(f'')$ by an estimated value based on a standard family of distributions. Silverman specifically used the normal

distribution as a natural choice for the standard family of distributions [21]. In other words, Silverman based this method on a normal reference rule. Let ϕ represents the standard normal density function then the unknown part of h_{AMISE} follows as

$$\begin{aligned} R(f'') &= \int f''(x)^2 dx \\ &= \sigma^{-5} \int \phi''(x)^2 dx \\ &= \frac{3}{8}\pi^{-1/2}\sigma^{-5} \\ &\approx 0.212\sigma^{-5}. \end{aligned}$$

Using this and a Gaussian kernel function h_{AMISE} will result in

$$\begin{aligned} h_{AMISE} &= (4\pi)^{-1/10}\frac{3}{8}\pi^{-1/2}\sigma n^{-1/5} \\ &= \left(\frac{4}{3}\right)^{1/5}\sigma n^{-1/5} \\ &= 1.06\sigma n^{-1/5}. \end{aligned}$$

The next step will then only be to substitute an estimate for σ like the sample standard deviation s or a more robust estimate for σ into this expression. If the distribution of the population is normal this method will work well but Silverman stated that this method may oversmooth the estimate of the density if the true distribution of the population is multimodal as a result of the value of $(R(f''))^{1/5}$ being larger relative to the standard deviation [21].

Silverman investigated the sensitivity of the optimal bandwidth to skewness and kurtosis in unimodal distributions. Discovering that for heavily skewed data using the above bandwidth estimator will oversmooth the estimate of the density but this bandwidth estimator is remarkably insensitive to kurtosis within the t family of distributions [21]. Therefore an improvement of this bandwidth method will be to use a more robust measure of spread i.e. the interquartile range R of the underlying normal distribution

$$h_{AMISE} = 0.79Rn^{-1/5}$$

but unfortunately, this will oversmooth the density estimate even more if the true distribution of the population is bimodal. Silverman reached the conclusion that the best will be using a adaptive estimate

of spread considering both situations

$$A = min(\hat{\sigma},\, R/1.34)$$

hence h_{AMISE} will be

$$h_{AMISE} = 1.06An^{-1/5}.$$

Reducing the factor from 1.06 to 0.9 will improve the bandwidth estimator even further. Hence Silverman's rule of thumb will be

$$h_{ROT} = 0.9An^{-1/5}.$$

Silverman stated that this rule of thumb will do very well for a wide range of densities and is trivial to evaluate. For many purposes, it will be an adequate choice of the bandwidth and for other purposes, it will be a good starting point in search for an adequate choice of the bandwidth [21].

Least squares (unbiased) cross-validation

The idea came from representing the integrated squared error (ISE) as

$$ISE_h(\hat{f}) = \int \left(\hat{f}_h(x) - f(x)\right)^2 dx = \int \hat{f}_h(x)^2 dx - 2\int \hat{f}_h(x)f(x)dx + \int f(x)^2 dx.$$

The optimal bandwidth h_{LSCV} that would minimize ISE will also be the same bandwidth that will minimize the first two terms of the above expression. The first term, $\int \hat{f}_h(x)^2 dx$ is entirely known and the second term $\int \hat{f}_h(x)f(x)dx$ can be estimated by using method of moments [10]. Refer to [21] where Silverman gives a brief discussion on the least squares cross-validation method.

Biased cross-validation

Scott and Terrell [19] developed the following bandwidth method. The biased cross-validation method is constructed such that it attempts to directly minimize the AMISE. Estimating the unknown part in AMISE i.e. $R(f'')$ results in another kernel density estimation problem. Hence it requires selecting another bandwidth. In [10] this difficulty is addressed by creating a dummy variable of minimization and then taking the bandwidth to be this dummy variable. The smallest local minimizer h_{BCV} of

$$BCV_h = \frac{1}{nh}R(K) + h^4\left[R(\hat{f}''_\alpha) - \frac{R(K'')}{mh}\right]\left(\int x^2 K/2\right)^2$$

gives better empirical performance than the global minimizer. See [19] for a detailed discussion on the biased cross-validation method.

Solve-the-equation plug-in-approach

Recall the integrated bias and the integrated variance approximations

$$\begin{aligned} Bias_h(x) &= E\left[\hat{f}_h(x) - f(x)\right] \\ &\approx \frac{h^2}{2} f'' \int z^2 K(z) dz \end{aligned}$$

$$var\{\hat{f}_h(x)\} = \frac{f(x)}{nh} \int K(z)^2 dz - \frac{f(x)^2}{n}$$

these approximations are used to describe a plug-in-approach [12]. Defined the optimal value for the bandwidth, h_{AMISE} to be

$$h_{AMISE} = \left[\frac{R(K)}{n\sigma_k^4 \int f''(x)^2 dx}\right]^{1/5}.$$

The idea of this method is to plug in an estimate of the unknown $\int f''(x)^2 dx$ into the expression for h_{AMISE} [10]. Loader [12] stated that this unknown $\int f''(x)^2 dx$ will usually be derived from a "pilot" kernel estimate of the second derivative

$$\begin{aligned} \hat{f}_p''(x) &= \frac{1}{np^3} \sum_{i=1}^{n} K''\left(\frac{X_i - x}{p}\right) \\ \int \hat{f}_p''(x)^2 dx &= \frac{1}{n^2 p^6} \sum_{i=1}^{n}\sum_{j=1}^{n} \int K''\left(\frac{X_i - x}{p}\right) K''\left(\frac{X_j - x}{p}\right) dx \end{aligned}$$

then, the standard normal kernel $\phi(x)$ is used to obtain

$$\int \hat{f}_p''(x)^2 dx = \frac{1}{n^2(\sqrt{2}p)^5} \sum_{i=1}^{n}\sum_{j=1}^{n} \phi^{(4)}\left(\frac{X_i - X_j}{\sqrt{2}p}\right).$$

A pilot bandwidth p is selected such that there exists a relation between the bandwidth used in the kernel estimator h and this pilot bandwidth p [12]. Clearly, the plug-in step alone doesn't solve what h should be since by varying p a wide range of choices for h as describe by Loader [12]. Jones also mentioned this difficulty in [10]. The solution that is most commonly used is to assume a relation between p and h [12]. There are various ideas on the plug-in method in the literature some stated by Loader [12] as different ideas for specifying p, alternative estimates of the unknown $\int f''(x)^2 dx$, using more accurate bias approximations in the expression for the MISE. Sheather and Jones [20] developed the following solution for the relation between p and h. The idea behind this solve-the-equation plug-in-approach developed by Sheather and Jones is to take h_{SJPI} to be the solution of the fixed-point equation

$$h = \left[\frac{R(K)}{nR(f^{"}{}_{g(h)})(\int x^2 K)^2}\right]^{1/5}.$$

An important difference between this approach and the biased cross-validation used in [10] is that for this approach the pilot bandwidth is written in the form $g(h)$. Since there is a great difference between bandwidths that are appropriate for curve estimation and bandwidths that are appropriate for estimating $R(f'')$ as referred to by [10].

3.4 Comparing classical bandwidth estimation methods with plug-in methods

This section refers to the comparison studies done by Jones [10] and Loader [12]. These comparison studies include evaluation of real and simulated data to compare the performance between classical and plug-in methods. The asymptotic performances of these methods were also explored. For the real data example, Jones [10] used the variable "Lean Body Mass" in the Australian Institute of Sports data in exercise 2.4 of [3]. The Old Faithful geyser dataset which consists of 107 eruption durations of the Old Faithful geyser given by [21] was used in the study of Loader [12]. The classical methods referred to by Jones are Silverman's rule of thumb, least squares cross-validation and biased cross-validation (using a dummy variable as referred to above). Loader [12] studied the likelihood cross-validation, akaike-style criterion, least squares cross-validation as classical methods. Both studies considered the SJPI plug-in method. Loader classified the biased cross-validation method as a plug-in method, assuming the relation to be $p = h$.

Through the real dataset example and considering other examples Jones concluded that the kernel density estimate using the classical method Silverman's rule of thumb is often extremely oversmoothed which leads to missing important features [10]. Least squares cross-validation is unreliable since its performance is variable and will often undersmooth the estimate. Biased cross-validation has a tendency to oversmooth the estimate and its performance is also variable. Jones stated supported by the real data example that the SJPI plug-in method results in the best performer which is consistent, stable and can also be used as the default in software packages [10].

It is extremely important to perform diagnostics to detect lack of fit and this is often neglected [12]. Loader used the Old Faithful data, simulations based on a smoothed bootstrap approach, residual diagnostics and higher order fits to conclude that classical methods are correct in choosing small bandwidths, and the plug-in methods incorrectly oversmooth the estimate, with regard to the integrated square error loss function.

Loader [12] draws the conclusion that much of the criticism towards classical methods especially cross-validation can actually be pointed towards kernel estimation and fixed bandwidth selection methods.

Loader [12] discovered that classical methods particularly the least squares cross-validation oversmooth the estimate when the underlying distribution of the data has heavy tails. Also pointed out by [18] when the data has heavy tails, the least cross-validation method produces inconsistent fixed bandwidth kernel estimates. This was the motivation for variable bandwidth kernels derived by Schuster and Gregory [18] where the authors concluded a fixed bandwidth estimate is inadequate for data with heavy tails.

In [12] Loader discussed some flaws of the literature for comparison studies done between classical selectors and plug-in methods. The plug-in methods essentially involve making some considerable prior assumptions about what the bandwidth should be and if this information is wrong the plug-in estimates will fail. Plug-in methods use higher order pilot estimates to obtain their information from the data. If classical selectors were also allowed to consider higher order methods then the result would be better estimates and comparison studies may have a different conclusion.

4 Multivariate kernel density estimation

4.1 Definition of the multivariate kernel estimator

The definition of the multivariate kernel density estimator, $\hat{f}(\underline{x})$ as in [5] for a d-variate random sample $\underline{X}_1, \underline{X}_2, \ldots, \underline{X}_n$is defined as

$$\hat{f}(\underline{x}) = \frac{1}{n}\sum_{i=1}^{n} |\boldsymbol{H}|^{-1/2} K\left[\boldsymbol{H}^{-1/2}(\underline{x} - \underline{X}_i)\right] = \frac{1}{n}\sum_{i=1}^{n} K_{\boldsymbol{H}}(\underline{x} - \underline{X}_i)$$

where $\underline{x} = (x_1, x_2, \ldots, x_n)'$ and $\underline{X}_i = (X_{i1}, X_{i2}, \ldots, X_{id})'$ for $i = 1, 2, \ldots, n$. The kernel function, $K(\underline{x})$ is now defined for a d-dimensional $\underline{x}$ satisfying

$$\int_{R^d} K(\underline{x}) d\underline{x} = 1.$$

Usually $K(\underline{x})$ is a symmetric probability density function and the bandwidth matrix, $\boldsymbol{H}$: $d \times d$ is symmetric and positive definite [5, 24].

In general the bandwidth matrix,$\boldsymbol{H}$ will have $\frac{1}{2}d(d+1)$ independent elements which means that even when d is not that large it will still be a considerable number of parameters that have to be chosen. Wand and Jones suggested the following two restrictions to simplify the kernel estimator [24]. Let $\mathcal{F}$ be the class of symmetric, positive definite $d \times d$ matrices, where the restriction $\boldsymbol{H}\epsilon\mathcal{D}$ is imposed, with $\mathcal{D} \subseteq \mathcal{F}$ being the subclass of diagonal positive definite $d \times d$ matrices. In other words this restriction $\boldsymbol{H}\epsilon\mathcal{D}$ leads to a bandwidth matrix with only diagonal elements $\boldsymbol{H} = diag(h_1^2, h_2^2, \ldots, h_d^2)$. Now the

kernel estimator can be written as

$$\hat{f}(\underline{x}) = \frac{1}{n}\left(\prod_{l=1}^{d} h_l\right)^{-1} \sum_{i=1}^{n} K\left(\frac{x_1 - X_{i1}}{h_1}, \frac{x_2 - X_{i2}}{h_2}, \ldots, \frac{x_d - X_{id}}{h_d}\right).$$

A futher (verkeerd gespel) restriction can be imposed by letting $\boldsymbol{H}\epsilon\mathcal{S}$ where $\mathcal{S} = \{h^2\boldsymbol{I} : h > 0\}$ and leads to the single bandwidth kernel estimator

$$\hat{f}(\underline{x}) = \frac{1}{n}h^{-d}\sum_{i=1}^{n} K\{(\underline{x} - \underline{X}_i)/h\}.$$

Hence Wand and Jones declared that there is a hierarchical class of smoothing parameterisation to choose from when using a multivariate kernel estimator and this will be discussed in detail later on. (cross referecing)

4.2 Optimal bandwidth matrix and kernel theory

Optimal bandwidth matrix

The performance of multivariate kernel density estimation is dependent on the selected bandwidth matrix and according to Duong [7], the development of good bandwidth estimation methods has been relatively slow. Recall that performance is measured by how close the kernel estimate $\hat{f}(\underline{x})$ is to the true density function $f(\underline{x})$. In the multivariate case, MISE will be defined as

$$\begin{aligned} MISE(\hat{f}(\underline{x})) &= E\left[\int \left(\hat{f}(\underline{x}) - f(\underline{x})\right)^2 d\underline{x}\right] \\ &= \int Bias\left(\hat{f}(\underline{x})\right)^2 d\underline{x} + \int var\left(\hat{f}(\underline{x})\right) d\underline{x}. \end{aligned}$$

For the derivations of finding the optimal bandwidth matrix first assume that the d-variate sample $\underline{X}_1, \underline{X}_2, \ldots, \underline{X}_n$ is independent and identically distributed. The expected value, bias, and variance of the kernel density estimate are given by

$$\begin{aligned} E\left[\hat{f}(\underline{x})\right] =& E\left[K_{\boldsymbol{H}}(\underline{x} - \underline{X})\right] \\ =& \int K_{\boldsymbol{H}}(\underline{x} - \underline{y}) f(\underline{y}) d\underline{y} \\ =& (K_{\boldsymbol{H}} * f)(\underline{x}) \end{aligned}$$

$$Bias\left(\hat{f}(\underline{x})\right) = (K_{\boldsymbol{H}} * f)(\underline{x}) - f(\underline{x})$$

$$Var\left(\hat{f}(\underline{x})\right) = n^{-1}\left\{(K^2_{\boldsymbol{H}} * f)(\underline{x}) - (K_{\boldsymbol{H}} * f)(\underline{x})^2\right\}$$

(where * is the convolution). Now obtaining an expression for the MISE through combining the squared bias and the variance

$$MISE(\hat{f}(\underline{x})) = n^{-1}\int[(K_{\boldsymbol{H}}^2 * f)(\underline{x}) - (K_{\boldsymbol{H}} * f)(\underline{x})^2]d\boldsymbol{x} + \int[(K_{\boldsymbol{H}} * f)(\underline{x}) - f(\underline{x})]^2 d\underline{x}$$
$$= n^{-1}R(K)|\boldsymbol{H}|^{-1/2} + (1-n^{-1})\int(K_{\boldsymbol{H}} * f)(\underline{x})^2 d\underline{x} - 2\int(K_{\boldsymbol{H}} * f)(\underline{x})f(\underline{x})d\underline{x} + R(f)$$

where $R(g) = \int g(\underline{x})^2 d\underline{x}$ for any square integrable function g. In the multivariate case MISE is not a mathematically tractable expression i.e. it does not have a closed form except if the true density function $f(\underline{x})$ has a normal mixture density and the Gaussian kernel function is used [4]. Therefore as referred to by Duong [4], it is extremely difficult to find the minimizer $\boldsymbol{H}_{MISE}$of the MISE

$$\boldsymbol{H}_{MISE} = \underset{\boldsymbol{H}\epsilon\mathcal{H}}{argmin} MISE(\hat{f}(\underline{x}))$$

where $\mathcal{H}$ is the space of symmetric, positive definite $d \times d$ matrices. The asymptotic approximation of the MISE is a mathematically tractable expression and the minimizer $\boldsymbol{H}_{AMISE}$ of the AMISE can be found more easily than $\boldsymbol{H}_{MISE}$. This asymptotic approximation will provide intuition on how the bandwidth matrix operates as the smoothing parameter. Wand and Jones derived a simple asymptotic approximation to the MISE using the multivariate version of Taylor's theorem [24].

Theorem 2. *Multivariate version of Taylor's theorem*

Let g be a d-variate function and $\underline{\alpha}_n$be a sequence of $d\times 1$ vectors with all components tending to zero. Also, let $\mathcal{D}_g(\underline{x})$ be the vector of first-order partial derivatives of g and $\mathcal{D}_g(\underline{x})$ be the Hessiam matrix of g, the $d \times d$ matrix having (i,j) entry equal to

$$\frac{\partial^2}{\partial x_i \partial x_j} g(\underline{x}).$$

Then, assuming that all entries of $\mathcal{D}_g(\underline{x})$ are continuous in a neighbourhood of x will lead to the following result

$$g(\underline{x} + \underline{\alpha}_n) = g(\underline{x}) + \underline{\alpha}_n'\mathcal{D}_g(\underline{x}) + \frac{1}{2}\underline{\alpha}_n'\mathcal{H}_g(\underline{x})\underline{\alpha}_n + o(\underline{\alpha}_n'\underline{\alpha}_n).$$

The following assuptions are made by Wand and Jones to be able to use the multivariate version of Taylor's theorem:

1. Each entry of $\mathcal{H}_f(.)$ is piecewise continuous and square integrable.
2. $\boldsymbol{H} = \boldsymbol{H}_n$ is a sequence of bandwidth matrices such that $n^{-1}|\boldsymbol{H}|^{-1/2}$ and all entries of $\boldsymbol{H}$ approach zero as $n \to \infty$. Also, assume the ratio of the largest and smallest eigenvalues of $\boldsymbol{H}$ is bounded for all n.

3. K is a bounded, compactly supported d-variate kernel satisfying

$$\int K(\underline{z})d\underline{z} = 1$$
$$\int \underline{z}K(\underline{z})d\underline{z} = 0$$
$$\int \underline{z}\underline{z}`K(\underline{z})d\underline{z} = \mu_2(K)\boldsymbol{I}$$

where $\mu_2(K) = \int z_i^2 K(\underline{z})d\underline{z}$ is independent of i. The AMISE can be expressed as

$$AMISE(\hat{f}(\underline{x})) = \int bias\left(\hat{f}(\underline{x})\right)^2 d\underline{x} + \int var\left(\hat{f}(\underline{x})\right) d\underline{x}$$

since the multivariate case is a generalization of the univariate case. The following matrix results is needed to be able to derive the asymptotic bias of the kernel estimator. Let $\boldsymbol{A} : d \times d$ be a square matrix then

$$tr(\boldsymbol{A}) = \sum_{i=1}^{d} a_{ii} \tag{1}$$

$$tr(\boldsymbol{AB}) = tr(\boldsymbol{BA})$$

whenever both matrix products are defined. The meaning of$vec\boldsymbol{A}$ is equal to stacking the columns of $\boldsymbol{A}$ underneath each other in order from left to right and $vech\boldsymbol{A}$ is obtained from $vec\boldsymbol{A}$ by eliminating all the above-diagonal elements of $\boldsymbol{A}$. For example in the bivariate case

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix}$$

$$vec(\boldsymbol{A}) = (a_{11},\, a_{12},\, a_{12},\, a_{22})`$$

$$vech(\boldsymbol{A}) = (a_{11},\, a_{12},\, a_{22})`.$$

If $\boldsymbol{A}$is symmetric then $vech\boldsymbol{A}$ contains each of the distinct elements of $\boldsymbol{A}$ and $vec\boldsymbol{A}$contains the elements of $vech\boldsymbol{A}$ with some duplicates. $\boldsymbol{D}_d : d^2 \times \frac{1}{2}d(d+1)$ is called the duplication matrix of order d. It is a unique matrix of zeros and ones such that

$$\boldsymbol{D}_d vech\boldsymbol{A} = vec\boldsymbol{A}. \tag{2}$$

For example if $d = 2$ then the duplication matrix of order 2 will be

$$\boldsymbol{D}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The following result holds for all square matrices $\boldsymbol{A}$

$$\boldsymbol{D}_d' vec(\boldsymbol{A}) = vech(\boldsymbol{A} + \boldsymbol{A}' - dg\boldsymbol{A}) \tag{3}$$

where $dg\boldsymbol{A}$ has the same diagonal elements as $\boldsymbol{A}$ and all the non-diagonal elements of the matrix $dg\boldsymbol{A}$ is equal to zero. For example in the bivariate case

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix}$$

$$dg\boldsymbol{A} = \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix}.$$

A useful result is

$$tr(\boldsymbol{A}'\boldsymbol{B}) = vec(\boldsymbol{A})' vec(\boldsymbol{B}). \tag{4}$$

Suppose $\boldsymbol{A} : d \times d$ is a invertible matrix then for linear changes of variables when integrating over $\mathbb{R}$

$$\int g(\boldsymbol{A}\underline{x}) d\underline{x} = |\boldsymbol{A}| \int g(\underline{y}) d\underline{y}. \tag{5}$$

Using the above matrix results and the multivariate version of Taylor's theorem Wand and Jones [24]

derived expressions for the asymptotic bias and variance of the kernel estimator

$$
\begin{aligned}
E\left[\hat{f}(\underline{x})\right] &= (K_{\boldsymbol{H}} * f)(\underline{x}) \\
&= \int K_{\boldsymbol{H}}(\underline{x}-\underline{y})f(\underline{y})d\underline{y} \\
&= \int |\boldsymbol{H}|^{-1/2}K(\boldsymbol{H}^{-1/2}(\underline{x}-\underline{y}))f(\underline{y})d\underline{y} \\
&= \int K(\underline{z})f(\underline{x}-\boldsymbol{H}^{1/2}\underline{z})d\underline{z} \\
&= \int K(\underline{z})\{f(\underline{x}) - (\boldsymbol{H}^{1/2}\underline{z})`\mathcal{D}_f(\underline{x}) + \frac{1}{2}(\boldsymbol{H}^{1/2}\underline{z})`\mathcal{H}_f(\underline{x})(\boldsymbol{H}^{1/2}\underline{z})\}d\underline{z} + o\{tr(\boldsymbol{H})\} \\
&= \int K(\underline{z})f(\underline{x})d\underline{z} - \int \underline{z}`\boldsymbol{H}^{1/2}\mathcal{D}_f(\underline{x})K(\underline{z})d\underline{z} + \frac{1}{2}\int \underline{z}`\boldsymbol{H}^{1/2}\mathcal{H}_f(\underline{x})\boldsymbol{H}^{1/2}\underline{z}K(\underline{z})d\underline{z} + o\{tr(\boldsymbol{H})\} \\
&= f(\underline{x}) + 0 + \frac{1}{2}tr\{\boldsymbol{H}^{1/2}\mathcal{H}_f(\underline{x})\boldsymbol{H}^{1/2}\int \underline{z}\underline{z}`K(\underline{z})d\underline{z}\} + o\{tr(\boldsymbol{H})\} = f(\underline{x}) + \frac{1}{2}\mu_2(K)tr\{\boldsymbol{H}\mathcal{H}_f(\underline{x})\} + o\{tr(\boldsymbol{H})\} \\
&= f(\underline{x}) + 0 + \frac{1}{2}tr\{\boldsymbol{H}^{1/2}\mathcal{H}_f(\underline{x})\boldsymbol{H}^{1/2}\int \underline{z}\underline{z}`K(\underline{z})d\underline{z}\} + o\{tr(\boldsymbol{H})\} \\
&= f(\underline{x}) + \frac{1}{2}\mu_2(K)tr\{\boldsymbol{H}\mathcal{H}_f(\underline{x})\} + o\{tr(\boldsymbol{H})\}
\end{aligned}
$$

$$
\begin{aligned}
bias\left(\hat{f}(\underline{x})\right) &= E\left[\hat{f}(\underline{x})\right] - f(\underline{x}) \\
&\approx \frac{1}{2}\mu_2(K)tr\{\boldsymbol{H}\mathcal{H}_f(\underline{x})\}
\end{aligned}
$$

$$
\begin{aligned}
var\left(\hat{f}(\underline{x})\right) &= n^{-1}\left\{(K_{\boldsymbol{H}}^2 * f)(\underline{x}) - (K_{\boldsymbol{H}} * f)(\underline{x})^2\right\} \\
&= n^{-1}\left\{\int K_{\boldsymbol{H}}(\underline{x}-\underline{y})^2 f(\underline{y})d\underline{y} - \left[\int K_{\boldsymbol{H}}(\underline{x}-\underline{y})f(\underline{y})d\underline{y}\right]^2\right\} \\
&= n^{-1}\left\{\int |\boldsymbol{H}|^{-1}K(\boldsymbol{H}^{-1/2}(\underline{x}-\underline{y}))^2 f(\underline{y})d\underline{y} - \left[\int |\boldsymbol{H}|^{-1/2}K(\boldsymbol{H}^{-1/2}(\underline{x}-\underline{y}))f(\underline{y})d\underline{y}\right]^2\right\} \\
&= n^{-1}\left\{|\boldsymbol{H}|^{-1/2}\int K(\underline{z})^2 f(\underline{x}-\boldsymbol{H}^{1/2}\underline{z})d\underline{z} - \left[\int K(\underline{z})f(\underline{x}-\boldsymbol{H}^{1/2}\underline{z})d\underline{z}\right]^2\right\} \\
&= n^{-1}|\boldsymbol{H}|^{-1/2}R(K)f(\underline{x}) + o(n^{-1}\boldsymbol{H}^{-1/2})
\end{aligned}
$$

$$
var(\hat{f}(\underline{x})) \approx n^{-1}|\boldsymbol{H}|^{-1/2}R(K)f(\underline{x})
$$

where $R(K) = \int K(\underline{z})^2 d\underline{z}$. By the integrability assumptions (assumption 1 and 3) Wand and Jones combined the above expressions to obtain the AMISE for the multivariate case

$$
\begin{aligned}
AMISE(\hat{f}(\underline{x})) &= \int bias\left(\hat{f}(\underline{x})\right)^2 d\underline{x} + \int var\left(\hat{f}(\underline{x})\right)d\underline{x} \\
&= \frac{1}{4}\mu_2(K)^2\int tr^2\{\boldsymbol{H}\mathcal{H}_f(\underline{x})\}d\underline{x} + n^{-1}|\boldsymbol{H}|^{-1/2}R(K).
\end{aligned}
$$

Duong also studied multivariate kernel density estimation and used the following notation

$$AMISE(\hat{f}(\underline{x})) = \frac{1}{4}\mu_2(K)^2 \int tr^2\{\boldsymbol{H}D^2 f(\underline{x})\}d\underline{x} + n^{-1}|\boldsymbol{H}|^{-1/2}R(K)$$

where $D^2 f(\underline{x})$ is the Hessian matrix of $f(\underline{x})$ [4]. This Hessian matrix is defined as a square matrix of second-order partial derivatives of $f(\underline{x})$. For example in the bivariate case the Hessian matrix of $f(x_1, x_2)$ i.e. $D^2 f(\underline{x})$ will be

$$D^2 f(\underline{x}) = \begin{pmatrix} \frac{\partial^2 f(x_1, x_2)}{\partial x_1 x_1} & \frac{\partial^2 f(x_1, x_2)}{\partial x_1 x_2} \\ \frac{\partial^2 f(x_1, x_2)}{\partial x_2 x_1} & \frac{\partial^2 f(x_1, x_2)}{\partial x_2 x_2} \end{pmatrix}.$$

Back to the expression by Wand and Jones they showed that $\int tr^2\{\boldsymbol{H}\mathcal{H}_f(\underline{x})\}d\underline{x}$ can be expanded using the matrix results (1) and (4) from above

$$\begin{aligned} \int tr^2\{\boldsymbol{H}\mathcal{H}_f(\underline{x})\}d\underline{x} &= \int vec(\boldsymbol{H})`vec(\mathcal{H}_f(\underline{x}))vec(\mathcal{H}_f(\underline{x}))`vec(\boldsymbol{H})d\underline{x} \\ &= \int vech(\boldsymbol{H})`\boldsymbol{D}_d`vec(\mathcal{H}_f(\underline{x}))vec(\mathcal{H}_f(\underline{x}))`\boldsymbol{D}_d vech(\boldsymbol{H})d\underline{x} \\ &= vech(\boldsymbol{H})`\boldsymbol{\Psi}_{\mathcal{F}} vech(\boldsymbol{H}) \end{aligned}$$

where $\boldsymbol{\Psi}_{\mathcal{F}} : \frac{1}{2}d(d+1) \times \frac{1}{2}d(d+1)$ is (using matrix result (3))

$$\boldsymbol{\Psi}_{\mathcal{F}} = \int vech\{2\mathcal{H}_f(\underline{x}) - dg\mathcal{H}_f(\underline{x})\} \times vech\{2\mathcal{H}_f(\underline{x}) - dg\mathcal{H}_f(\underline{x})\}`d\underline{x}.$$

Hence another expression for AMISE is given by

$$AMISE(\hat{f}(\underline{x})) = n^{-1}|\boldsymbol{H}|^{-1/2}R(K) + \frac{1}{4}\mu_2(K)^2 vech(\boldsymbol{H})`\boldsymbol{\Psi}_{\mathcal{F}} vech(\boldsymbol{H}).$$

The $\boldsymbol{\Psi}_{\mathcal{F}}$ matrix might look complicated but Wand and Jones found a simple formula to obtain the entries of $\boldsymbol{\Psi}_{\mathcal{F}}$ by using integration by parts [24]. Hence an expression for $\boldsymbol{\Psi}_{\mathcal{F}}$ can be explicitly stated in terms of its individual elements [4]. Consider the following notation let $\underline{r} = (r_1, r_2, \ldots, r_d)$ where the elements of this $\underline{r}$ vector are non-negative integers and let $|\underline{r}| = \sum_{i=1}^{d} r_i$ then the $\underline{r}$-th partial derivative of $f(\underline{x})$ (assuming the derivatives exists) can be written as

$$f^{(\underline{r})}(\underline{x}) = \frac{\partial^{|\underline{r}|}}{\partial x_1^{r_1} \partial x_2^{r_2} \ldots \partial x_d^{r_d}} f(\underline{x}).$$

Then Wand and Jones showed that

$$\int f^{(\underline{r})}(\underline{x}) f^{(\underline{r}')}(\underline{x})d\underline{x} = (-1)^{|\underline{r}|} \int f^{(\underline{r}+\underline{r}')}(\underline{x}) f(\underline{x})d\underline{x}$$

if $|\underline{r}+\underline{r}'|$ is even, and 0 otherwise. Using this, the authors discovered that each entry of $\boldsymbol{\Psi}_{\mathcal{F}}$ can be written in the form

$$\psi_{\underline{r}} = \int f^{(\underline{r})}(\underline{x}) f(\underline{x}) d\underline{x} = E\left[f^{(\underline{r})}(\underline{X})\right]$$

which is called the integrated density derivative functional by Duong and Hazelton [6] where $|\underline{r}|$ is even. For example consider the bivariate case then $\boldsymbol{\Psi}_{\mathcal{F}}$ will be given by

$$\boldsymbol{\Psi}_{\mathcal{F}} = \begin{bmatrix} \psi_{4,0} & 2\psi_{3,1} & \psi_{2,2} \\ 2\psi_{3,1} & 4\psi_{2,2} & 2\psi_{1,3} \\ \psi_{2,2} & \psi_{1,3} & \psi_{0,4} \end{bmatrix}.$$

Recall the expression for the AMISE used by Duong

$$AMISE(\hat{f}(\underline{x})) = \frac{1}{4}\mu_2(K)^2 \int tr^2\{\boldsymbol{H} D^2 f(\underline{x})\} d\underline{x} + n^{-1}|\boldsymbol{H}|^{-1/2} R(K).$$

An alternative expression for the AMISE is given by

$$AMISE(\hat{f}(\underline{x})) = n^{-1}R(K)|\boldsymbol{H}|^{-1/2} + \frac{1}{4}\mu_2(K)^2 (vech\boldsymbol{H})`\boldsymbol{\Psi}_4 (vech\, \boldsymbol{H})$$

since $\int_{R^d} tr^2(\boldsymbol{H} D^2 f(\underline{x})) d\underline{x} = (vech\boldsymbol{H})`\boldsymbol{\Psi}_4(vech\, \boldsymbol{H})$ under the conditions that all elements of $D^2 f(\underline{x})$ are piecewise continuous and squared integrable, $\boldsymbol{H} \to \boldsymbol{0}$ and $n^{-1}|\boldsymbol{H}|^{-1/2} \to \boldsymbol{0}$ as $n \to \infty$[4]. The matrix $\boldsymbol{\Psi}_4$ has dimensions $\frac{1}{2}d(d+1) \times \frac{1}{2}d(d+1)$ and note that the subscript 4 on $\boldsymbol{\Psi}$indicates the order of the derivatives involved

$$\boldsymbol{\Psi}_4 = \int_{R^s} vech(2D^2 f(\underline{x}) - dgD^2 f(\underline{x})) vech(2D^2 f(\underline{x}) - dgD^2 f(\underline{x}))` d\underline{x}.$$

In the univariate case a general explicit expression for the AMISE-optimal bandwidth exists but in the multivariate case it is not available but Wand and Jones showed in the case where $\boldsymbol{H}\epsilon\mathcal{D}$ and $\boldsymbol{H}\epsilon\mathcal{S}$ it is possible to write down an explicit expression [25]. Hence the bandwith matrix reduces to

$$\boldsymbol{H} = h^2 \boldsymbol{I}$$

and then the alternative expression of the AMISE describe by Wand and Jones reduces to

$$AMISE(\hat{f}(\underline{x})) = n^{-1}h^{-d}R(K) + \frac{1}{4}h^4\mu_2(K)^2 \int \{\nabla^2 f(\underline{x})\}^2 d\underline{x}$$

Multivariate kernel function	$K(\underline{x})$
Standard multivariate normal	$K(\underline{x}) = (2\pi)^{-d/2} exp(-\frac{1}{2}\underline{x}`\underline{x})$
Multivariate Epanechnikov	$K_e(\underline{x}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-\underline{x}`\underline{x}) & if\ \underline{x}`\underline{x} < 1 \\ 0 & otherwise \end{cases}$
$K_2(\underline{x})$	$K_2(\underline{x}) = \begin{cases} 3\pi^{-1}(1-\underline{x}`\underline{x})^2 & if\ \underline{x}`\underline{x} < 1 \\ 0 & otherwise \end{cases}$
$K_3(\underline{x})$	$K_3(\underline{x}) = \begin{cases} 4\pi^{-1}(1-\underline{x}`\underline{x})^3 & if\ \underline{x}`\underline{x} < 1 \\ 0 & otherwise \end{cases}$

Table 2: Some multivariate kernel functions

where

$$\nabla^2 f(\underline{x}) = \sum_{i=1}^{d} \left(\frac{\partial^2}{\partial x_i^2} \right) f(\underline{x})$$

the explicit expression for the AMISE-optimal bandwidth is then given by

$$h_{AMISE} = \left[\frac{dR(K)}{\mu_2(K)^2 \int \{\nabla^2 f(\underline{x})\}^2 d\underline{x}n} \right]^{1/(d+4)}.$$

Kernel theory

In Table 2 different multivariate kernel functions are given. For the multivariate Epanechnikov kernel function c_d represents the volume of the d-dimensional sphere [21]. Meaning for 1-dimensional $c_1 = 2$, for 2-dimensional $c_2 = \pi$, for 3-dimensional $c_3 = 4\pi/3$, etc. In the univariate case different kernel functions achieve very similar results for the MISE, which also holds for the multivariate case. Therefore the choice of the kernel function should be based on other considerations for example taking concern about the computational effort involved in calculating the kernel estimate. Also, note that the kernel estimator will inherit the smoothness properties of the selected kernel function [21]. The Epanechnikov kernel is the optimal kernel for only considering minimizing the MISE [21]. The most useful kernel functions in the bivariate case are $K_2(\underline{x})$ and $K_3(\underline{x})$ according to Silverman [21]. Since these kernel functions lead to kernel estimates which have higher order differentiability and in addition to this they can be calculated with less computational effort.

4.3 Optimal bandwidth matrix methods

Multivariate kernel density estimators involves far more mathematically and computationally aspects than univariate kernel density estimators [4]. Selecting a matrix and not just a single value for the bandwidth raises difficulties [4]. The bandwidth matrix brings about some important information about the orientation of the kernel function [4]. The parameterisation of the bandwidth matrix controls the type of orientation of the kernel function [4]. Wand and Jones [23] considered parameterisation for bivariate bandwidth matrices and classified them into three main classes of parameterisation which are summa-

Class 1	Symmetric, positive definite matrices	$\boldsymbol{H} = \begin{pmatrix} h_{11} & h_{12} \\ h_{12} & h_{22} \end{pmatrix}$
Class 2	Diagonal, positive definite matrices	$\boldsymbol{H} = \begin{pmatrix} h_{11} & 0 \\ 0 & h_{22} \end{pmatrix}$
Class 3	Positive constants times the identity matrix	$h^2\boldsymbol{I} = \begin{pmatrix} h^2 & 0 \\ 0 & h^2 \end{pmatrix}$

Table 3: Parameterisation for bivariate bandwidth matrices

rized in Table 3. Class 1 parameterisation for bandwidth matrices are also referred to as unconstrained bandwidth matrices and class 2 are also referred to as constrained bandwidth matrices. Duong [5] studied the performance of unconstrained and diagonal bandwidth matrices in the bivariate case. Where the simulation part of Duong's study included simulations of the 'dumbbell' density which is given by the following normal mixture

$$\frac{4}{11}N\left(\begin{bmatrix} -2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) + \frac{3}{11}N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.8 & -0.72 \\ 0.72 & 0.8 \end{bmatrix}\right) + \frac{4}{11}N\left(\begin{bmatrix} 2 \\ -2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

note that this density is unimodal. Duong found that the diagonal bandwidth matrix constrains the smoothing to be performed in directions parallel to the coordinate axes and therefore applies incorrect levels of smoothing to the diagonally oriented central portion of the density estimate and results in a bimodal density estimate [5]. Whereas the unconstrained bandwidth matrix produces a unimodal density estimate which is correct. Considering the study of Wand and Jones [23] and other papers Duong stated that the general conclusion from all these papers is that an unconstrained bandwidth matrix will yield remarkably improved performance for densities which has large probability mass oriented away from the coordinate axes (oriented diagonally to the coordinate axes), such as the dumbbell density [5].

Silverman's multivariate rule of thumb

Assume that the unknown density $f(\underline{x})$ has bounded and continuous second derivatives. Silverman investigated multivariate kernel density estimation by considering bandwidth matrices of class 3 parameterisation i.e. a positive constant times the identity matrix $h^2\boldsymbol{I}$. Hence the kernel estimator is defined as

$$\hat{f}(\underline{x}; h) = n^{-1}h^{-d}\sum_{i=1}^{n} K(h^{-1}(\underline{x} - \underline{X}_i)).$$

Silverman found that a closed form solution of the optimal bandwidths considering the AMISE is only available if $h_1 = h_2 = h_3 = \ldots = h_d = h$. Hence a closed form will exist if the following condition is met

$$h_{AMISE} = \left[\frac{dR(K)}{n\mu_2(K)^2 \int tr^2(D^2 f(\underline{x}))d\underline{x}}\right]^{1/(d+4)}. \tag{6}$$

The problem now remains that, the optimal bandwidth h_{AMISE} depends on the unknown density function being estimated [21]. If the unknown density being estimated is the unit d-variate normal density (let ϕ be the unit d-variate normal density) then it can be shown that

$$\int tr^2(D^2 f(\underline{x}))d\underline{x} = \int tr^2(D^2\phi(\underline{x}))d\underline{x} = (2\sqrt{\pi})^{-d}(\frac{1}{2}d + \frac{1}{4}d^2). \tag{7}$$

This value can be substituted back into (6)(cross-reference) which will give the optimal bandwidth for the smoothing of normally distributed data with unit variance [21]. Then Silverman's rule of thumb for the optimal bandwidth is given by

$$h_{opt} = A(K)n^{-1/(d+4)}$$

where the constant

$$A(K) = \left[\frac{dR(K)}{\mu_2(K)^2(2\sqrt{\pi})^{-d}(\frac{1}{2}d + \frac{1}{4}d^2)}\right]^{1/(d+4)}$$

depends on the kernel function [21].

Plug-in bandwidth methods

This section discusses some multivariate extensions of the popular univariate SJPI plug-in bandwidth method which was developed by Sheather and Jones as referred to in [20]. Firstly in the multivariate case, the plug-in bandwidth method requires pilot estimates of $\psi_{\underline{r}}$ (the integrated density derivative functionals) since it is the unknown factors in the AMISE. Recall that

$$\psi_{\underline{r}} = E\left[f^{(\underline{r})}(\underline{X})\right]$$

then a natural estimator of $\psi_{\underline{r}}$ are

$$\hat{\psi}_{\underline{r}}(\boldsymbol{G}) = n^{-1}\sum_{i=1}^{n}\hat{f}^{(\underline{r})}(\underline{X}_i) = n^{-2}\sum_{i=1}^{n}\sum_{j=1}^{n}K_{\boldsymbol{G}}^{(\underline{r})}(\underline{X}_i - \underline{X}_j)$$

where $\boldsymbol{G}$ is a pilot bandwidth matrix typically different from $\boldsymbol{H}$ [6]. An appropriate way for choosing the values for the matrix $\boldsymbol{G}$ is to consider $\boldsymbol{G} = g^2\boldsymbol{I}$ and pre-transformation of the data also called sphering the data. This refers to choosing $\boldsymbol{G} = g^2\boldsymbol{S}$ where $\boldsymbol{S}$ is the sample covariance matrix [24]. The remaining objective is to select an appropriate value for g and for this various methods exists. In the case of multivariate normal data sphering the data will be appropriate but there is no theory supporting sphering the data for estimation of general density shapes according to Wand and Jones [24] but it is sufficient for selecting the pilot bandwidth $\boldsymbol{G}$. Since it is not necessary to select $\boldsymbol{G}$ with the same accuracy as $\boldsymbol{H}$ [5]. This restriction makes it possible to derive an analytical expression for the optimal

pilot bandwidth avoiding the intensive computational effort involved [5]. When using a multivariate normal kernel function the plug-in estimate of the AMISE is given by

$$PI(\boldsymbol{H}) = n^{-1}(4\pi)^{-d/2}|\boldsymbol{H}|^{-1/2} + \frac{1}{4}(vech(\boldsymbol{H}))`\hat{\boldsymbol{\Psi}}_4(vech\boldsymbol{H})$$

which can be numerically minimized to give $\hat{\boldsymbol{H}}_{PI}$ (an estimate of the optimal plug-in bandwidth matrix) [5].

Wand and Jones [25] developed a multivariate extension of the plug-in method where the bandwidth matrices are restricted to be diagonal. According to Wand and Jones, it is impossible to derive an explicit expression for the optimal plug-in bandwidth matrix for general multivariate kernel estimators except in the bivariate case for a diagonal bandwidth matrix. This bandwidth method is derived from the principle of diagonal bandwidth matrices for bivariate density estimation where the analysis is more straightforward [25]. Wand and Jones selected g such that it would be the minimizer of the AMSE criterion. Further on the authors used the so-called "l-stage direct plug-in" bandwidth method for selecting appropriate values for the elements of the diagonal bandwidth matrix $\boldsymbol{H}$. A brief explanation of the "l-stage direct plug-in" bandwidth method can be found in [24] where a common choice for the number of stages is considered to be at least two.

The following plug-in method was developed by Duong and Hazelton [6] which selects a full bandwidth matrix and is based on the principle of bivariate kernel density estimation. These authors note that using this approach it may result in a bandwidth matrix with problems such as being positive definite or even almost singular therefore they used a single, common tuning parameter to optimize all the elements of the bandwidth matrix [6]. This ensures that the bandwidth matrix will be positive definite and also eases the computational effort. Duong and Hazelton selected g to be the estimate of the bandwidth which minimizes the sum of AMSE (the SAMSE criterion) for $\boldsymbol{\Psi}_4$

$$g = \underset{g>0}{agrmin}\, SAMSE(\hat{\boldsymbol{\Psi}}_4)$$

where

$$SAMSE(\hat{\boldsymbol{\Psi}}_4) = SAMSE_4(g) = \sum_{\underline{r}:|\underline{r}|=4} AMSE\hat{\psi}_{\underline{r}}(g).$$

The SAMSE criterion has better numerical and theoretical properties than the AMSE according to Duong [5]. There exists a closed form of the optimal pilot bandwidth using the SAMSE criterion. A brief discussion on the plug-in method developed by Duong and Hazelton can be found in [6].

Least squares cross-validation

LSCV is defined as

$$LSCV(\boldsymbol{H}) = \int_{R^d} \hat{f}(\underline{x}; \boldsymbol{H})^2 d\underline{x} - 2n^{-1} \sum_{i=1}^{n} \hat{f}_{-i}(\underline{X}_i; \boldsymbol{H})$$

where the estimator $\hat{f}_{-i}(\underline{x}; \boldsymbol{H})$ is called the leave-one-out estimator and is defined in [5] as

$$\hat{f}_{-i}(\underline{x}; \boldsymbol{H}) = (n-1)^{-1} \sum_{\substack{j=1 \\ j \neq i}}^{n} K_{\boldsymbol{H}}(\underline{x} - \underline{X}_j).$$

The bandwidth matrix of the kernel estimator is then selected to be

$$\hat{\boldsymbol{H}}_{LSCV} = argmin_{\boldsymbol{H} \epsilon \mathcal{F}} LSCV(\boldsymbol{H})$$

as showed in [24]. Duong also easily proofed that $E[LSCV(\boldsymbol{H})] = MISE(\boldsymbol{H}) - \int_{R^d} f(\underline{x})^2 d\underline{x}$ and therefore the LSCV estimates the MISE directly [5]. The choice of the kernel function is not crucial. Using the multivariate standard normal kernel function the LSCV can be rewritten as

$$LSCV(\boldsymbol{H}) = n^{-1}(4\pi)^{-d/2}|\boldsymbol{H}|^{-1/2} + n^{-1}(n-1)^{-1} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} (K_{2\boldsymbol{H}} - 2K_{\boldsymbol{H}})(\underline{X}_i - \underline{X}_j).$$

Biased cross-validation

Sain, Baggerly, and Scott [17] derived two multivariate version of the univariate biased cross-validation bandwidth method. For the development of these two methods Sain, Baggerly, and Scott only considered diagonal bandwidth matrices for the multivariate normal kernel estimators. Note that the multivariate normal density function was used as the kernel function. These biased cross-validation methods and the plug-in method discussed above are related in the sense of depending on an estimator of $\psi_{\underline{r}}$ [5]. A great difference is that for the biased cross-validation methods the pilot bandwidth matrix $\boldsymbol{G}$ will be set equal to the bandwidth matrix for the kernel estimator $\boldsymbol{H}$ but for the plug-in method these two matrices are completely independent of one another [5]. The principle on which Sain, Baggerly, and Scott derived these biased cross-validation methods were treating the estimator of $\psi_{\underline{r}}$ as a function of h and hence the unknown part of the AMISE criterion have been solved then $\hat{h}$ (the estimate of the bandwidth which will be used in the multivariate normal kernel estimator) is selected to be the minimizer of the AMISE criterion [17]. Recall from above sphering the data (in the case of using $\boldsymbol{H} = h^2\boldsymbol{S}$) is correct when working with multivariate normal data but there is no theory supporting sphering the data for estimation of general density shapes [24].

These two versions of the biased cross-validation method are different in the sense of using two different estimators for $\psi_{\underline{r}}$. The full derivation of these two methods can be found in [17]. The two different estimators for $\psi_{\underline{r}}$ is given by

$$\hat{\psi}_{\underline{r}}(\boldsymbol{H}) = n^{-2} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} K_{2\boldsymbol{H}}^{(\underline{r})}(\underline{X}_i - \underline{X}_j)$$

$$\tilde{\psi}_{\underline{r}}(\boldsymbol{H}) = n^{-1} \sum_{i=1}^{n} \hat{f}_{-i}^{(\underline{r})}(\underline{X}_i; \boldsymbol{H}) = n^{-1}(n-1)^{-1} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} K_{\boldsymbol{H}}^{(\underline{r})}(\underline{X}_i - \underline{X}_j).$$

Let $\hat{\boldsymbol{\Psi}}_4$ and $\tilde{\boldsymbol{\Psi}}_4$ be the estimates of the matrix$\boldsymbol{\Psi}_4$where the estimates $\hat{\psi}_{\underline{r}}$ and $\tilde{\psi}_{\underline{r}}$have been substituted respectively. Then the two versions of the biased cross-validation method follow as

$$BCV1(\boldsymbol{H}) = n^{-1}(4\pi)^{-d/2}|\boldsymbol{H}|^{-1/2} + \frac{1}{4}\mu_2(K)^2(vech(\boldsymbol{H}))'\hat{\boldsymbol{\Psi}}_4(vech(\boldsymbol{H}))$$

$$BCV2(\boldsymbol{H}) = n^{-1}(4\pi)^{-d/2}|\boldsymbol{H}|^{-1/2} + \frac{1}{4}\mu_2(K)^2(vech(\boldsymbol{H}))'\tilde{\boldsymbol{\Psi}}_4(vech(\boldsymbol{H})).$$

The estimates $\hat{\boldsymbol{H}}_{BCV1}$and $\hat{\boldsymbol{H}}_{BCV2}$ of the bandwidth matrix of the multivariate kernel estimator are the minimizers of the $BCV1(\boldsymbol{H})$ and $BCV2(\boldsymbol{H})$ functions respectively [5].

Minimum leave-one-out entropy (MLE) bandwidth estimator

Van der Walt broadly categorised conventional kernel bandwidth estimators into rule-of-thumb, least-squares cross-validation (LSCV), likelihood CV (LCV), and plug-in methods. The motivation for this bandwidth estimation method was that conventional kernel bandwidth estimators were not developed for the purpose of high-dimensional density estimation problems as often encountered in pattern recognition [22]. Discussion of this method further on will refer to Van der Wald [22]. This kernel bandwidth estimation method was derived by optimizing the LOUT ML objective function (criterion) with respect to the kernel bandwidth matrix and then finding an optimal solution in a maximum likelihood sense. The ML objective function is often referred to as maximizing the log-likelihood function where this function is defined as

$$l_{\boldsymbol{H}}(\underline{X}) = \sum_{i=1}^{n} log(\hat{f}(\underline{X}_i; \boldsymbol{H})).$$

The following expression for the LOUT ML objective function

$$l_{\boldsymbol{H}(-i)}(\underline{X}) = \sum_{i=1}^{n} log(\hat{f}_{-i}(\underline{X}_i; \boldsymbol{H}))$$

was designed to prevent the trivial solution of the ML objective function where $l_{\boldsymbol{H}}(\underline{X}) = \infty$when $\boldsymbol{H} = 0$. Sample entropy is defined as

$$H(\underline{X}) = -\frac{1}{n}\sum_{i=1}^{n} log(\hat{f}(\underline{X}_i; \boldsymbol{H})).$$

Substituting the log-likelihood function in the sample entropy results in

$$H(\underline{X}) = -\frac{l_{\boldsymbol{H}}(\underline{X})}{n}.$$

This shows that the sample entropy can be expressed in terms of the log-likelihood function. Hence maximizing the log-likelihood function for the ML objective function are equivalent to minimizing the sample entropy. This is important since this result shows that deriving kernel bandwidth estimators that maximise the ML objective function is equivalent to deriving kernel bandwidth estimators that minimise the sample entropy.

If a multivariate Gaussian density function with a full covariance matrix is chosen for the kernel function

$$K_{\boldsymbol{H}_k}(\underline{x}_i - \underline{x}_k) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{H}_k|^{1/2}} exp\left(-\frac{1}{2}(\underline{x}_i - \underline{x}_k)'\boldsymbol{H}_k^{-1}(\underline{x}_i - \underline{x}_k)\right)$$

the optimal bandwidth matrix for the MLE estimator will be

$$\boldsymbol{H}_{MLE(i);k} = \frac{\sum_{i=1}^{n} \frac{(\underline{x}_i - \underline{x}_k)(\underline{x}_i - \underline{x}_k)' K_{\boldsymbol{H}_k}(\underline{x}_i - \underline{x}_k)}{\hat{f}_{-i}(\underline{x}_i; \boldsymbol{H})}}{\sum_{i=1}^{n} \frac{K_{\boldsymbol{H}_k}(\underline{x}_i - \underline{x}_k)}{\hat{f}_{-i}(\underline{x}_i; \boldsymbol{H})}}$$

Hence the matrix $\hat{\boldsymbol{H}}_{MLE(i);k}$ is an estimate of the optimal bandwidth for the MLE estimator with a multivariate Gaussian density function with a full covariance matrix as the kernel function centered on data point $\underline{x}_k$. There are $D(D+1)/2$ bandwidth parameters to be estimated, for each of the n symmetric full covariance matrices. (maak net seker oor hierdie)

A multivariate Gaussian density function with diagonal covariance matrix can be expressed as the product of univariate Gaussian density functions

$$K_{\boldsymbol{H}_j}(\underline{x}_i - \underline{x}_j) = \prod_{p=1}^{D} K_{h_{jp}}(x_{ip} - x_{jp})$$

where x_{ip} is the value of the data point $\underline{x}_i$ in dimension p, similar forx_{jp} and h_{jp} is the bandwidth of the univariate Gaussian kernel centred on the data point $\underline{x}_j$ in dimension p. The diagonal bandwidth matrix $\boldsymbol{H}_j$ is such that $\boldsymbol{H}_{j(p,p)} = h_{jp}^2$. If a multivariate Gaussian density function with diagonal covariance

matrix is chosen for the kernel function, the MLE estimator will have the following optimal bandwidth

$$h_{kd}^2 = \frac{\sum_{i=1}^{n} \frac{K_{h_{kd}}(x_{id}-x_{kd})(x_{id}-x_{kd})^2}{\hat{f}_{-i}(\underline{x}_i;\boldsymbol{H})} \prod_{p\neq d} K_{h_{kp}}(x_{ip}-x_{kp})}{\sum_{i=1}^{n} \frac{K_{h_{kd}}(x_{id}-x_{kd})}{\hat{f}_{-i}(\underline{x}_i;\boldsymbol{H})} \prod_{p\neq d} K_{h_{kp}}(x_{ip}-x_{kp})}.$$

Substituting the definition of a multivariate Gaussian density function results in

$$\boldsymbol{H}_{k(d,d)} = \frac{\sum_{i=1}^{n} \frac{K_{\boldsymbol{H}_k}(\underline{x}_i-\underline{x}_k)(x_{id}-x_{kd})^2}{\hat{f}_{-i}(\underline{x}_i;\boldsymbol{H})}}{\sum_{i=1}^{n} \frac{K_{\boldsymbol{H}_k}(\underline{x}_i-\underline{x}_k)}{\hat{f}_{-i}(\underline{x}_i;\boldsymbol{H})}}$$

where $\boldsymbol{H}_{MLE(ii);k}$ is the diagonal bandwidth matrix such that $\boldsymbol{H}_{k(d,d)} = h_{kd}^2$. Hence the matrix $\hat{\boldsymbol{H}}_{MLE(ii);k}$ is an estimate of the optimal bandwidth for the MLE estimator with a multivariate Gaussian kernel with diagonal covariance matrix as the kernel function centered on the data point $\underline{x}_k$. There are D bandwidth parameters to be estimated for each of the n diagonal covariance matrices. Hence there are a total number of nD parameters to be estimated using the MLE bandwidth estimation method choosing a multivariate Gaussian kernel with a diagonal covariance matrix, which is less than the total number of parameters $D(D+1)/2$ to be estimated for the full covariance matrix case. Note that assuming a diagonal covariance matrix is not the same as assuming independence between the variables. Kernel density estimators with diagonal bandwidth matrices are able of modelling correlation to some degree, since they are centred on the data points $i = 1, 2, \ldots, k, \ldots, n$ and if the data points vary together between dimensions, the kernel density estimate will capture the covariance.

Van der Wald also developed an MLE bandwidth estimator that estimates an identical diagonal bandwidth matrix for all kernels as opposed to where he originally derived a unique bandwidth matrix per kernel. This estimator is called the global MLE estimator since it estimates a single diagonal bandwidth matrix used globally. The leave-one-out estimator $\hat{f}_{-i}(\underline{x};\boldsymbol{H}_g)$ is now defined differently as

$$\hat{f}_{-i}(\underline{x};\boldsymbol{H}_g) = (n-1)^{-1} \sum_{\substack{j=1 \\ j\neq i}}^{n} K_{\boldsymbol{H}_g}(\underline{x}-\underline{X}_j)$$

where $\boldsymbol{H}_g$ is the diagonal bandwidth matrix that is identical for all kernels. Choosing a Gaussian kernel

with diagonal covariance matrix $\boldsymbol{H}_g$results in the estimate

$$\hat{\boldsymbol{H}}_{g(d,d)} = \frac{\sum_{i=1}^{n} \frac{\sum_{j \neq i} K_{\boldsymbol{H}_g}(x_{id} - x_{jd})(x_{id} - x_{kd})^2}{\hat{f}_{-i}(\underline{x}_i; \boldsymbol{H}_g)}}{\sum_{i=1}^{n} \frac{\sum_{j \neq i} K_{\boldsymbol{H}_g}(x_{id} - x_{jd})}{\hat{f}_{-i}(\underline{x}_i; \boldsymbol{H}_g)}}.$$

For a furtherer discussion on this method and other bandwidth estimation methods using the ML criterion refer to [22].

4.4 Comparison

Duong did a comparison study of different bandwidth matrices in the bivariate case [5]. The plug-in, LSCV, BCV1, BCV2 and Smoothed cross-validation (SCV) bandwidth methods were considered in the study. The multivariate normal density function was selected to be the kernel function. Smoothed cross-validation was first introduced by Hall, Marron, and Park [8] in the univariate case. Duong and Hazelton [7] generalised the SCV methodology for the multivariate case and declared that the SCV method for full bandwidth matrices is the most reliable cross-validation method amongst these they had studied. In the comparison study of Duong unconstrained and constrained bandwidth matrices were considered where it was possible. The conclusion was that unconstrained bandwidth matrices will produce a better density estimate than there diagonal counterparts when the data have large mass oriented diagonally to the coordinate axes as referred to above. Duong's general recommendations were the SCV and the 2-stage plug-in bandwidth method as defined by Wand and Jones in [25]. This also supports the statement of Duong and Hazelton [7] that the SCV method is reasonably comparable to the best plug-in methods currently available in the bivariate case. Another conclusion of Duong's study was that the LSCV bandwidth method is useful in some cases but the performance of this bandwidth method is highly variable [5].

Sain, Baggerly, and Scott [17] performed a comparison study between the LSCV, BCV1, BVC2, and bootstrap bandwidth methods. This study was mainly between cross-validation bandwidth methods. For a complete description of the bootstrap bandwidth method see [17]. The study considered asymptotic results, simulations and real-life data examples in dimensions 1, 2 and 3. The overall results found, suggested that the BCV2 method had performed the best. The LSCV method is inconsistent meaning sometimes producing good estimates and sometimes producing seriously undersmoothed estimates. Sain, Baggerly, and Scott stated that in their experience the LSCV method has a tendency to choose very small bandwidths even when the size of the dataset is larger than a thousand.

Referring now to a comparison study done by van der Walt [22] where the performance of bandwidth methods in higher dimensions (higher than 10 dimensions), were considered. Plug-in methods were not considered in this study since the performance is difficult to predict in higher dimensions [22]. Silverman's rule of thumb, MSP, ML Gauss, MLE(i), LLCV, LSCV, and ULCV were the bandwidth methods considered in this study. A clear definition of each of these bandwidth methods can be found in [22]. The ML Gauss method main focus is to calculate the covariance matrix and the MSP bandwidth method assumes a normal reference rule like Silverman's rule of thumb. A number of simulated data sets and real-world data sets were used in this study. Datasets which are representative of the samples sizes and dimensionalities of typical pattern recognition problems were selected. The highest dimension considered in these datasets was 617.

Van der Walt selected a multivariate normal density function as the kernel function. A diagonal bandwidth matrix was considered for dimensions larger than 10 and an unconstrained bandwidth matrix was considered for dimensions between 1 and 10. Table 4, 5 and 6 classifies which bandwidth estimators perform well in which class of dimensions according to the results found by Van der Walt. Considering the tests done using unimodal simulated data Van der Walt found that the ML Gauss estimator generally performed the best. The estimators LLCV and LSCV performance severely decreased as dimensions increased beyond 10. On the bimodal simulated data, the MLE(i) estimator's performance decreased as the dimensions decreased. Considering higher dimensions on the bimodal simulated data resulted in the ML Gauss estimator performing the best although the ML Gauss estimator assumes a unimodal distribution. Van der Walt stated that this superior performance of the ML Gauss estimator might change if the distance between the means of the two modes is increased. At higher dimensions, Silverman's rule of thumb generally outperforms the MSP estimator. Using real-world data the LLCV, LSCV and ULCV estimators' performance decreased as the dimensions increased where the MLE(i) and ML Gauss estimators' performance increased as the dimensions increased. Silverman and the MSP estimator performed well across all dimensionalities and were generally very competitive. Taking all the results together found is this comparative study leads to the conclusion that the LLCV, LSCV, and ULCV estimators are not suitable for dimensions higher than 10. The MLE(i) and ML Gauss estimators perform better for dimensions higher than 10. Silverman's rule of thumb and the MSP estimator consistently perform well across all dimensions.

All dimensions	Lower dimensions	Higher dimensions
ML Gauss ULCV Silverman MSP MLE(i)	LLCV LSCV	

Table 4: Performance of estimators using unimodal simulated data

All dimensions	Lower dimensions	Higher dimensions
Silverman MSP	LLCV LSCV ULCV	ML Gauss MLE(i)

Table 5: Performance of estimators using bimodal simulated data

5 Application

5.1 Univariate

The objective of this section is to show differences in the kernel estimates using different bandwidth estimation methods and to confirm some of the bandwidth method comparisons done by previous researchers. Throughout this section, the Gaussian kernel function will be used. The graphs in Figure 1,2, 3 and 4 were obtained using the package "ks" in the software program R [14]. In these figures different kernel estimates were graphed by using different bandwidth methods. In Figure 1 different kernel estimates (using the Gaussian kernel function but different bandwidths) were applied to a simulated sample from the standard normal distribution. The bandwidth method "nrd0" refers to Silverman's rule of thumb, "ucv" refers to least squares (unbiased) cross-validation, "bcv" refers to biased cross-validation and "SJ-ste" refers to the SJPI plug-in method of Sheater and Jones [20] where the fixed point equation

$$h = \left[\frac{R(K)}{nR(f''_{g(h)})(\int x^2 K)^2}\right]^{1/5}$$

is solved by using the function "uniroot". This function searches the interval from lower to upper for a root of the fixed point equation.

In Figure 1 the different bandwidth methods produced approximately the same density estimates except the "ucv" bandwidth method produced a density estimate with some spurious features (undersmoothed) in this case the bandwidth may be too small. Jones stated in [10] that the least squares

All dimensions	Lower dimensions	Higher dimensions
Silverman MSP	LLCV LSCV ULCV	ML Gauss MLE(i)

Table 6: Performance of estimators using real-world data

(unbiased) cross-validation method is unreliable since it is too variable particularly in the direction of undersmoothing and this statement is supported by the results presented in Figure 1. Otherwise, all the methods considered performed well when the underlying structure of the data is standard normal. In Figure 2 different kernel estimates were applied to a simulated sample from a gamma distribution with shape parameter 1 and scale parameter 1. Similarly in Figure 3 different kernel estimates were applied to a simulated sample from an exponential distribution with parameter 1. Observing the graphs in Figure 2 and 3 it seems Silverman's rule of thumb has performed the best with respect to producing a smooth density estimate. The gamma and exponential distributions only exist for positive values including zero, as mentioned above in the introduction, bounded data can cause problems for kernel density estimation. Kernel density estimates will often go beyond the bounds of the data and then the estimate is considerable bias at and near the bounds of the data [11]. Silverman's rule of thumb results in an estimate which is considerable bias at 0. The gamma and the exponential distributions also have a long tail to the right which can cause problems for kernel estimation. Note that there is a difference between a distribution with long tails and a distribution with heavy tails. A distribution can have both but when a distribution has long tails it does not necessarily have heavy tails and vice versa. Examples of distributions with heavy tails include the Pareto, Weibull, and Cauchy distributions. As stated above in the introduction and shown there will appear spurious (false) noise in the tails of the estimate if the underlying distribution of the data has long tails because the bandwidth is fixed across the entire sample. This is shown in Figure 2 and 3. Silverman's rule of thumb led to an estimate which has the least spurious noise in the tail where the other estimates have considerable spurious noise in the tails. Also note that the density estimates produced by the bandwidth methods "ucv", "bcv" and "SJ-ste" have relative larger magnitudes at the mode when compared to the density estimate produced by Silverman's rule of thumb.

A beta distribution will be bimodal if both parameters are smaller than 1. In Figure 4 different kernel estimates were applied to a simulated sample from a beta distribution with both shape parameters equal to 0.5 (the magnitude of the density function at both modes will be equal). In Figure 5 different kernel estimates were applied to a simulated sample from a beta distribution with shape parameters equal to $\alpha = 0.5$ and $\beta = 0.75$ (the magnitude of the density function at one of the modes is larger). As mentioned above data from a multimodal distribution may cause some problems. Since it is difficult to find a single bandwidth which will adequately differentiate between distinct peaks and valleys between the peaks [16]. Choosing a bandwidth which is too large can erase modes which were significant and choosing a bandwidth which is too small may introduce spurious peaks by undersmoothing. In Figure 4 and 5, the bandwidth methods "ucv" and "SJ-ste" perform the same. The following observations are drawn from Figure 4 and 5. The densities produced by "ucv" and "SJ-ste" has a lot of spurious noise between the two modes i.e. the bandwidth is too small. Note that the densities produced by "bcv" and

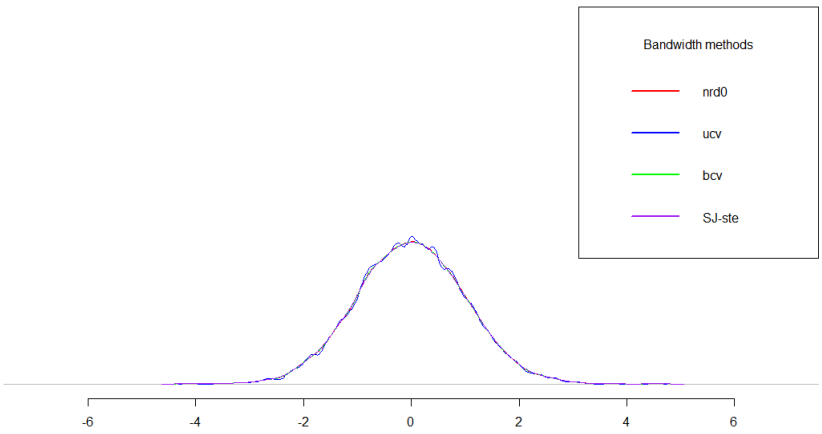


Figure 1: Kernel estimates applied to a simulated sample from a standard normal distribution

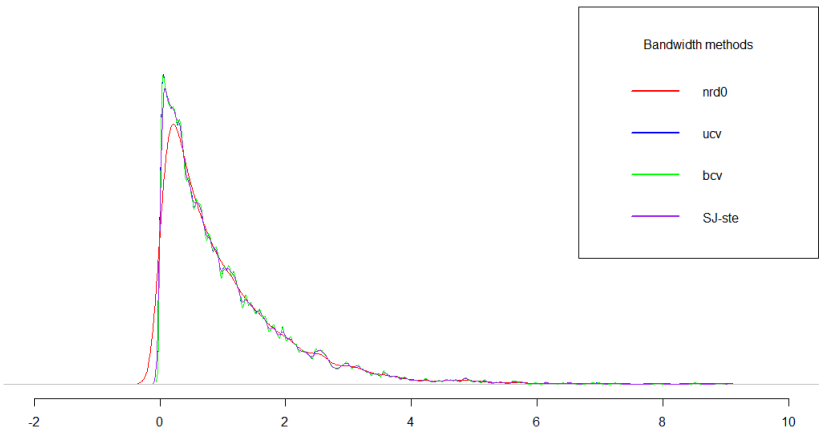


Figure 2: Kernel estimates applied to a simulated sample from a gamma distribution with shape parameter 1 and scale parameter 1

Silverman's rule of thumb are a lot smoother. The density estimates produced by "ucv" and "SJ-ste" have a relatively larger magnitude at the modes compared to the estimates produced by "bcv" and Silverman's rule of thumb. These beta distributions from which the samples were drawn are also bounded between 0 and 1. Silverman's rule of thumb and "bcv" produced estimates which are considerable bias at and near 0 and 1 where the estimates produced by "ucv" and "SJ-ste" are less bias at and near the bounds. In Figure 4 "bcv" and Silverman's rule of thumb perform very similarly (produced estimates which generally look the same) but in Figure 5 they perform less similar.

Testing kernel estimates on real data are very important since it will indicate how well the kernel estimation performs in practice [10]. The first real dataset on which kernel estimation will be performed

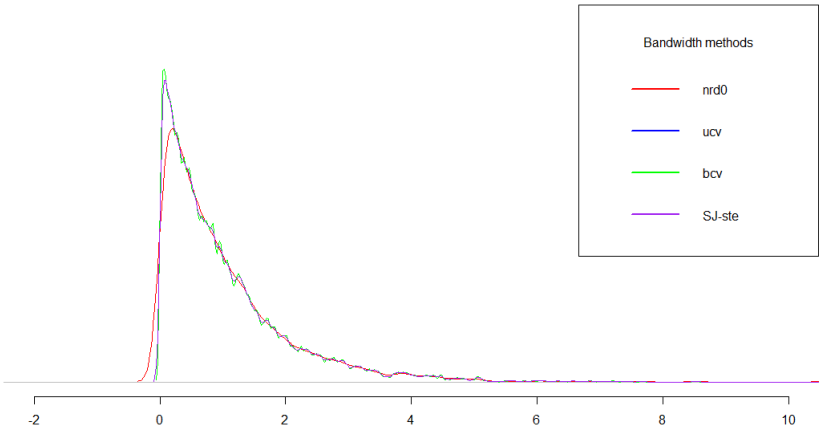


Figure 3: Kernel estimates applied to a simulated sample from a exponential distribution with parameter 1

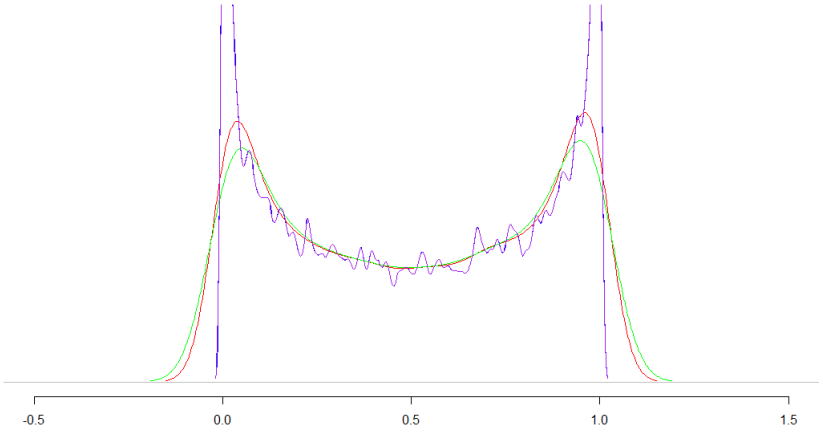


Figure 4: Kernel estimates applied to a simulated sample from a beta distribution with both shape parameters equal to 0.5

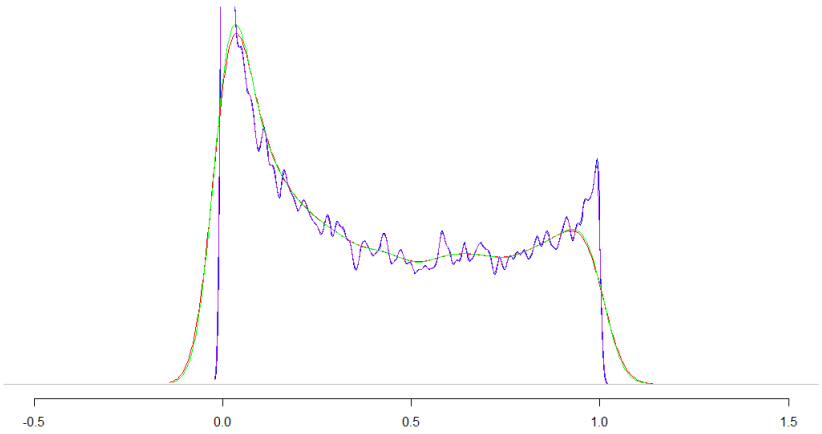


Figure 5: Kernel estimates applied to a simulated sample from a beta distribution with $\alpha = 0.5$ and $\beta = 0.75$

is the Old Faithful dataset as referred to above. Using the waiting variable and a sample size of 272. The waiting variable refers to recorded times between consecutive eruptions of the Old Faithful geyser. Another dataset that will be used is secondary school students' Mathematics and Portuguese scores (a score out of 20). Kernel estimation will be performed on the Mathematics scores using a sample of 200 students [1]. The first graphs in Figure 6 up until Figure 9 are produced by using the KDE procedure in the software program SAS and the second graphs are produced by using the "ks" package in the software program R [14]. The KDE procedure in SAS performs univariate and multivariate kernel density estimation using the Gaussian kernel function. In the KDE procedure, there is multiple bandwidth selection methods available in the univariate case. Selecting the SJPI plug-in method the KDE procedure solves the fixed point equation

$$h = \left[\frac{R(K)}{nR(f''_{g(h)})(\int x^2 K)^2} \right]^{1/5}$$

where $R(K) = \int K(x)^2 dx$. This procedure uses a bisection algorithm which can be described as a root-finding method that repeatedly divides an interval into two equal parts and then chooses a subinterval in which a root should be for furtherer processing. The starting values of this bisection algorithm are selected to be the two largest values from a grid of values on a log scale that bound a solution of the fixed point equation. A bisection algorithm is simple and robust but it is also relatively slow.

Looking at these graphs it seems that Silverman's rule of thumb is producing density estimates which are oversmoothed. Jones [10] stated that using Silverman's rule of thumb often lead to density estimates which are extremely oversmoothed leading to missing important features. For both of these datasets, the SJPI bandwidth method performed the best considering the graphs. Note that the histograms do

[1] Datasets used can be found on Kaggle (https://www.kaggle.com/datasets).

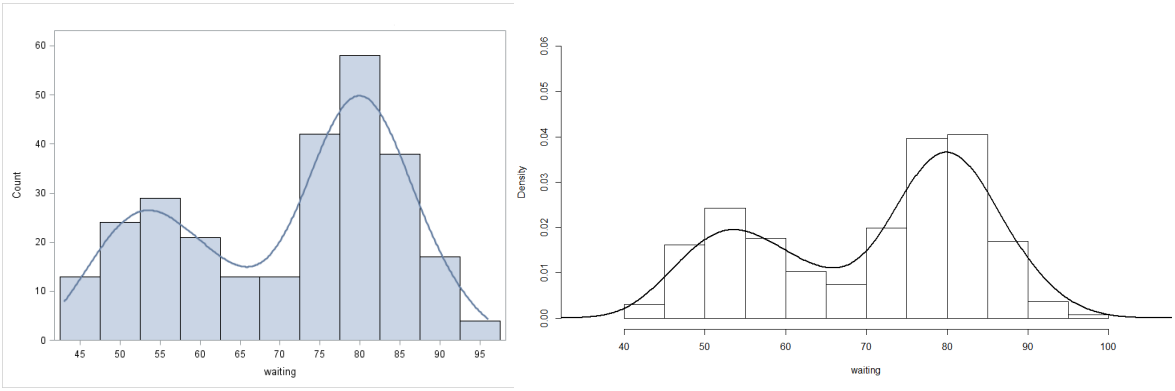


Figure 6: Kernel estimate using Silverman's rule of thumb applied to the Old faithful geyser dataset

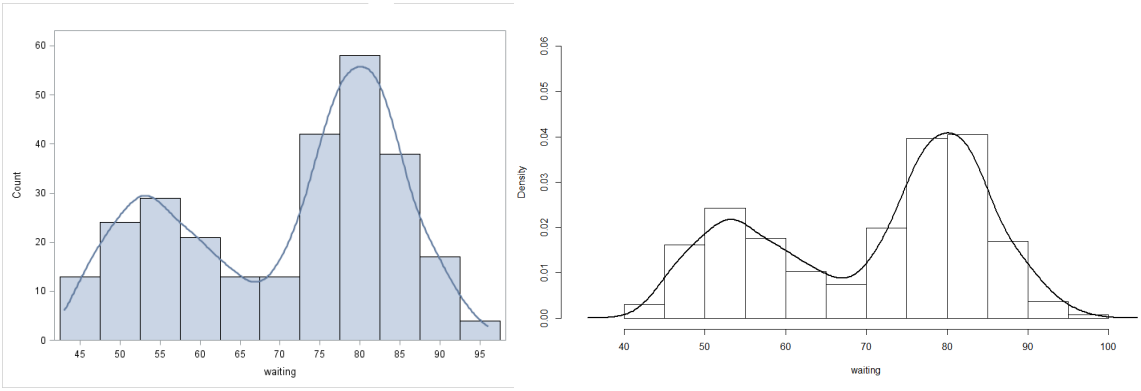


Figure 7: Kernel estimate using SJPI plug-in method applied to the Old faithful geyser dataset

not look the same using the two different software packages since the default settings for the histogram in the KDE procedure cannot be changed. Jones [10] declared that the SJPI plug-in method results in the best performer which is consistent, stable and can also be used as the default in software packages. The default bandwidth selection method of the KDE procedure in the SAS software program is the SJPI plug-in method.

5.2 Bivariate

The "ks" package in R will be used to generate different bivariate kernel estimates produced by using different bandwidth estimation methods and the multivariate Gaussian kernel function. The objective of this section is to compare different bandwidth estimation methods by fitting different kernel estimates firstly to simulated data and then to real-data examples. Simulated data from a bivariate normal distribution

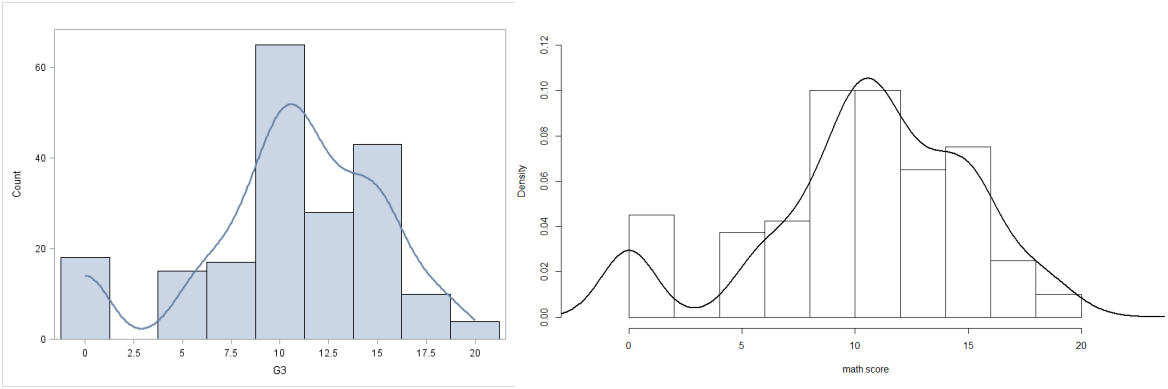


Figure 8: Kernel estimate using Silverman's rule of thumb applied to the Mathematic scores dataset

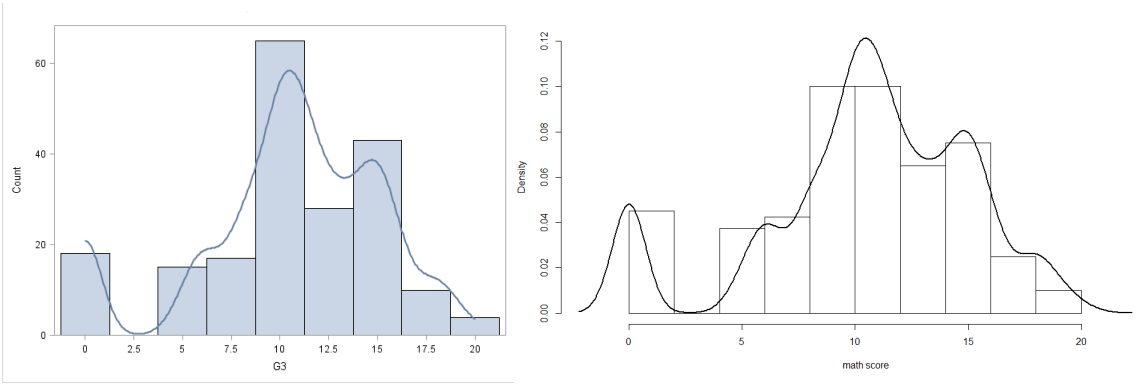


Figure 9: Kernel estimate using SJPI plug-in method applied to the Mathematic scores dataset

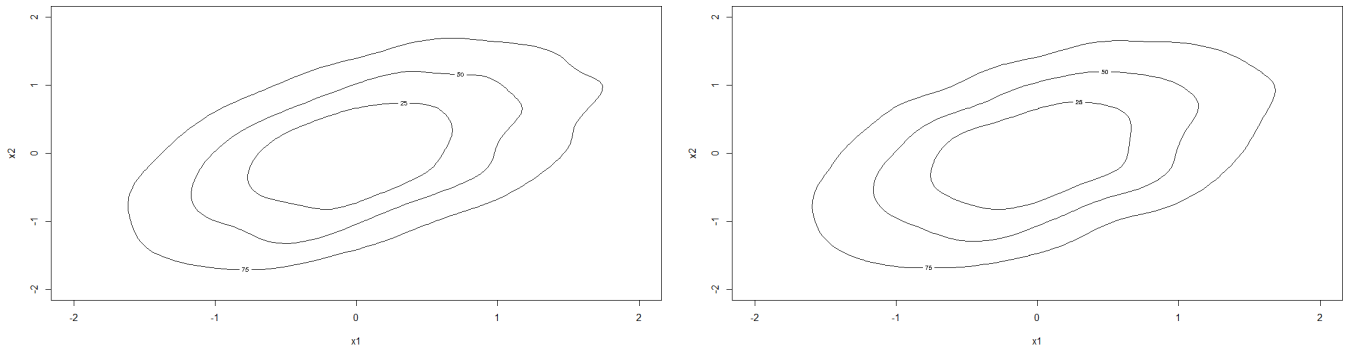


Figure 10: Kernel estimate using the LSCV bandwidth method for simulated bivariate normal data

with parameters

$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

was used with the following bandwidth methods LSCV, Plug-in, BCV1, BCV2 and Silverman's rule of thumb respectively to produce the different density estimates in Figure 10, 11, 12, 13 and 14. For the LSCV, plug-in, BCV1, and BCV2 bandwidth methods two different versions were used, a diagonal and unconstrained bandwidth matrix, to produce a kernel estimate. Those unconstrained bandwidth matrices which are not described above can be found in [5]. The first kernel estimate in each of the figures is produced by using the diagonal bandwidth matrix and the second one is produced by using the unconstrained bandwidth matrix. Looking at the graphs in Figure 10 and 11, only considering the class of parameterisation, it seems as if class 2 of parameterisation i.e. the diagonal bandwidth matrix led to a smoother density estimate for the LSCV and Plug-in method. Considering the class of parameterisation in Figure 12 and 13 results in no difference between the two kernel estimates produced for the BCV1 and BCV2 method. Actually, there is no difference between the four kernel estimates produced by the BCV1 method and the BCV2 method. Recall from above that the unconstrained bandwidth matrices can give a remarkable better performance for some types of density in this case the bivariate normal distribution is not one of the types of density. The kernel estimate produced by Silverman's rule of thumb looks very similar to the kernel estimates produced by the LSCV and plug-in unconstrained bandwidth matrices. By looking at the graphs a conclusion can be drawn that all the different bandwidth matrices considered here performed well when the underlying structure of the data has a bivariate normal distribution.

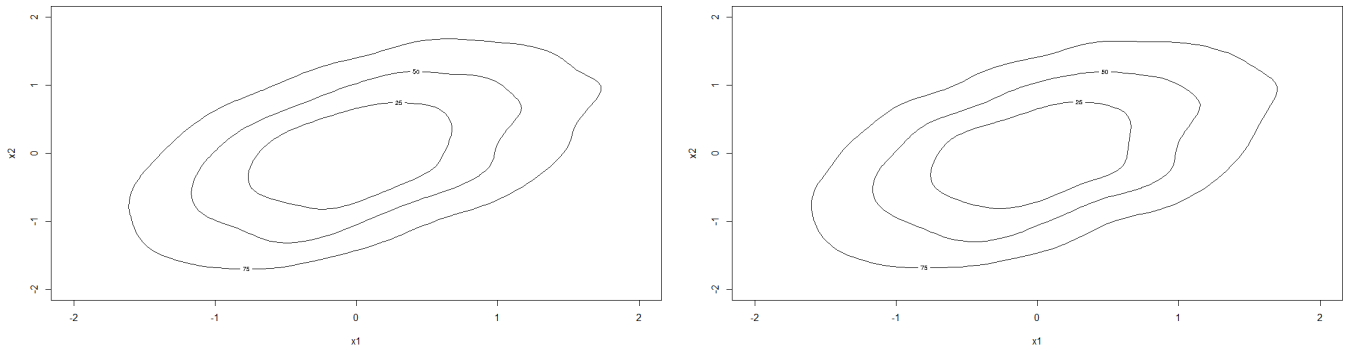


Figure 11: Kernel estimate using the plug-in method for simulated bivariate normal data

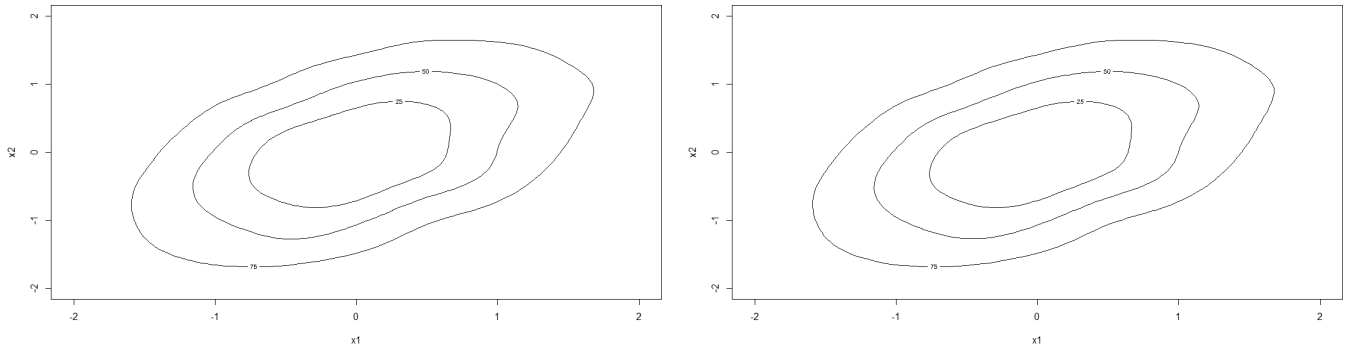


Figure 12: Kernel estimate using the BCV1 method for simulated bivariate normal data

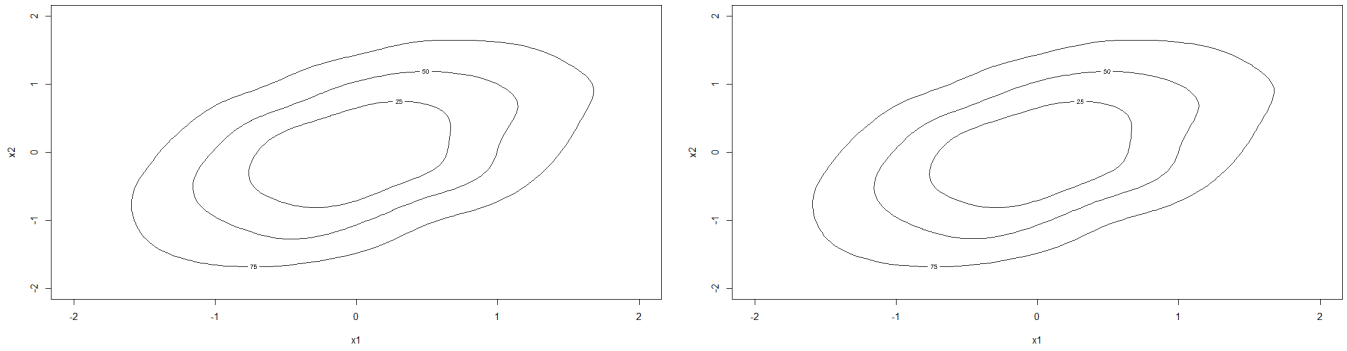


Figure 13: Kernel estimate using the BCV2 method for simulated bivariate normal data

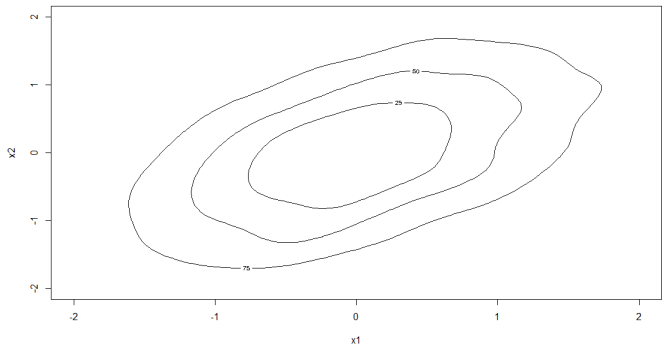


Figure 14: Kernel estimate using Silverman's multivariate rule of thumb for simulated bivariate normal data

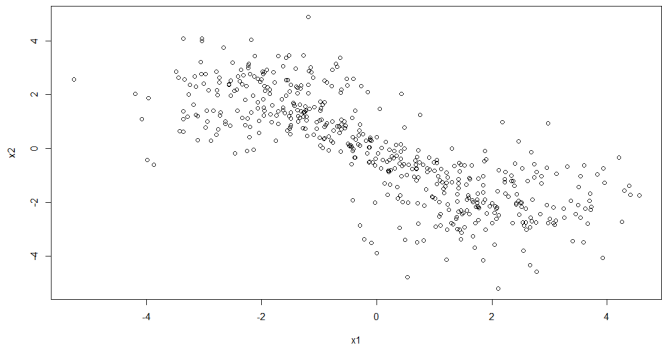


Figure 15: Scatterplot of a sample of 500 from the 'dumbbell' density

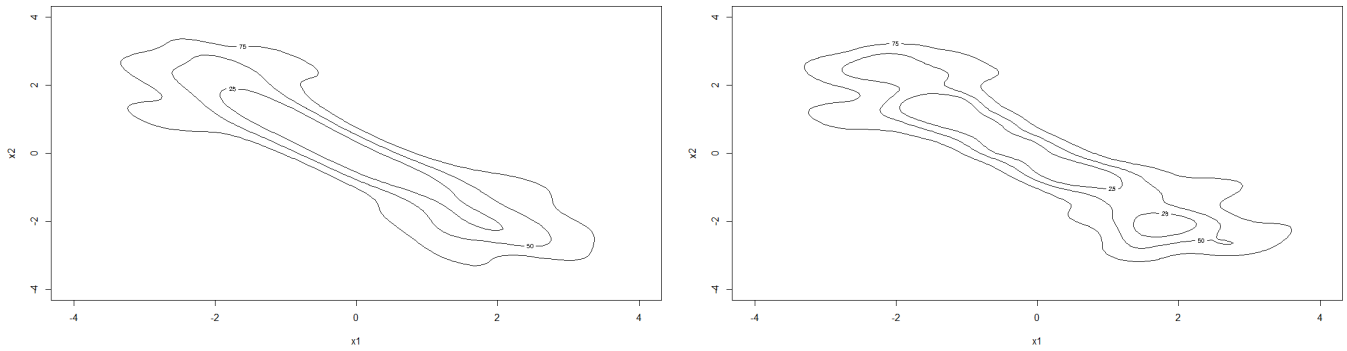


Figure 16: Kernel estimate using the LSCV bandwidth method for simulated bivariate 'dumbbell' data

Simulated data from the 'dumbbell' density proposed by Duong [5] was used with the following bandwidth methods LSCV, Plug-in, BCV1, BCV2, and Silverman's rule of thumb respectively to produce the different density estimates in Figure 16, 17, 18, 19 and 20. For the LSCV, Plug-in, BCV1, and BCV2 bandwidth methods two different versions were used, a diagonal and unconstrained bandwidth matrix, to produce a kernel estimate. By observing the graphs in Figure 16 and 17 it seems class 1 of parameterisation i.e. unconstrained bandwidth matrices led to a smoother density estimate. Considering the class of parameterisation in Figure 18 and 19 results in no difference between the two kernel estimates produced for the BCV1 and BCV2 method. These kernel estimates incorrectly produced a bimodal density estimate. The kernel estimates produced by using a diagonal bandwidth matrix version of the LSCV and Plug-in method also results in a bimodal density estimate. On the other hand, the unconstrained bandwidth matrix correctly produces a unimodal density estimate. Therefore this simulation study supports Duong's conclusion, excluding the BCV1 and BCV2 methods, that an unconstrained bandwidth matrix can give a remarkable better performance for densities which has large probability mass oriented away from the co-ordinate axes (oriented diagonally to the co-ordinate axes), such as the dumbbell density [5]. Judging the performance of these bandwidth matrices on the basis of observing the graphs it seems as if the unconstrained Plug-in bandwidth matrix has performed the best. Silverman's rule of thumb also performed very well but there may also be some oversmoothing present in the density estimate.

As mentioned it is important to see how the estimation technique performs in practice. The students' scores dataset will again be used for the real dataset examples including the Portuguese language scores for the bivariate case. Another dataset which will be used is the Default of Credit Card Clients Dataset [2]. The variables which will be considered are the age and the credit limit given in terms of New Taiwan dollar and a sample size of 500 will be used.

The kernel estimates of the scores sample in Figure 22, 24 and 25 are generated by the Plug-in,

[2]Datasets used can be found on Kaggle (https://www.kaggle.com/datasets).

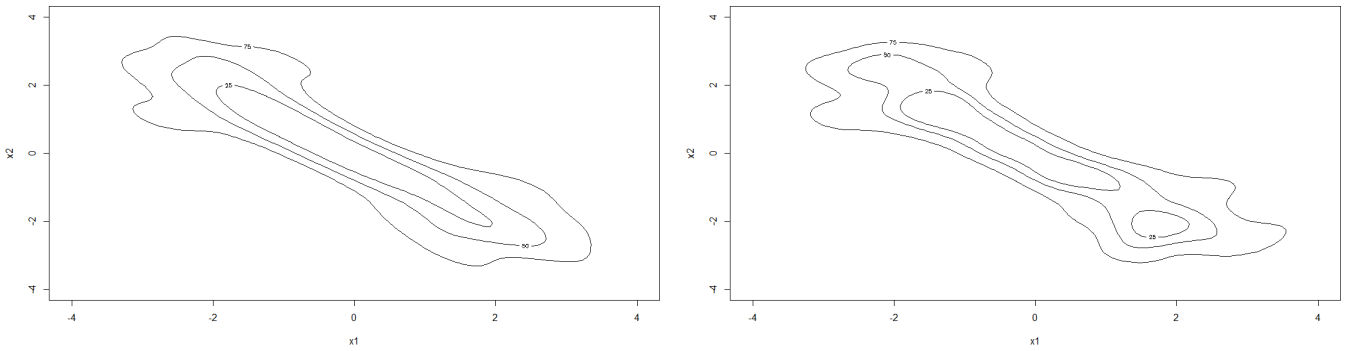


Figure 17: Kernel estimate using the Plug-in method for simulated bivariate 'dumbbell' data

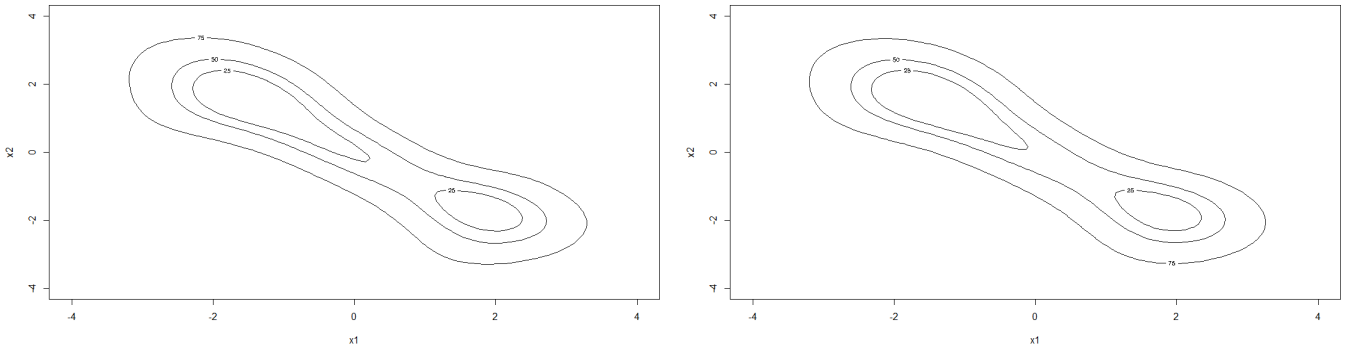


Figure 18: Kernel estimate using the BCV1 method for simulated bivariate 'dumbbell' data

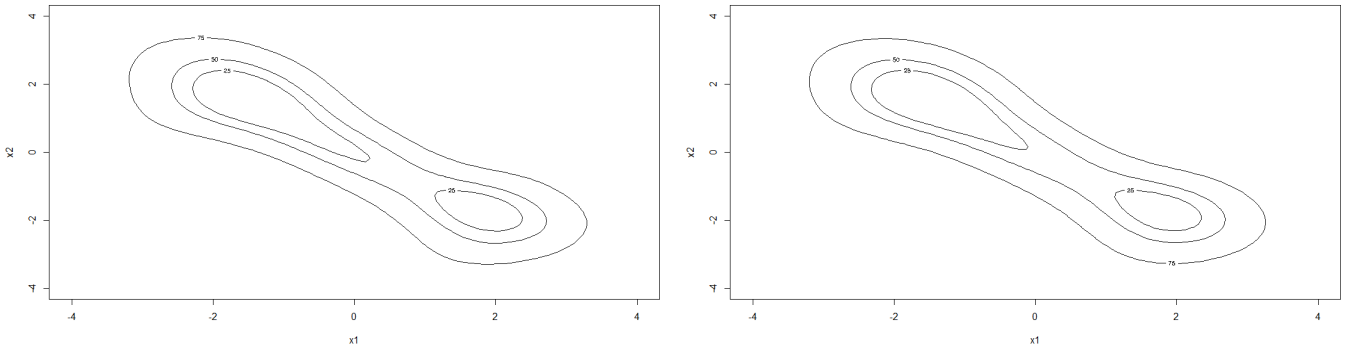


Figure 19: Kernel estimate using the BCV2 method for simulated bivariate 'dumbbell' data

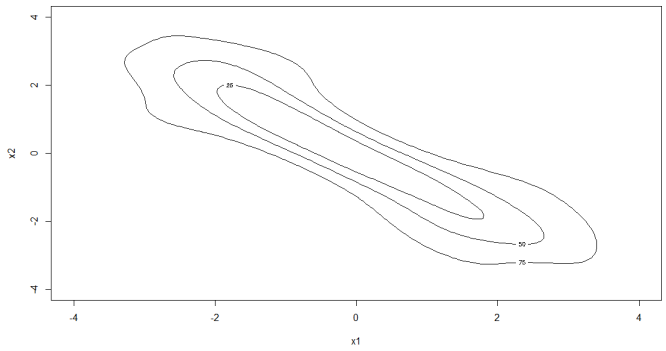


Figure 20: Kernel estimate using Silverman's multivariate rule of thumb for simulated bivariate 'dumbbell' data

BCV1 and BCV2 methods with unconstrained and diagonal bandwidth matrices respectively. In Figure 26 the kernel estimate for the scores sample is produced by using Silverman's rule of thumb. For the LSCV method, the duplicate values in the sample first have to be removed since the LSCV method is programmed in such a way that it can't function with duplicate values present in the sample. A sample of 93 students is left when the duplicate values of the bivariate sample are removed. In Figure 23 the kernel estimates of this new sample without the duplicate values are generated by the LSCV method with unconstrained and diagonal bandwidth matrices respectively. The contours of the density estimate produced by the LSCV method for the scores sample differs significantly from the others. For this dataset, the LSCV method may be incomparable to the other methods since the sample size was decreased dramatically to be able to apply the LSCV method. Comparing the other density estimates for the scores sample it seems the unconstrained and diagonal bandwidth matrices of the BCV1 and BCV2 methods has performed very similarly. Comparing these kernel estimates to Silverman's rule of thumb also results in very similar performance. The Plug-in method with the unconstrained bandwidth matrix in Figure 22 performed well with the scores dataset where the estimate is a bit less smooth than the one produced by Silverman's rule of thumb.

For the credit sample the estimates produced by the LSCV method may be comparable since the sample size was large and when the duplicates were removed it was still relatively large. The two kernel estimates in Figure 29 produced by the LSCV method (with unconstrained and diagonal bandwidth matrices) performed very similarly in the sense of producing a smooth density estimate but there may be oversmoothing present. Comparing the kernel estimates produced by the unconstrained and diagonal bandwidth matrices of the BCV1 and BCV2 methods results in very similar performance as in the case for the scores dataset. Comparing these kernel estimates to Silverman's rule of thumb also results in very similar performance (as with the scores dataset). The kernel estimates produced by the LSCV method may have some oversmoothing present, which will lead to missing important features. Comparing all

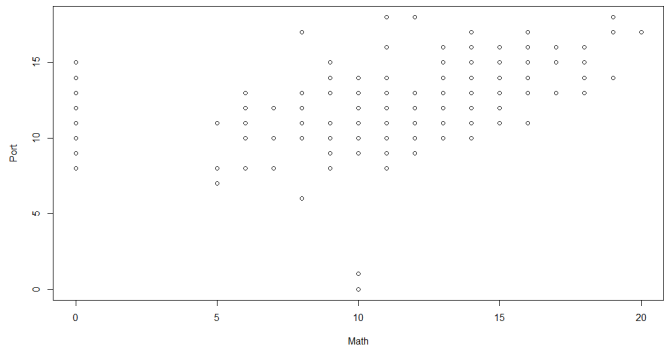


Figure 21: Scatter plot of a sample of the student scores dataset

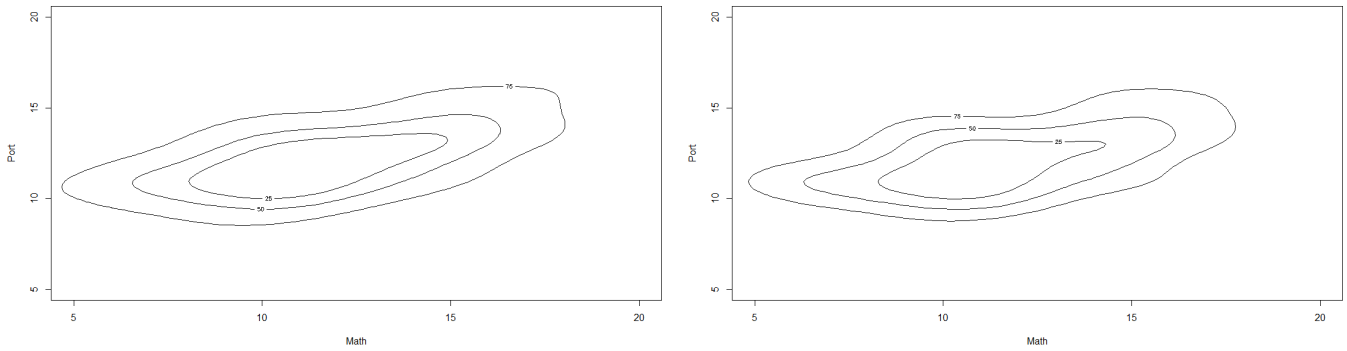


Figure 22: Kernel estimates using the Plug-in method for the student scores dataset

the kernel estimates it seems as if the diagonal bandwidth matrix of the plug-in method has performed the best although the unconstrained and diagonal matrix of the plug-in method have performed very similarly. For the credit dataset the values of the credit limit can only be positive but looking at the kernel estimates produced from this dataset led to kernel estimates including values less than 0. Bivariate kernel estimation is just a generalisation of univariate kernel estimation and the problem with bounded data continues when working with fixed bandwidth estimators. Recall from above kernel density estimates will often go beyond the bounds of the data and then the estimate is considerable bias at and near the bounds of the data [9]. Therefore all the kernel estimates produced for the credit dataset will be biased at and near 0 since they are all fixed bandwidth estimators.

In summary, the LSCV method is variable which is supported by Duong who also compared bandwidth methods in the bivariate case and concluded that the LSCV method is highly variable but is useful in some cases [5]. The 2-stage Plug-in method was one of the general recommended methods by Duong [5]. Silverman's rule of thumb will perform well across all dimensions was mentioned by Van der Walt and is supported by this study [22].

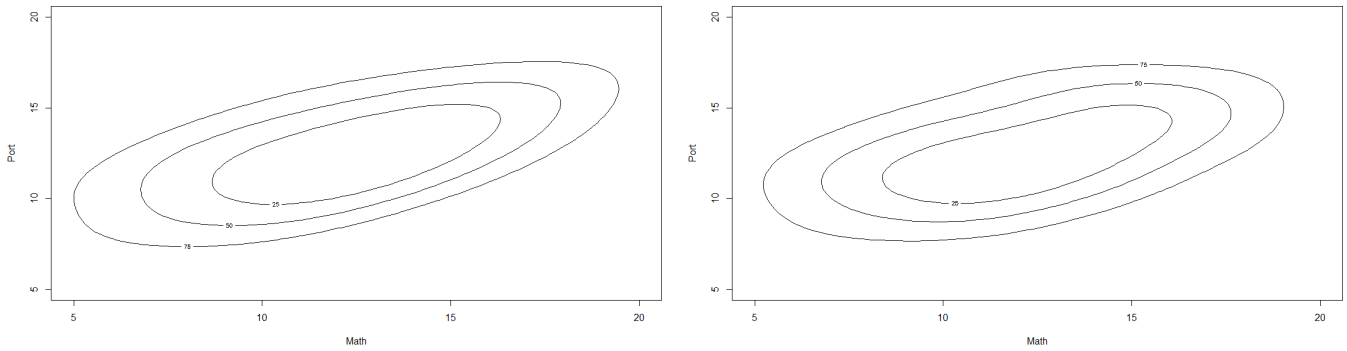


Figure 23: Kernel estimates using the LSCV method for the student scores dataset

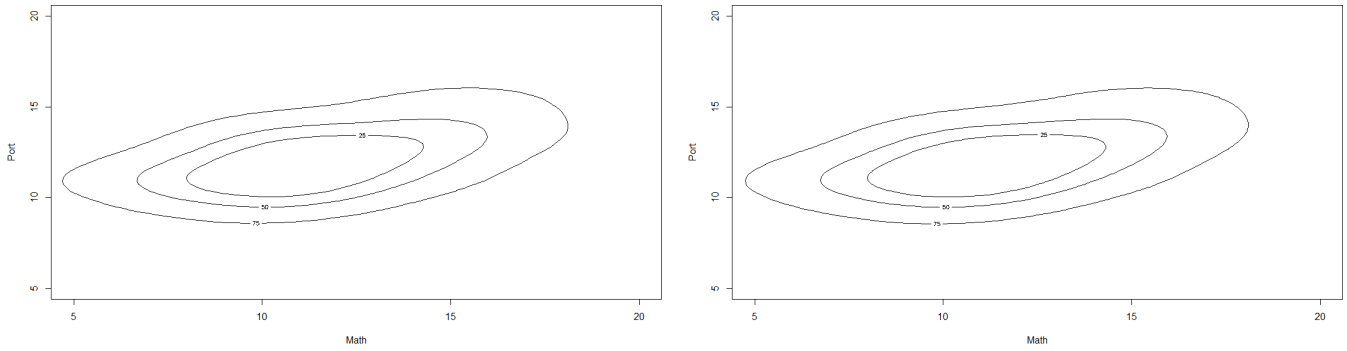


Figure 24: Kernel estimates using the BCV1 method for the student scores dataset

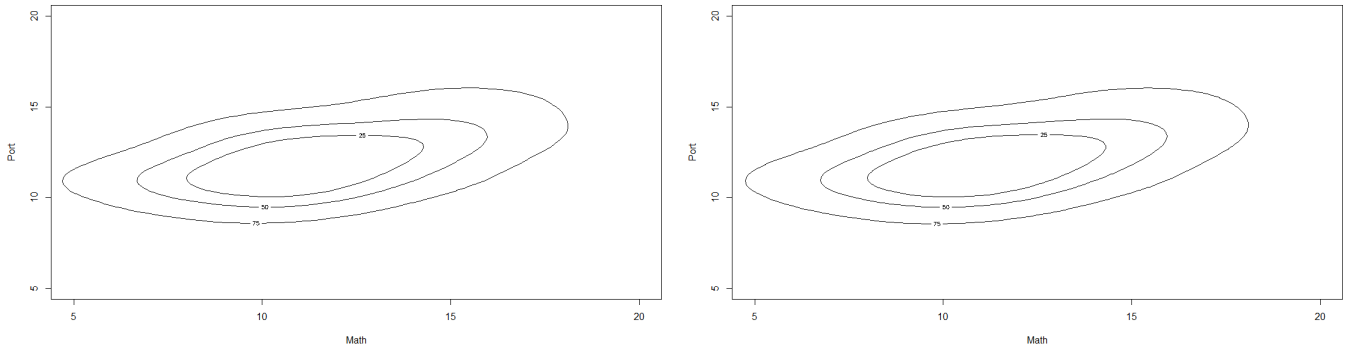


Figure 25: Kernel estimates using the BCV2 method for the student scores dataset

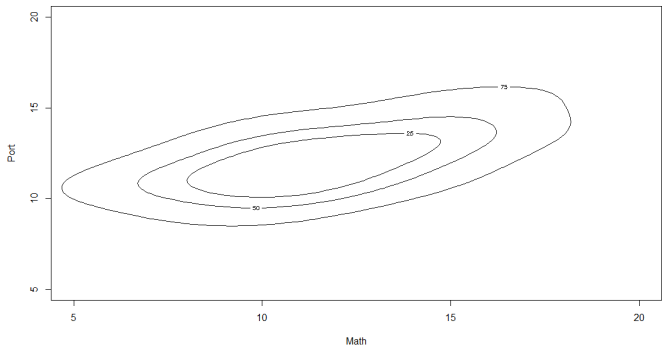


Figure 26: Kernel estimate using Silverman's rule of thumb for the student scores dataset

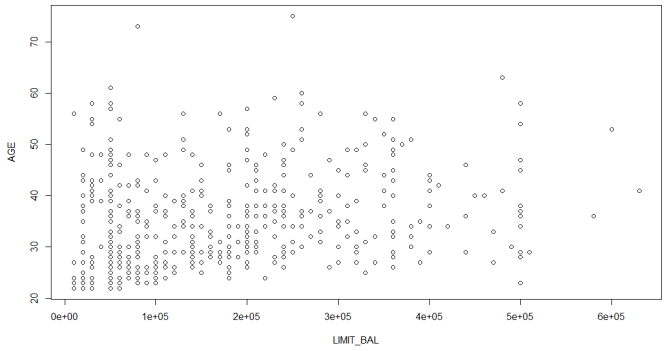


Figure 27: Scatter plot of a sample of the credit dataset

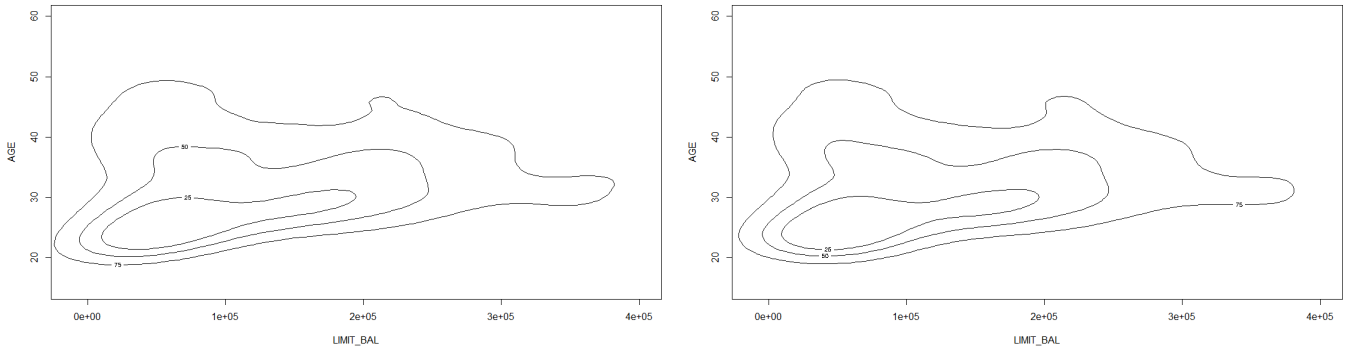


Figure 28: Kernel estimate using Plug-in method for the credit dataset

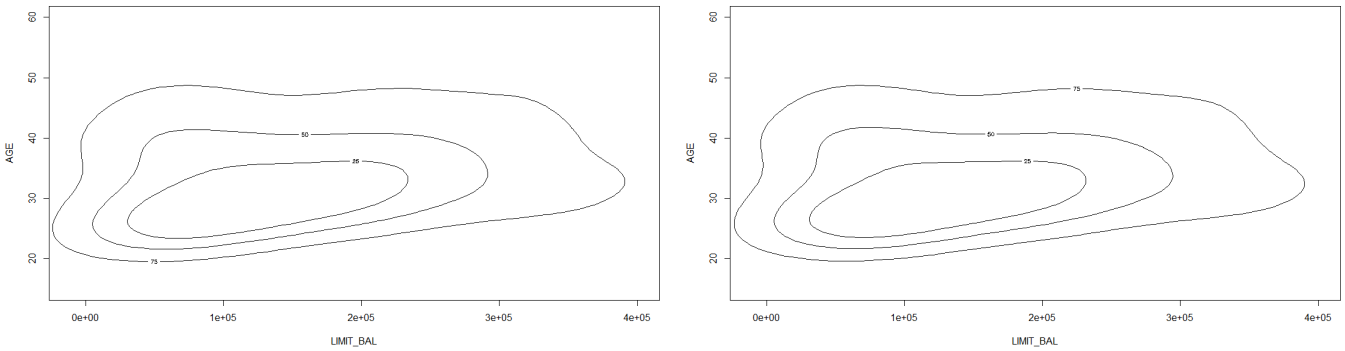


Figure 29: Kernel estimate using LSCV method for the credit dataset

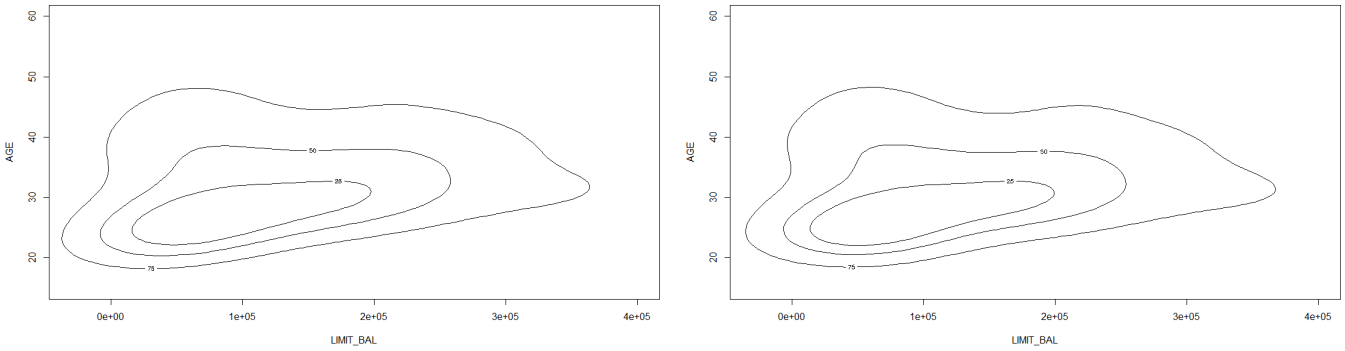


Figure 30: Kernel estimate using BCV1 method for the credit dataset

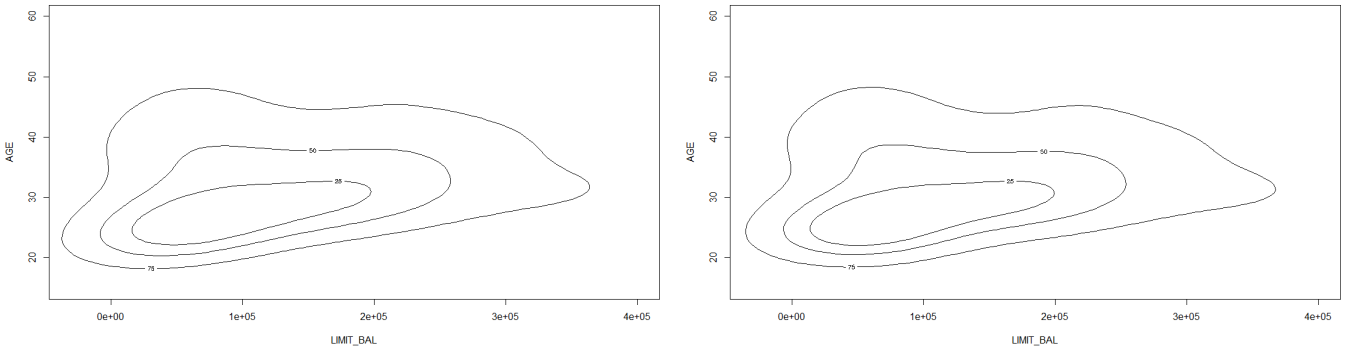


Figure 31: Kernel estimate using BCV2 method for the credit dataset

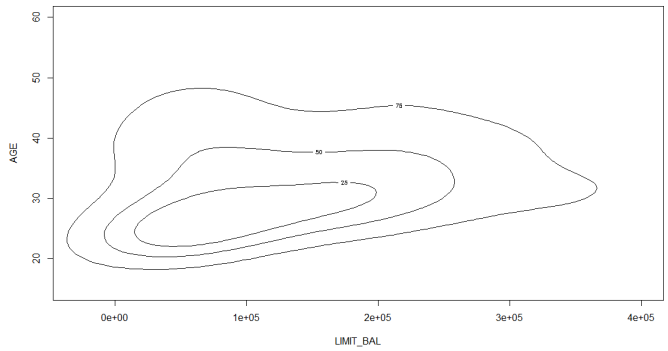


Figure 32: Kernel estimate using Silverman's rule of thumb for the credit dataset

6 Conclusion

For the univariate case, there has been good progress towards bandwidth estimation methods. The differences between the kernel estimates produced by Silverman's rule of thumb, least squares (unbiased) cross-validation, biased cross-validation and SJPI plug-in method of Sheater and Jones were investigated. The least squares (unbiased) cross-validation method performance is variable particularly in the direction of undersmoothing and therefore is unreliable. Also pointed out by [18] when the data has heavy tails, the least cross-validation method produces inconsistent fixed bandwidth kernel estimates. Biased cross-validation has a tendency to oversmooth the estimate. Silverman's rule of thumb is often extremely oversmoothed which leads to missing important features. The SJPI plug-in method results in the best performer which is consistent and stable.

More focus should be on performing diagnostics to detect lack of fit [12]. Loader used the Old Faithful geyser data, simulations based on a smoothed bootstrap approach, residual diagnostics and higher order fit to conclude that classical methods are correct in choosing small bandwidths, and the plug-in methods incorrectly oversmooth the estimate, with regard to the integrated square error loss function.

There will appear spurious (false) noise in the tails of the kernel estimate if the underlying distribution of the data has long tails because the bandwidth is fixed across the entire sample. Furtherer investigation into variable bandwidth methods is required.

For data from a multimodal distribution, it is difficult to find a single bandwidth which will adequately differentiate between distinct peaks and valleys between the peaks [16]. Choosing a bandwidth which is too large can erase modes which were significant and choosing a bandwidth which is too small may introduce spurious peaks by undersmoothing. More attention can go towards adaptive kernel density estimation introduced by Sain [16] which includes variable kernel estimators but also estimators which

attempts to identify and utilize local structure and other features in the underlying density through the sample data.

One-sided or two-sided bounded data can be problematic for kernel density estimation. Since the kernel function used in this method is not bounded it will result in treating this observed bounded data as if it is not bounded. Kernel density estimates will often go beyond the bounds of the data and then the estimate is considerable bias at and near the bounds of the data [11]. In [9] Jones provided a variety of boundary correction methods for kernel density estimation using a "generalised jackknifing" approach for many of the straight forward methods.

In the bivariate case only considering the parameterization of the bandwidth matrix the following conclusion was made. An unconstrained bandwidth matrix can give a remarkable better performance for densities which has large probability mass oriented away from the coordinate axes (oriented diagonally to the coordinate axes), such as the dumbbell density. The performance of the following bandwidth methods LSCV, Plug-in, BCV1, BCV2, and Silverman's rule of thumb were tested in the bivariate case. The LSCV method is highly variable but is useful in some cases, Silverman's rule of thumb performed well but the 2-stage Plug-in method resulted in the best performer. The results for the bivariate case are very similar to the univariate case. The problem with bounded data was also seen to be present in the bivariate case and a should be investigated further.

References

[1] Zdravko I Botev, Joseph F Grotowski, and Dirk P Kroese. Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.

[2] Daren BH Cline. Admissibile kernel estimators of a multivariate density. *The Annals of Statistics*, 16(4):1421–1427, 1988.

[3] R Dennis Cook and Sanford Weisberg. *An Introduction to Regression Graphics*, volume 405. John Wiley & Sons, 2009.

[4] Tarn Duong. *Bandwidth selectors for multivariate kernel density estimation*. PhD thesis, School of Mathematics and Statistics, University of Western Australia, 2004.

[5] Tarn Duong. ks: Kernel density estimation and kernel discriminant analysis for multivariate data in r. *Journal of Statistical Software*, 21(7):1–16, 2007.

[6] Tarn Duong and Martin Hazelton. Plug-in bandwidth matrices for bivariate kernel density estimation. *Journal of Nonparametric Statistics*, 15(1):17–30, 2003.

[7] Tarn Duong and Martin L Hazelton. Cross-validation bandwidth matrices for multivariate kernel density estimation. *Scandinavian Journal of Statistics*, 32(3):485–506, 2005.

[8] Peter Hall, JS Marron, and Byeong U Park. Smoothed cross-validation. *Probability Theory and Related Fields*, 92(1):1–20, 1992.

[9] M Chris Jones. Simple boundary correction for kernel density estimation. *Statistics and Computing*, 3(3):135–146, 1993.

[10] M Chris Jones, James S Marron, and Simon J Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91(433):401–407, 1996.

[11] MC Jones and PJ Foster. A simple nonnegative boundary correction method for kernel density estimation. *Statistica Sinica*, pages 1005–1013, 1996.

[12] Clive R Loader. Bandwidth selection: classical or plug-in? *Annals of Statistics*, 27(2):415–438, 1999.

[13] JS Marron and D Nolan. Canonical kernels for density estimation. *Statistics & Probability Letters*, 7(3):195–199, 1988.

[14] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[15] Vikas Chandrakant Raykar and Ramani Duraiswami. Fast optimal bandwidth selection for kernel density estimation. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 524–528. SIAM, 2006.

[16] Stephan R Sain. *Adaptive kernel density estimation*. PhD thesis, Rice University, 1994.

[17] Stephan R Sain, Keith A Baggerly, and David W Scott. Cross-validation of multivariate densities. *Journal of the American Statistical Association*, 89(427):807–817, 1994.

[18] Eugene F Schuster and Gavin G Gregory. On the nonconsistency of maximum likelihood nonparametric density estimators. In *Computer Science and Statistics: Proceedings of the 13th Symposium on the interface*, pages 295–298. Springer, 1981.

[19] David W Scott and George R Terrell. Biased and unbiased cross-validation in density estimation. *Journal of the American Statistical Association*, 82(400):1131–1146, 1987.

[20] Simon J Sheather and Michael C Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):683–690, 1991.

[21] Bernard W Silverman. *Density Estimation for Statistics and Data Analysis*, volume 26. CRC press, 1986.

[22] Christiaan Maarten Van der Walt. *Maximum–likelihood kernel density estimation in high–dimensional feature spaces*. PhD thesis, North-West University, 2014.

[23] Matt P Wand and M Chris Jones. Comparison of smoothing parameterizations in bivariate kernel density estimation. *Journal of the American Statistical Association*, 88(422):520–528, 1993.

[24] Matt P Wand and M Chris Jones. *Kernel Smoothing*. CRC Press, 1994.

[25] Matt P Wand and M Chris Jones. Multivariate plug-in bandwidth selection. *Computational Statistics*, 9(2):97–116, 1994.

Appendix

```
#Coding using the software program R:

#Univariate:

#Kernel estimates for simulated data from an exponential distribution with parameter 1

set.seed(10) x <- rexp(10000, rate = 1)
plot(density(x, bw = "nrd0"), bty = 'n', xlim = c(-2,10), ylim = c(0,1.1),
                          yaxt="n",ylab="", xlab="", main="Kernel density estimates",
                           col = "red")
lines(density(x, bw = "ucv"),bty = 'n', xlim = c(-2,10),
ylim = c(0,1.1), yaxt="n", ylab="", xlab="", col = "blue")
lines(density(x, bw = "bcv"),bty = 'n', xlim = c(-2,10),
                          ylim = c(0,1.1),yaxt="n", ylab="", xlab="", col = "green")
lines(density(x, bw = "SJ-ste"),bty = 'n', xlim = c(-2,10), ylim = c(0,1.1),
                            yaxt="n",yaxt="n",ylab="", xlab="", col = "purple")
label <- c("nrd0", "ucv", "bcv", "SJ-ste")
legend("topright", title="Bandwidth methods", label,lwd=2,
            col=c("red","blue","green","purple"))

#Kernel estimate for the geyser dataset (variable waiting)
#using silvermans rule of thumb with a histogram overlay

hist(geyser$waiting, freq =FALSE, main="" , xlim = c(35,105), ylim = c(0,0.06),
         xlab= "waiting")
lines(density(geyser$waiting, bw = "nrd0"), bty = 'n', xlim = c(35,105),
           ylim = c(0,0.06) ,yaxt="n",ylab="", xlab="", main="", lwd=2)

#Kernel estimate for the geyser dataset (variable waiting) using SJPI method
#with a histogram overlay

hist(geyser$waiting, freq =FALSE, main="" , xlim = c(35,105), ylim = c(0,0.06),
         xlab= "waiting")
```

```
lines(density(geyser$waiting, bw = "SJ-ste"), bty = 'n', xlim = c(35,105),
          ylim = c(0,0.06) ,yaxt="n",ylab="", xlab="", main="", lwd=2)

#Coding using the software program SAS:

PROC KDE data=geyser;
UNIVAR waiting /method=srot plots=all;
UNIVAR waiting /method=sjpi plots=all;
run;
PROC KDE data=Math;
UNIVAR G3 /method=srot plots=all;
UNIVAR G3 /method=sjpi plots=all;
run;

#Coding using the software program R:

#Bivariate:

#Kernel estimates for bivariate normal simulated data

library(MASS)
set.seed(25)
mu <- c(0,0)
Sigma <- matrix(c(1, .5, .5, 1), 2)
bivn <- mvrnorm(5000, mu = mu, Sigma = Sigma )
library(ks)
plugin_diag <- Hpi.diag(bivn)
plugin_full <- Hpi(bivn)
lscrossval_diag <- Hlscv.diag(bivn)
lscrossval_full <- Hlscv(bivn)
silver <- Hns(bivn, deriv.order=0)
fhat_plugin_diag <- kde(bivn,plugin_diag)
fhat_plugin_full <- kde(bivn,plugin_full)
fhat_lscv_diag <- kde(bivn,lscrossval_diag)
```

```
fhat_lscv_full <- kde(bivn,lscrossval_full)
fhat_silv <- kde(bivn, silver) plot(fhat_plugin_diag, xlim = c(-2,2),
                        ylim= c(-2,2), ylab="x2", xlab="x1", main="kernel
                        estimate using the Plug-in method with class 2
                        parameterisation bandwidth matrix" )
plot(fhat_plugin_full, xlim = c(-2,2), ylim= c(-2,2), ylab="x2",
        xlab="x1", main="kernel estimate using the Plug-in method with class 1
        parameterisation bandwidth matrix" )
plot(fhat_lscv_diag, xlim = c(-2,2), ylim= c(-2,2), ylab="x2", xlab="x1",
        main="kernel estimate using the Plug-in method with class 2
        parameterisation bandwidth matrix" )
plot(fhat_lscv_full, xlim = c(-2,2), ylim= c(-2,2), ylab="x2", xlab="x1",
        main="kernel estimate using the Plug-in method with class 1
        parameterisation bandwidth matrix" )
plot(fhat_silv, xlim = c(-2,2), ylim= c(-2,2), ylab="x2", xlab="x1",
        main="kernel estimate using Silverman's multivariate rule of thumb with
        class 3 parameterisation bandwidth matrix" )

#Kernel estimates for simulated data from the dumbbell density function:

library(ks)
samp <- 500
mus <- rbind(c(-2,2), c(0,0), c(2,-2))
Sigmas <- rbind(diag(2), matrix(c(0.8, -0.72, -0.72, 0.8), nrow = 2),
                    diag(2))
cwt <- 3/11
props <- c((1-cwt)/2, cwt, (1-cwt)/2)
x <- rmvnorm.mixt(n = samp, mu = mus, Sigma = Sigmas, props = props)
dens <- dmvnorm.mixt(x, mus, Sigmas, props)
plot(x, ylab="x2", xlab="x1", main="Scatter plot of a sample of 500 from the
        'dumbbell' density" )
library(ks)
plugin_diag <- Hpi.diag(x)
plugin_full <- Hpi(x)
```

```
lscrossval_diag <- Hlscv.diag(x)
lscrossval_full <- Hlscv(x)
silver <- Hns(x, deriv.order=0)
bcv1_diag <- Hbcv.diag(x, whichbcv = 1)
bcv1_full <- Hbcv(x, whichbcv = 1)
bcv2_diag <- Hbcv.diag(x, whichbcv = 2)
bcv2_full <- Hbcv(x, whichbcv = 2)
fhat_plugin_diag <- kde(x,plugin_diag)
fhat_plugin_full <- kde(x,plugin_full)
fhat_lscv_diag <- kde(x,lscrossval_diag)
fhat_lscv_full <- kde(x,lscrossval_full)
fhat_silv <- kde(x, silver)
fhat_bcv1_diag <- kde(x,bcv1_diag)
fhat_bcv1_full <- kde(x, bcv1_full)
fhat_bcv2_diag <- kde(x,bcv2_diag)
fhat_bcv2_full <- kde(x, bcv2_full)
plot(fhat_plugin_diag, xlim = c(-4,4), ylim= c(-4,4), ylab="x2", xlab="x1",
        main="kernel estimate using the Plug-in method with class 2
        parameterisation bandwidth matrix" )
plot(fhat_plugin_full, xlim = c(-4,4), ylim= c(-4,4), ylab="x2", xlab="x1",
        main="kernel estimate using the Plug-in method with class 1 parameterisation
        bandwidth matrix" )
plot(fhat_lscv_diag, xlim = c(-4,4), ylim= c(-4,4), ylab="x2", xlab="x1",
        main="kernel estimate using the Plug-in method with
        class 2 parameterisation bandwidth matrix" )
plot(fhat_lscv_full, xlim = c(-4,4), ylim= c(-4,4), ylab="x2", xlab="x1",
        main="kernel estimate using the Plug-in method with
        class 1 parameterisation bandwidth matrix" )
plot(fhat_silv, xlim = c(-4,4), ylim= c(-4,4), ylab="x2", xlab="x1",
        main="kernel estimate using Silverman's multivariate rule of thumb
        with class 3 parameterisation bandwidth matrix" )
plot(fhat_bcv1_diag, xlim = c(-4,4), ylim= c(-4,4), ylab="x2", xlab="x1",
        main="kernel estimate using the BCV1 method with
        class 2 parameterisation bandwidth matrix" )
```

```
plot(fhat_bcv1_full, xlim = c(-4,4), ylim= c(-4,4), ylab="x2", xlab="x1",
        main="kernel estimate using the BCV1 method with
        class 1 parameterisation bandwidth matrix" )
plot(fhat_bcv2_diag, xlim = c(-4,4), ylim= c(-4,4), ylab="x2", xlab="x1",
        main="kernel estimate using the BCV2 method with
        class 2 parameterisation bandwidth matrix" )
plot(fhat_bcv2_full, xlim = c(-4,4), ylim= c(-4,4), ylab="x2", xlab="x1",
        main="kernel estimate using the BCV2 method with
        class 1 parameterisation bandwidth matrix" )

#Kernel estimates for the student scores dataset

library(ks)
plot(scores, main="Scatter plot of a sample of the student scores dataset")
plugin_diag <- Hpi.diag(scores)
plugin_full <- Hpi(scores)
lscrossval_diag <- Hlscv.diag(scores2)
lscrossval_full <- Hlscv(scores2)
silver <- Hns(scores, deriv.order=0)
bcv1_diag <- Hbcv.diag(scores, whichbcv = 1)
bcv1_full <- Hbcv(scores, whichbcv = 1)
bcv2_diag <- Hbcv.diag(scores, whichbcv = 2)
bcv2_full <- Hbcv(scores, whichbcv = 2)
fhat_diag <- kde(scores,plugin_diag)
fhat_full <- kde(scores,plugin_full)
fhat_lscv_diag <- kde(scores2,lscrossval_diag)
fhat_lscv_full <- kde(scores2,lscrossval_full)
fhat_silv <- kde(scores, silver)
fhat_bcv1_diag <- kde(scores,bcv1_diag)
fhat_bcv1_full <- kde(scores, bcv1_full)
fhat_bcv2_diag <- kde(scores,bcv2_diag)
fhat_bcv2_full <- kde(scores, bcv2_full)
plot(fhat_diag, xlim= c(5,20), ylim = c(5,20), main="kernel estimate using
        the Plug-in method with class 2 parameterisation bandwidth matrix" )
```

```
plot(fhat_full, xlim= c(5,20), ylim = c(5,20), main="kernel estimate using
        the Plug-in method with class 1 parameterisation bandwidth matrix" )
plot(fhat_lscv_diag, xlim= c(5,20), ylim = c(5,20), main="kernel estimate
        using the LSCV method with class 2 parameterisation bandwidth matrix" )
plot(fhat_lscv_full, xlim= c(5,20), ylim = c(5,20), main="kernel estimate
        using the LSCV method with class 1 parameterisation bandwidth matrix" )
plot(fhat_silv, xlim= c(5,20), ylim = c(5,20), main="kernel estimate using
        Silverman's multivariate rule of thumb with
        class 3 parameterisation bandwidth matrix" )
plot(fhat_bcv1_diag, xlim= c(5,20), ylim = c(5,20), main="kernel estimate
using the BCV1 method with class 2 parameterisation bandwidth matrix" )
plot(fhat_bcv2_full, xlim= c(5,20), ylim = c(5,20), main="kernel estimate
using the BCV2 method with class 1 parameterisation bandwidth matrix" )
plot(fhat_bcv2_diag, xlim= c(5,20), ylim = c(5,20), main="kernel estimate
using the BCV2 method with class 2 parameterisation bandwidth matrix" )
plot(fhat_bcv1_full, xlim= c(5,20), ylim = c(5,20), main="kernel estimate
using the BCV1 method with class 1 parameterisation bandwidth matrix" )

#Kernel estimates for the credit card dataset

library(ks)
plot(credit_card, main="Scatter plot of a sample of the credit dataset")
plugin_diag <- Hpi.diag(credit_card)
plugin_full <- Hpi(credit_card)
silver <- Hns(credit_card, deriv.order=0)
bcv1_diag <- Hbcv.diag(credit_card, whichbcv = 1)
bcv1_full <- Hbcv(credit_card, whichbcv = 1)
bcv2_diag <- Hbcv.diag(credit_card, whichbcv = 2)
bcv2_full <- Hbcv(credit_card, whichbcv = 2)
fhat_diag <- kde(credit_card,plugin_diag)
fhat_full <- kde(credit_card,plugin_full)
fhat_silv <- kde(credit_card, silver)
fhat_bcv1_diag <- kde(credit_card,bcv1_diag)
fhat_bcv1_full <- kde(credit_card, bcv1_full)
```

```
fhat_bcv2_diag <- kde(credit_card,bcv2_diag)
fhat_bcv2_full <- kde(credit_card, bcv2_full)
plot(fhat_diag, xlim = c(-10000,4e+05), ylim=c(15,60), main="kernel estimate
     using the Plug-in method with class 2 parameterisation bandwidth matrix" )
plot(fhat_full,xlim = c(-10000,4e+05), ylim=c(15,60), main="kernel estimate
     using the Plug-in method with class 1 parameterisation bandwidth matrix" )
plot(fhat_silv, xlim = c(-30000,4e+05), ylim=c(15,60),main="kernel estimate
     using Silverman's multivariate rule of thumb with
    class 3 parameterisation bandwidth matrix" )
plot(fhat_bcv1_diag, xlim = c(-30000,4e+05), ylim=c(15,60),
     main="kernel estimate using the BCV1 method with class 2
     parameterisation bandwidth matrix" )
plot(fhat_bcv2_full, xlim = c(-30000,4e+05), ylim=c(15,60), main="kernel
     estimate using the BCV2 method with class 1 parameterisation bandwidth matrix"
plot(fhat_bcv2_diag, xlim = c(-30000,4e+05), ylim=c(15,60),
     main="kernel estimate using the BCV2 method with
    class 2 parameterisation bandwidth matrix" )
plot(fhat_bcv1_full, xlim = c(-30000,4e+05), ylim=c(15,60),
    main="kernel estimate using the BCV1 method with
     class 1 parameterisation bandwidth matrix" )

#Kernel estimates for the credit card dataset without duplicates
#using the LSCV method

lscrossval_diag <- Hlscv.diag(credit_88)
lscrossval_full <- Hlscv(credit_88)
fhat_lscv_diag <- kde(credit_88,lscrossval_diag)
fhat_lscv_full <- kde(credit_88,lscrossval_full)
plot(fhat_lscv_diag, xlim = c(-10000,4e+05), ylim=c(15,60),
    main="kernel estimate using the LSCV method with
     class 2 parameterisation bandwidth matrix" )
plot(fhat_lscv_full, xlim = c(-10000,4e+05), ylim=c(15,60),
     main="kernel estimate using the LSCV method with
     class 1 parameterisation bandwidth matrix" )
```

Maximum likelihood estimation for incomplete tables

Nozibusiso Tembe 14316073

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr N Strydom

Department of Statistics, University of Pretoria

30 October 2017 (final)

Abstract

Data in different fields of study, such as medicine and biostatistics, can be analysed using multiway contingency tables. Loglinear models are useful for this analysis. Maximum likelihood estimation is used to obtain the parameter estimates. The purpose of this study is to look at aspects of maximum likelihood estimation in loglinear modeling. The problems caused by having zero cells in the contingency table are considered specifically.

Declaration

I, *Nozibusiso Tenele Tembe,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____

Nozibusiso Tenele Tembe

_____

Dr N Strydom

_____

Date

Acknowledgements

I acknowledge the financial support from the Center for AI Research, Meraka Institute, CSIR. I would also like to convery my gratitude to my supervisor, Dr N Strydom. It is wholeheartedly expressed that your advice for my research proved to be a landmark effort towards the success of my project.

Contents

List of Figures

List of Tables

1 Introduction

"All models are wrong, but some are useful." said Box [5]. Loglinear models are no exception to the rule. A loglinear model is a mathematical model used to analyse categorical data whose logarithm is a linear combination of parameters. We will estimate these parameters using the maximum likelihood estimation method. Categorical data can be represented in the form of a contingency or a multi-way table. Contingency tables can either be complete or incomplete. The former refers to a case where the table has values for all the cells available and the latter is a case where the table is partially filled; meaning at least one of the cells have a zero value. These zeros are one of two types:

a) Structural (fixed) zeros i.e. zero values caused by having no observations in that particular cell.

b) Sampling (random) zeros i.e. zero value caused by variation in sample or a sample size that is too small, rectified by increasing the sample size since there exists a probability of having a non-zero value in the cell.

Under statistical independence, the formula used to calculate the expected frequencies for a particular two-way contingency table is given by the following formula:

$$\mathrm{E_{ij}} = \frac{(\text{Marginal total of row i}) \times (\text{Marginal total of column j})}{(\text{grand total})}$$

Having sampling zeros reduces the expected frequency calculated by the formula above and may potentially cause the maximum likelihood estimate (MLE) to be non-existent. The following example illustrates the impact of having zero valued cells when the calculation of expected frequencies ($\mathrm{E_{ij}}$) is concerned. Table 1 summarises the observed frequencies:

	1	2	
1	8	3	11
2	5	9	14
	13	12	25

Table 1: 2×2 table of observed values

The expected values are calculated in Table 2:

	1	2	
1	5.72	5.28	11
2	7.28	6.72	14
	13	12	25

Table 2: Expected frequencies of each cell obtained.

If the value 3 in row 1, column 2 of Table 1 is replaced by 0 then the new expected frequencies of row 1 are given in Table 3:

	1	2	
1	4.72	4.36	8

Table 3: Adjusted expected frequencies incorporating the zero cell.

The calculation problem caused by having zero valued cells will be referred to as the problem of zeros [7]. The existence of the MLE contributes vastly towards model selection. Even in cases where the distribution of a statistic measuring the goodness of fit is taken from a known distribution, the MLE is needed to identify the discrepancy of the observed data from the fitted data. In this paper we will address the following questions:

1. What are the conditions for the existence of the MLE?
2. If some cells in our contingency table contain structural zeros, how is this handled in the calculation of the MLE?
3. How accurate are the values produced after a method that deals with zero cells has been applied?

1.1 Literature Review

Categorical data analysis dates back to the 19th century with Statistician Karl Pearson [1], but is first described at the start of the 20th century [7]. From the 20th century onwards the largest strides in categorical data analysis were taken, where model development is concerned. A contingency table can be used to represent categorical data and if information in one or more categorical variables is missing it is called an incomplete contingency table. The term "contingency" originated with Pearson who used it to refer to a measure of total deviation from "independent probabilities" in an $i \times j$ table. Years later the word is used to refer to the actual table itself. In 1940 Pearson developed the tetrachoric correlation coefficient for 2×2 tables with the assumption of bi-variate normality but there were problems countered in the development [11]. The chosen approach at the time was the cross-classification using discrete fixed variables, finding a structural relationship between them.

Loglinear models for multiway tables can be separated into dependence and independence models as well as, saturated and unsaturated models. The saturated models are more complex involving association terms that incorporate deviations from independence, unsaturated models are similar but do not include association terms. In practice, unsaturated models are preferred because they are easier to interpret and fit smoothly on sample data [1].

For a two-way $I \times J$ table , the general loglinear model of independence is

$$\log m_{ij} = u + u_{1(i)} + u_{2(j)}$$

where m_{ij} denotes the expected value of cell (i, j), $u_{1(i)}$ is the row effect and $u_{2(j)}$ is the column effect. The saturated model includes the association term $u_{12(ij)}$ and is given by:

$$\log m_{ij} = u + u_{1(i)} + u_{2(j)} + u_{12(ij)} \tag{1}$$

The total number of non-redundant parameters is $1 + (I - 1) + (J - 1) + (I - 1)(J - 1) = IJ$, which is the same as the total number of cells in the table.

2 Background Theory

2.1 Sampling distributions

Sampling distributions assist with obtaining maximum likelihood estimates. There are various methods of sampling that may be used and the three most commonly encountered distributions are elaborated on below.

2.1.1 Independent Poisson sampling

Under this type of sampling, each cell has an independent Poisson distribution. For a two dimensional array, the probability density function (PDF) is:

$$f(\{x_{ij}\}) = \prod_{i,j} \frac{\exp(-m_{ij}) m_{ij}^{x_{ij}}}{x_{ij}!}$$

The Poisson distribution occurs when sampling is done over a fixed period of time with no prior knowledge about the number of observations during that period.

2.1.2 Simple multinomial sampling

Under this type of sampling, the total sample size is a fixed N. Given a series of independent Poisson distributions, if this restriction is imposed it yields a multinomial distribution. The PDF of a two dimensional array is:

$$f(\{x_{ij}\}) = \frac{N!}{\prod_{i,j} x_{ij}!} \prod_{i,j} (\frac{m_{ij}}{N})^{x_{+j}}$$

2.1.3 Binomial sampling (special case of simple multinomial sampling)

Under this type of sampling, the number of units/sample size N is predetermined and classified according to two levels of a categorical variables. The PDF of a two dimensional array is:

$$p(x) = \frac{x_{ij}!}{m_{ij}!(x_{ij} - m_{ij})!} (\frac{x_{ij}}{N})^{m_{ij}} (1 - \frac{x_{ij}}{N})^{x_{ij} - m_{ij}}$$

where $\frac{x_{ij}}{N}$ is the probability of a count falling in cell (i, j).

2.2 Sufficient statistics

To fit the loglinear models, first sufficient statistics must be derived. Sufficient statistics can be derived for the different distributions. Below, is the derivation for the Poisson distribution and for the other distributions it follows on similarly.

Consider the following joint Poisson probability:

$$\prod_i \prod_j \frac{\exp(-m_{ij}) m_{ij}^{n_{ij}}}{n_{ij}!}$$

The likelihood function is given by:

$$\mathrm{L}(\mu) = \sum_i \sum_j n_{ij} \log m_{ij} - \sum_i \sum_j m_{ij} \tag{2}$$

Substituting the general loglinear model (1) ,described above, into (2) results in the likelihood function being written as follows:

$$\begin{aligned} \mathrm{L}(\mu) &= nu + \sum_i n_{i+} u_{1(i)} + \sum_j n_{+j} u_{2(j)} \\ &\quad + \sum_i \sum_j n_{ij} u_{12(ij)} - \sum_i \sum_j \exp(u + \ldots + u_{12(ij)}) \end{aligned} \tag{3}$$

where the $\{n_{ij}\}$ are the coefficients of $\{u_{12(ij)}\}$. The coefficients are called the minimal sufficient statistics. To find the estimate for the minimal sufficient statistic, differentiate equation (3) with respect to the relevant $u-$ term and equate the results of differentiation to zero.

2.3 Existence of maximum likelihood estimates

The first method of computing MLEs calculates the MLEs for a model with no second order association for a 2^3 table [2]. Two PhD students from University of North Carolina, later added the concept of explanatory variables and response variables, having explanatory variables being fixed factors [9, 10]. Collectively, they formulated the theory of two-way and three-way contingency tables and how they are integrated through conditioning of different factors. Later, the method of testing for any interaction between variables in a three-way table was derived but the existence of the MLE was not considered in the research [9, 10]. The use of logarithmic expansion of cell mean vector was introduced thereafter [3]. Then, assuming that there are no sampling zeros, it was also showed that there is a unique maximum for the log likelihood function. He also showed that the marginal sufficient statistics as well as the minimal

sufficient statistics are equal to MLEs of expectations. The conditions sufficient for the existence of unique non-zero MLEs were later given [6]. They were then updated to give necessary and sufficient conditions under the Poisson and product-multinomial sampling schemes [8] .

The pressing matter is the problem caused by having cells containing structural in the contingency table, affecting the accuracy of the model [1]. Throughout the centuries methods of determining necessary and sufficient conditions for a MLE to exist have been derived and updated continuiously. These necessary and sufficient conditions deal with the base level of what will be referred to from here onwards as the "zero values matter". The rest of the problem encompasses the tedious task of finding a procedure to deal with the zero values in an attempt to compute accurate, reliable MLEs and hence an accurate reliable model.

2.4 Maximum likelihood estimation

Maximum likelihood estimation aids in the obtaining of estimates for the parameters of the loglinear model by maximizing the likelihood function. There are some advantages to using maximum likelihood estimation for the loglinear model, namely:

1. The MLEs for loglinear models are easier to compute.
2. The MLEs follow certain intuitive marginal constraints which are not so intuitive for other estimation methods.
3. The maximum likelihood estimation method can be applied directly to multinomial data consisting of zero values without producing zero valued estimates.

2.4.1 Completeness

In as much as an incomplete table may have zero valued cells, a complete table may also have zero valued cells. The difference between the zero valued cells in a complete table, however, is that those zeros are sampling zeros. For a particular two-way contingency table, to find the expected cell estimates $\{\hat{m}_{ij}\}$ under the model of independence, the formula described earlier for E_{ij} is applied as follows:

$$\hat{m}_{ij} = \frac{(m_{i+})(m_{+j})}{m_{++}}$$

where the number of individual observation from a sample size of N in cell $(i., j)$ of the $I \times J$ contingency table is given by m_{ij}. If we are summing a subscripted variable over a subscript, a " + " is used to indicate the subscript we are summing. For example,

$$m_{i+} = \sum_{j=1}^{J} m_{ij}$$

for $i = 1, \ldots, I$, is the expected number of individual observations summing $j = 1, \ldots, J$ over each subscript $i = 1, \ldots, I$. Similarly we have

$$m_{+j} = \sum_{i=1}^{I} m_{ij}$$

for $j = 1, \ldots, J$ and

$$m_{++} = \sum_{i=1}^{I} \sum_{j=1}^{J} m_{ij}.$$

These are the maximum likelihood estimates under the two-way independence model. In the event that the complete contingency table has cells with no observed counts (sampling zeros) the pattern formed by the zero values in the table determines which loglinear model must be fitted.

To illustrate a situation where a complete contingency table has zero valued cells but they are positioned in such a manner that they do not affect how estimates for the model are calculated, consider the Table 4:

a	0
0	d

Table 4: A complete table where a and d denote nonzero positive values

Under the general two-way independence model described earlier, the estimates for the model are given in Table 5:

$\frac{a^2}{a+d}$	$\frac{ad}{a+d}$
$\frac{ad}{a+d}$	$\frac{d^2}{a+d}$

Table 5: Expected value estimates for the two-way independence model.

The cell estimates obtained in Table 5 are then used to calculate the $u-$ term estimates under the two-way table's unsaturated loglinear independence model.

$$\hat{u} = \log \frac{ad}{a+d}$$

$$\hat{u}_{1(1)} = -\hat{u}_{1(2)} = \frac{1}{2} log(\frac{a}{b})$$

$$\hat{u}_{2(1)} = -\hat{u}_{2(2)} = \frac{1}{2} log(\frac{a}{b})$$

To illustrate a situation where a complete table has zero valued cells but they are positioned in such a manner that it is impossible to derive cell estimates since the top row adds up to 0, consider Table 6:

0	0
c	d

Table 6: A complete table where c and d denote nonzero positive values

In this case, an alternative method is to fit a different model. A general two-way independence, model as described earlier, may be used where $u_{1(i)} = 0$, which means that the first variable has no effect on the model. The estimates for the model are given in Table 7:

$\frac{c}{2}$	$\frac{d}{2}$
$\frac{c}{2}$	$\frac{d}{2}$

Table 7: Expected value estimates for the two-way independence model.

2.5 Maximum likelihood estimation for complete tables

Consider the likelihood function from equation (3) above which now becomes:

$$\begin{aligned} \mathrm{L}(\mu) &= nu + [\sum_i \sum_j n_{ij} U_{12}] \\ &\quad -[\sum_i n_{i+} u_1 + \sum_j n_{+j} u_2] - \sum_i \sum_j \exp(u + \ldots + u_{12(ij)}) \end{aligned}$$

The term isolated within first set of square brackets above yields the minimal sufficient statistic, the terms in the second set of square brackets can be ignored. The minimal sufficient statistic corresponding to U_{12} is n_{ij}. The unique set of maximum likelihood estimates for every cell may be derived from sufficient statistics. Under some models the estimates can be written directly as a function of sufficient statistics.

For the two-way table model given in equation (1), if we put $u_{12} = 0$ the minimal sufficient statistic are n_{i+} and n_{+j}.

Note that $n_{i+} = \hat{m}_{i+}$ and $n_{+j} = \hat{m}_{+j}$. Isolating m_{ij} in the model it follows that:

$$m_{i+} = \exp(u + u_{1(i)})(\sum_j \exp(u_{2(j)})) \tag{4}$$

$$m_{+j} = \exp(u + u_{2(j)})(\sum_i \exp(u_{1(i)})) \tag{5}$$

$$N = \exp(u)(\sum_i \exp(u_{1(i)}))(\sum_j \exp(u_{2(j)}))$$

therefore

$$\frac{(m_{i+})(m_{+j})}{N} = \exp(u + u_{1(i)} + u_{2(j)}) = m_{ij}$$

In this case cell estimates are given by n_{i+} and n_{+j} as:

$$\hat{m}_{ij} = \frac{n_{i+}n_{+j}}{N}$$

For the purpose of illustrating direct estimation, consider the following model for a three-way table $\log m_{ijk} = u + u_{1(i)} + u_{2(j)} + u_{3(k)} + u_{12(ij)} + u_{23(jk)} + u_{13(ik)} + u_{123(ijk)}$. If $u_{12} = u_{123} = 0$, m_{ijk} is isolated as follows:

$$m_{ijk} = \exp(u + u_{1(i)} + u_{2(j)} + u_{3(k)} + u_{13(ik)} + u_{23(jk)})$$

the elements of n_{i+k} and n_{+jk}, are estimates of :

$$m_{i+k} = \exp(u + u_{1(i)} + u_{3(k)} + u_{13(ik)})(\sum_j \exp(u_{2(j)} + u_{23(jk)}))$$

$$m_{+jk} = \exp(u + u_{2(j)} + u_{3(k)} + u_{23(jk)})(\sum_i \exp(u_{1(i)} + u_{13(ik)}))$$

The minimal sufficient statistic that they have in common is n_{++k} which is an estimate of :

$$m_{++k} = \exp(u + u_{3(k)}) \sum_{i,j} \exp(u_{1(i)} + u_{2(j)} + u_{13(ik)} + u_{23(jk)})$$

Rearranging, the above equation may be written as:

$$m_{++k} = \exp(u + u_{3(k)}) \sum_i \exp(u_{1(i)} + u_{13(ik)}) \sum_j \exp(u_{2(j)} + u_{23(jk)})$$

Dividing $(m_{i+k})(m_{+jk})$ by m_{++k} the following is obtained:

$$m_{ijk} = \frac{(m_{i+k})(m_{+jk})}{m_{++k}}$$

where the estimates can be directly obtained as:

$$\hat{m}_{ijk} = \frac{(n_{i+k})(n_{+jk})}{n_{++k}}$$

3 Introduction to maximum likelihood estimation for incomplete tables

The problem that is investigated in this paper revolves around having zero valued cells in incomplete contingency tables. A researcher dealing with an incomplete contingency tables may fail to identify that their contingency table is incomplete. The repercussions of this are that the researcher might choose to:

1. Fill in the zero valued cells with values that are deemed "appropriate".
2. Collapse the table until there are no longer any structural zeros visible.
3. Discard the whole investigation.

All these cases result in either inaccurate or non-existent conclusions for the particular investigation. In the event that the researcher does identify that their contingency table as incomplete, a loglinear model suitable for incomplete data may be applied. This is called the quasi-loglinear model. In the upcoming sections quasi independence is introduced and explained further.

3.1 Quasi independence for two-way tables

Quasi independence states that a particular subset of cells (all cells not falling on the main diagonal) satisfies the independence structure whereby the cell expected count is a product of a row effect and a column effect. Let S be a set of cells in an incomplete two-way array made up of the cells not containing structural zeros and, under a common independence model of variables, it is assumed that:

$$m_{ij} = a_i b_j \tag{6}$$

for all $m_{ij} \in S$, where a_i and $b_j$are positive constants for $i = 1, \ldots, I$ and $j = 1, \ldots, J$. Other notation that is used to describe this definition of independence is $\Pi_{i,j}$, where $\Pi_{i,j}$ denotes the probability that a value in the $I \times J$ population table falls in the (i, j) cell. The rows and columns are considered independent if the cell probabilities can be written as $\Pi_{ij} = \Pi_{i+}\Pi_{+j}$ for $i = 1, \ldots, I$ and $j = 1, \ldots, J$. This allows for the use of the m_{ij} formulas discussed initially to find the marginal totals of S, since the cells in S contain no structural zeros.

Using the logarithmic model approach on the incomplete two-way tables we have:

$$\log m_{ij} = u + u_{1(i)} + u_{2(j)} + u_{12(ij)} \text{ for } (i, j) \in S \tag{7}$$

$$\sum_{i=1}^{I} \delta_i^{(2)} u_{1(i)} = \sum_{j=1}^{J} \delta_j^{(1)} u_{2(j)} = 0 \tag{8}$$

$$\sum_{i=1}^{I} \delta_{ij} u_{12(ij)} = \sum_{j=1}^{J} \delta_{ij} u_{12(ij)} = 0$$

where

$$\delta_{ij} \quad = \quad \begin{cases} 1 & \text{for } (i,j) \in S \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

$$\delta_i^{(2)} \quad = \quad \begin{cases} 1 & \delta_{ij} = 1 \text{ for some j} \\ 0 & \text{otherwise,} \end{cases}$$

$$\delta_j^{(1)} \quad = \quad \begin{cases} 1 & \delta_{ij} = 1 \text{ for some i} \\ 0 & \text{otherwise,} \end{cases}$$

setting $u_{12(ij)} = 0$ the quasi independence model is defined as:

$$\log m_{ij} = u + u_{1(i)} + u_{2(j)} \text{ for } (i,j) \in S \tag{10}$$

Isolating m_{ij} result in the following equation:

$$m_{ij} = \exp(u_{1(i)}) \exp(u_{2(j)}) \exp(\text{u}) \text{ for } (i,j) \in S \tag{11}$$

The equation (11) is equivalent to (6) since the m_{ij} is given as the product of a parameter depending on i, one depending on j. The last parameter depends on neither i or j.

3.2 Interpretation of the quasi independence model

3.2.1 Method 1

Quasi independence means that the relative proportions of values in the corresponding cells of two rows/columns are the same, given that there are no structural zeros in one of the cells in these rows/columns. So quasi independence is dependent on the fact that attention is diverted from the structural zero entries to the non-zero portion of the table. To illustrate quasi independence consider Table 8 below:

0	a	b
c	d	e
f	g	h

Table 8: Table illustrating the position of structural zero valued cells

Applying quasi independence, the first row can be deleted or alternatively the first column can be deleted. For each case the remaining form is shown below in Table 9 and Table 10, respectively.

c	d	e
f	g	h

Table 9: 2×3 table remaining after the first row is deleted.

a	b
d	e
g	h

Table 10: 3×2 table remaining after the first column is deleted.

The values in the cells are calculated using equation (6). The problem with quasi independence occurs when the values of d and h in Table 8 are also zero; so the diagonals are also zero valued. Applying the same method would imply that the whole table must be collapsed.

3.2.2 Method 2

This method is defined using the notation of interaction contrasts:

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\delta_{ij}\beta_{ij}\log m_{ij} \tag{12}$$

($\beta_{ij} \neq 0$ for some i, j) where

$$\sum_{i=1}^{I} \delta_{ij}\beta_{ij} \quad = \quad \sum_{j=1}^{J} \delta_{ij}\beta_{ij} = 0$$

Recall that δ_{ij} is defined in equation (9), so equation (12) is then restricted to contrasts involving only cells in S. Consider equations (7) and (8), the interaction contrast (12) can be rewritten as :

$$\sum_{i=1}^{I}\sum_{j=1}^{J} \delta_{ij}\beta_{ij}u_{12(ij)}$$

($\beta_{ij} \neq 0$ for some i, j). Therefore $u_{12(ij)} = 0$ if the contrast is set to 0 for various values of $\{\beta_{ij}\}$. In other words, quasi independence is equivalent to zero for all interactive contrasts.

3.3 Sampling Distributions

3.3.1 Quasi independence model for Poisson likelihood

Earlier, a likelihood function for a two-way contingency table following a Poisson probability was stated and therefore a loglinear model was built. Similarly, in the following steps a quasi independence model for an incomplete table will be derived.

Let $\prod^{*}$ denote the product over all the cells contained in S.

Therefore, the likelihood function is given by:

$$\prod{}^{*} m_{ij}^{x_{ij}} \frac{\exp(-m_{ij})}{x_{ij}!}$$

$$L(m) \quad = \quad \sum \delta_{ij}[x_{ij}\log(m_{ij}) - m_{ij} - \log(x_{ij}!)]$$

substituting (10) into the above result gives:

$$L(m) \quad = \quad x_{++}u + \sum_{i} x_{i+}u_{1(i)} + \sum_{j} x_{+j}u_{2(j)} - m_{++} - \sum \delta_{ij}\log(x_{ij}!)$$

where the total sample size $N = x_{++}$ and the marginal totals $x_{i+}\bigvee i$ and $x_{+j}\bigvee j$ are minimal sufficient statistics.

3.4 Existence of MLE and Degrees of freedom of quasi independence

In general, the degrees of freedom are calculated by finding the difference between the number of cells in the table and the number of independent parameters. To find the degrees of freedom for quasi independence first an important theorem must be highlighted. The theorem which constitutes the necessary and sufficient conditions for the MLE to exist was derived and is given as follows [1]:

Theorem 1

Let $\boldsymbol{n}$ be the vector of cell counts, let $\boldsymbol{m}$ be their expected values. Result (1) – (5) apply to Poisson sampling and result (6) applies to multinomial sampling.

1. The log likelihood function is a strictly concave of log($\boldsymbol{m}$).
2. If a ML estimate of m exists, it is unique and satisfies the likelihood equations X'n=X'$\hat{\boldsymbol{m}}$. Conversely, if $\hat{\boldsymbol{m}}$ satisfies the model and also the likelihood equations, then it is the ML estimate of m.
3. If all the $n_i > 0$, then ML estimates of loglinear model parameters exist.
4. Suppose ML parameter estimates exist for a loglinear model that equates observed and fitted counts in certain marginal tables. Then those marginal tables have uniformly positive counts.
5. Suppose model M_2 is a special case of model M_1. If ML estimates exist for M_1, then they also exist for M_2.
6. For any loglinear model, the ML estimates $\hat{\boldsymbol{m}}$ are identical for multinomial and independent Poisson sampling, and those estimates exist in the same situations.

For a quasi independence model of an inseparable incomplete table satisfying Theorem 1, the degrees of freedom are calculated as $IJ - z_e - [I + J - 1] = (I-1) \times (J-1) - z_e$ (z_e is the number of cells containing structural zeros). It is possible to have zero degrees of freedom with the table satisfying the conditions discussed in the Theorem 1. For instance Table 11 below, will result in zero degrees of freedom:

4	6	-	-	-
-	5	7	-	-
-	-	6	3	-
-	-	-	4	5

Table 11: Incomplete table resulting in zero degrees of freedom

For a separable incomplete table with, say k subtables, the nonempty cells of each subtable are denoted as $S_1, S_2, \ldots, S_k$. The value of degrees of freedom for the quasi independence model of the whole table is the sum of the degrees of freedom of each $S_1, S_2, \ldots, S_k$. For an inseparable incomplete table there are zero degrees of freedom if and only if the cells not containing structural zeros are non-interactive. A non-interactive cell is one that does not contribute to any of the interaction contrasts.

3.5 Goodness of fit

The goodness of fit test is used to test the fit of the quasi-independence model. The test statistic is a chi-square (χ^2) random variable that is, in general, defined as:

$$\sum \frac{(O_i - E_i)^2}{E_i}$$

where O_i is the observed frequency count at the $i-th$ level of the categorical variables and E_i is the expected frequency count at the $i-th$ level of the categorical variables.

In order to apply the goodness of fit test to the quasi-independence model, the Pearson statistic is used and defined as follows:

$$\chi^2 = \sum \frac{(x_{ij} - \widehat{m}_{ij})^2}{\widehat{m}_{ij}}$$

where $\widehat{m}_{ij}$ is the MLE of m_{ij} and the sum is taken over all the cells in an incomplete subtable S. Alternatively, the likelihood-ratio test may be used and is defined as:

$$G^2 = 2 \sum x_{ij} \log(x_{ij}/\widehat{m}_{ij})$$

Under the null hypothesis of quasi independence both the test statistics given above have an asymptotic central χ^2 distribution. The degrees of freedom are calculated as show in section 3.4 above.

3.6 Connectivity and Separability in incomplete tables

3.6.1 Introduction

Connectivity and separability is found amoungst a set of non-structural zero cells. The concept of separability is important to understand in order to deal with the degrees of freedom under quasi independence. For a two-way contingency table, two cells are associated if there are no structural zeros in either and if the cells are in the same row or the same column. If every pair of cells can, in some way, be linked by a chain of other cells then there is connectivity within that pair. Since the main focus here is a two

way table, an incomplete two-way table is connected if its non-structural zero cells form a connected set. If the table is not connected then it is separable because the nonempty cells of a separable table can be divided into two or more separate subtables. Each subtable has no repeating rows or columns in it. To define separability we do not distinguish between structural and sampling zeros. Therefore it can be said that an observed table is separable if the incomplete table is not connected.

3.6.2 Dealing with separable tables

Consider an incomplete table that is separable into k subtables. In other words, the set S is made up of $S_1, S_2, \ldots, S_k$. Which means that the set of nonempty cells of the incomplete table is made up of k subsets of nonempty cells of each subtable. Since, by definition of separability, each of the subsets have no rows or columns in common, equation (6) can be written such that each of the subsets have no parameters in common. Therefore, each of the subsets has a separate quasi-independence model.

3.7 Methods to determine if a cell is non-interactive

There are other methods that may be used to obtain the MLEs for the expected cell counts of an incomplete contingency table. These are iterative procedures and they are referred to as the indirect procedures. In this paper the main focus is direct estimation procedure, so the preceding section describes the direct estimation procedure used to get the MLEs for the expected cell counts in an incomplete table under the quasi independence.

It is important to conclude whether a cell is non-interactive or not. If a cell is identified as a non-interactive cell, it is known that under quasi independence the observed counts for individual cells is the MLE for those cells. This knowledge allows for removal of these cells from the incomplete subtable during the analysis of the table and replacement after the analysis is complete. The first two principles of the procedure that will be introduced address the determination of non-interactive cells. The last two principles allow the MLEs for two classes of complete tables to be captured.

3.7.1 Principle 1 (Cell Isolates)

When $\delta_{ij} = 1$ for some cell (i, j) but still in the same row, the remaining δ_{ij} are zero, then $\widehat{m}_{ij} = x_{ij}$. This is because the MLEs are determined in a unique way, preserving of marginal total. Clearly, the relevant cell is non-interactive so the $i - th$ row and $j - th$ column is deleted. For clarity, the following example demonstrates Principle 1. Consider a 4×4 tables with expected cells given in Table 12:

m_{11}	m_{12}	m_{13}	m_{14}
m_{21}	m_{22}	m_{23}	–
m_{31}	m_{32}	m_{33}	-
–	-	-	m_{44}

Table 12: 4×4 table with expected values and zero valued cells

Deleting the fourth row results in the $(1, 4)$ cell becoming a cell isolate and it is also non-interactive so the fourth column may also be deleted. The resulting table after the deletion is given in Table 13:

m_{11}	m_{12}	m_{13}
m_{21}	m_{22}	m_{23}
m_{31}	m_{32}	m_{33}

Table 13: Resulting 3×3

3.7.2 Principle 2 (Semiseparability)

Semiseparability is a concept that implies that the incomplete table can be made separable in two or more subtables if a single row or column is removed. Now, consider a separable table that is partitioned into sets of rows. Each set corresponds to one separable subtable derived from column deletion. Estimation of expected cell counts is now made easier in each partitioned set of rows. Under quasi independence, the estimation is done in the same way as it would if each set where a separable subtable. i.e specific MLEs may be found for the table given that specific MLEs may be found for each set of rows. To illustrate principle 2 more clearly, consider the following 5×5 incomplete table with expected cell counts given in Table 14 below:

m_{11}	m_{12}	-	-	-
m_{21}	m_{22}	-	-	m_{25}
-	-	m_{33}	m_{34}	m_{35}
-	-	m_{43}	m_{44}	-
-	-	m_{53}	m_{54}	m_{55}

Table 14: 5×5 table with expected cell counts m_{ij} and zero entries

It can be concluded that the table is semiseparable because if column 5 is deleted this results in two separable subtables. Partitioning the table into the first 2 rows and the last three rows, consider the reduced subtables given in Table 15 and Table 16, respectively:

m_{11}	m_{12}	-
m_{21}	m_{22}	m_{25}

Table 15: Reduced subtable 1

m_{33}	m_{34}	m_{35}
m_{43}	m_{44}	-
m_{53}	m_{54}	m_{55}

Table 16: Reduced subtable 2

In Table 15, cell $(2, 5)$ is a cell isolate yet it was not a cell of this nature in the original table. Deleting the third column of Table 15 results in a 2×2 subtable whose MLEs maybe be calculated directly. Deletion of the third column of Table 16 also admits direct MLEs. Using the proceeding principles, principle 3 and principle 4, the direct MLEs may be pieced together to acquire the MLEs of the original incomplete table.

3.7.3 Principle 3 (Block-Triangular Tables)

An incomplete table is in "block triangular" form if after the fitting of row and column permutations, $\delta_{ij} = 0$ implies $\delta_{kl} = 0 \bigvee k \geqq i$ and $l \geqq j$. From "block triangular" forms explicit formulas for expected cell values may be determined. Table 17 , Table 18 and Table 19 illustrate different examples of "block triangular" tables.

m_{11}	m_{12}	m_{13}	m_{14}
m_{21}	m_{22}	m_{23}	
m_{31}	m_{32}		
m_{41}			

Table 17: Block triangular table 1

m_{11}	m_{12}	m_{13}	m_{14}
	m_{22}	m_{23}	m_{24}
	m_{32}	m_{33}	m_{34}
		m_{43}	m_{44}
		m_{53}	m_{54}

Table 18: Block triangular table 2

		m_{13}	m_{14}
		m_{23}	m_{24}
m_{31}	m_{32}	m_{33}	m_{34}
m_{41}	m_{42}	m_{43}	m_{44}

Table 19: Block triangular table 3

The above forms are called block triangular because the non-structural zero cells form a right angled triangle with blocks of cells lying along the hypotenuse of the triangle, given that the relevant permutations of rows and columns have been performed.

MLEs for block triangular tables

$\mathrm{M}_{\boldsymbol{I_1 \times J_1}}$	$\mathrm{M}_{\boldsymbol{I_1 \times J_2}}$	$\mathrm{M}_{\boldsymbol{I_1 \times J_3}}$
$\mathrm{M}_{\boldsymbol{I_2 \times J_1}}$	$\mathrm{M}_{\boldsymbol{I_2 \times J_2}}$	$0_{\boldsymbol{I_2 \times J_3}}$
$\mathrm{M}_{\boldsymbol{I_3 \times J_1}}$	$0_{\boldsymbol{I_3 \times J_2}}$	$0_{\boldsymbol{I_3 \times J_3}}$

Table 20: $I \times J$ block triangular table where $I = I_1 + I_2 + I_3, J = J_1 + J_2 + J_3$

With reference to Table 20 above, it is noted that the structural zeros lie in two blocks of cells, specified by setting $\delta_{ij} = 0$ for $i \geqq I_1+1$ and $j \geqq J_1+J_2+1$ and for $i \geqq I_1+I_2+1$ and $j \geqq J_1+1$. To find the MLEs of the incomplete subset S, m_{ij} must be written in its multiplicative form in order for the function to only be of marginal totals. Considering the blocks on the diagonals defined by $i = I_1 + 1, I_1 + 2, \ldots, I_1 + I_2$ and $j = J_1 + 1, J_1 + 2, \ldots, J_1 + J_2$ we have:

$$\begin{aligned} m_{i+}m_{+j} &= (\sum_{l=1}^{J_1+J_2} \exp(u + u_{1(i)} + u_{2(l)}))(\sum_{k=1}^{I_1+I_2} \exp(u + u_{1(k)} + u_{2(j)})) \\ &= \exp(u + u_{1(i)} + u_{2(j)})(\sum_{l=1}^{J_1+J_2}\sum_{k=1}^{I_1+I_2} \exp(u + u_{1(k)} + u_{2(l)})) \\ &= m_{ij}(\sum_{l=1}^{J_1+J_2}\sum_{k=1}^{I_1+I_2} m_{kl}) \end{aligned}$$

and

$$\sum_{k=1}^{J_1+J_2}\sum_{l=1}^{I_1+I_2} m_{kl} = m_{++} - \sum_{i=I_1+I_2+1}^{I_1+I_2+I_3} m_{i+} - \sum_{j=J_1+J_2+1}^{J_1+J_2+J_3} m_{+j}$$

The m_{ij} for this block can be written as direct functions of the marginal totals, hence the MLEs are given by:

$$\widehat{m}_{ij} = \frac{x_{i+}x_{+j}}{\sum_{k=1}^{J_1+J_2}\sum_{l=1}^{I_1+I_2} x_{kl}}$$

for $i = I_1 + 1, \ldots, I_1 + I_2$, $j = J_1 + 1, \ldots, J_1 + J_2$

The above formulas give the MLEs for the two diagonal blocks of cells. Now, revised marginal totals are computed for the remaining cells by subtracting the known MLEs of the three diagonal blocks from the original marginal totals and considering the already estimated cells as structural zeros in the rest of

the proceedings. The block triangle will reduce in size thereafter and the calculation will continue using the updated versions of the above formulas.

3.7.4 Principle 4 (Block-stairway Tables)

An inseparable incomplete table is deemed a "block-stairway" table if the table may be divided into sets of rows each containing one rectangular array of nonzero cells, all after the relevant permutation of rows and columns. In addition, each of these rectangular arrays shares columns only with those array immediately above it and immediately below it. Under quasi independence, block-stairway incomplete tables have closed form MLEs for nonzero expected counts. If one of the rectangular arrays in a block-stairway table has one row or one column, that one column or row may have cell isolates or it may potentially be semiseparable.

MLEs for block-stairway tables

$0_{\boldsymbol{I_1\times J_1}}$	$0_{\boldsymbol{I_1\times J_2}}$	$\mathrm{M}_{\boldsymbol{I_1\times J_3}}$	$\mathrm{M}_{\boldsymbol{I_1\times J_4}}$
$0_{\boldsymbol{I_2\times J_1}}$	$\mathrm{M}_{\boldsymbol{I_2\times J_2}}$	$\mathrm{M}_{\boldsymbol{I_2\times J_3}}$	$0_{\boldsymbol{I_2\times J_4}}$
$0_{\boldsymbol{I_3\times J_1}}$	$\mathrm{M}_{\boldsymbol{I_3\times J_2}}$	$0_{\boldsymbol{I_3\times J_3}}$	$0_{\boldsymbol{I_3\times J_4}}$
$\mathrm{M}_{\boldsymbol{I_4\times J_1}}$	$\mathrm{M}_{\boldsymbol{I_4\times J_2}}$	$0_{\boldsymbol{I_4\times J_3}}$	$0_{\boldsymbol{I_4\times J_4}}$

Table 21: $I \times J$ block stairway incomplete table where $I = I_1 + I_2 + I_3 + I_4, J = J_1 + J_2 + J_3 + J_4$

In Table 21, there are seven nonzero entries. The sum of the MLEs is equal to the sum of the observed counts. The calculation will first consider cell " $\mathrm{M}_{\boldsymbol{I_2\times J_3}}$", looking at the product of the marginal totals.

$$\begin{aligned} m_{i+}m_{+j} &= (\sum_{l=J_1+1}^{J_1+J_2+J+3} \exp(u+u_{1(i)}+u_{2(l)}))(\sum_{k=1}^{I_1+I_2} \exp(u+u_{1(k)}+u_{2(j)})) \\ &= \exp(u+u_{1(i)}+u_{2(j)})(\sum_{l=J_1+1}^{J_1+J_2+J_3}\sum_{k=1}^{I_1+I_2} \exp(u+u_{1(k)}+u_{2(l)})) \end{aligned}$$

thus

$$m_{ij} = \frac{m_{i+}m_{+j}}{(\sum_{l=J_1+1}^{J_1+J_2+J_3}\sum_{k=1}^{I_1+I_2}\exp(u+u_{1(k)}+u_{2(l)}))} \tag{13}$$

summing over i, j in the block of cells gives:

$$\frac{\sum_{i=I_1+1}^{I_1+I_2}\sum_{j=J_1+J_2+1}^{J_1+J_2+J_3} m_{i+}m_{+j}}{\sum_{i=I_1+1}^{I_1+I_2}\sum_{j=J_1+J_2+1}^{J_1+J_2+J_3} m_{ij}} = \sum_{l=J_1+1}^{J_1+J_2+J_3}\sum_{k=1}^{I_1+I_2}\exp(u+u_{1(k)}+u_{2(l)}) \quad (14)$$

Substituting result (14) in equation (13) results in an expression for m_{ij} that in terms of marginal totals $\{m_{i+}\}$ and $\{m_{+j}\}$, the following is obtained:

$$\widehat{m}_{ij} = \frac{x_{i+}x_{+j}(\sum_{l=J_1+1}^{J_1+J_2+J_3}\sum_{k=1}^{I_1+I_2} x_{kl})}{\sum_{l=J_1+1}^{J_1+J_2+J_3}\sum_{k=1}^{I_1+I_2} x_{k+}x_{+l}}$$

for $i = I_1+1,\ldots,I_1+I_2; j = J_1+J_2+1,\ldots,J_1+J_2+J_3$

3.8 Equivalence of closed-form and iterative estimates

Earlier in this paper it was noted that under quasi independence the MLEs for the expected cell counts in an incomplete table are unique and exist if the conditions stated in Theorem 1 are met. Hence, both the closed-form formulas and the iterative proportional fitting procedure are the same estimates, if the MLEs for a particular incomplete table are written in closed form.

4 Application

4.1 Example 1 (To illustrate the block-triangular procedure)

	Final					
Initial state	A	B	C	D	E	Totals
E	11	23	12	15	8	69
D	9	10	4	1	-	24
C	6	4	4	-	-	14
B	4	5	-	-	-	9
A	5	-	-	-	-	5
	35	42	20	16	8	121

Table 22: Initial and final disability of stroke patients rating

Table 22 is an extract from [4] and will be used to illustrate direct estimation when block-triangles are considered. The data summarised in Table 22 is about the severity of physical disability after a stroke of 121 patients from Massachusetts General Hospital, the grading was done on admission and again on discharge of each patient. The "block-triangular" form is formed because a patient cannot be discharged if their condition becomes worse, so their score on the discharge evaluation may only remain the same or get better.

The calculation of the expected cell values begins with consideration of the diagonal cells, for instance $(A,A),(B,B),(C,C)$ and so on. The expected count for the cell with initial state B and final state B is

calculated as follows:

$$\frac{9 \times 42}{121 - (20 + 16 + 8 + 5)} = 5.25$$

Similarly, the expected cell values for (A, A) is 5, for (C, C) is 3.37, for (D, D) is 4.52 and for (E, E) is 8. Note that (A, A) and (E, E) are cell isolates as described in Principle 1 above. Hence, their expected values are equal to the observed values. Now, the diagonal cell estimates calculated above are subtracted from the marginal totals and the procedure is repeated. The expected cell count for (C, B) is given as follows:

$$\frac{(14 - 3.37)(42 - 5.25)}{94.86 - (16.63 + 11.48) - 3.75} = 6.20$$

Similarly, the expected cell values for (B, A) ,(D, C) and (E, D) are 3.75, 4.69 and 11.48, respectively. Repeatedly doing this procedure will result in the values in Table 23 below:

	Final					
Initial State	A	B	C	D	E	Totals
A	15.66	21.92	11.94	11.48	8	69
B	6.16	8.63	4.69	4.52	-	24
C	4.43	6.20	3.37	-	-	14
D	3.75	5.25	-	-	-	9
E	5	-	-	-	-	5
Totals	35	42	20	16	8	121

Table 23: Estimated expected cell counts resulting form procedure explain above.

The goodness-of-fit statistics computed for this table are:

$$\chi^2 = 8.37$$

$$G^2 = 9.60$$

both with 6 degrees of freedom. This implies that the quasi-independence model is a relatively good fit.

4.2 Example 2 (To illustrate the block-stairway procedure)

Table 24 below, is data from two experiments done to study male Drosophila melanogaster carrying a specific translocation between the X chromosome and the minute fourth chromosome, as well as a Y chromosome and a normal fourth chromosome [4]. In the latter experiment, a random sample of males were mated with attached-X females with the distal and of the translocation. In the former, a random sample of males were mated with attached-X females with a Y chromosome. The sperm produced

by the males carries either the proximal end of translocation (A) or $Y(A')$, and either the distal end of the translocation (B) or the fourth chromosome (B'). This knowledge give the following possible combinations: (AB), ($A'B'$), ($A'B$), (AB'). The main purpose of this example, however, is to illustrate the block stairway method. The cells we consider to be structural zeros in this case are labeled "lethal" and there is scientific understanding as to why mating that combination results in a lethal result.

	Male sperm type			
Female type	AB	A'B'	A'B	AB'
FemY	1413	1029	lethal	2240
FemProx	lethal	548	346	1287

Table 24: Summary of observed data values

After rearranging the table in Table 25 it is evident that the model of quasi independence fits the data and cells (FemProx, A'B) and (FemY, AB) are cell isolates, so under quasi independence their estimates are equal to their observed counts.

	Type 1			
Type 2	A'B	A'B'	AB'	AB
FemY	-	1029	2240	1413
FemProx	346	548	1287	-

Table 25: Observed values in block-stairway table form

Cells (FemY, A'B'), (FemY, AB') and (FemProx, A'B'), (FemProx, AB') form blocks. The estimates of these blocks are calculated as follows:

$$\hat{m}_{\text{FemY,A'B'}} = \frac{(1577 \times 4682) \times 3269}{(1577 \times 4682) + (3527 \times 4682)} = \frac{1577 \times 3269}{5104} = 1010.03$$

$$\hat{m}_{\text{FemY,AB'}} = 2258.97 \text{ (by subtraction)}$$

$$\hat{m}_{\text{FemProx,A'B'}} = \frac{(1577 \times 2181) \times 1835}{(1577 \times 2181) + (3527 \times 2181)} = \frac{1577 \times 1835}{5104} = 566.97$$

$$\hat{m}_{\text{FemProx,AB'}} = 1268.03$$

The Pearson goodness-of-fit statistic computed for this table is:

$$\chi^2 = 1.43$$

with one degree of freedom. This implies that the quasi-independence model is a very good fit.

The approach that will be taken for the rest of the practical application section is that the estimation of expected cell values will be done using SAS. Firstly, the same examples (1 and 2) done above will be

programmed for the purpose of comparing the computerized answers with the answers calculated above. Secondly, another example (Example 3) will be introduced to solidify the concept.

For the first example, the estimated values from the SAS output are summarised in Table 26 below and the SAS output and code can be found in the appendix.

	Final				
Initial	A	B	C	D	E
A	15.6607	21.92498	11.93196	11.48235	8
B	6.161588	8.626223	4.694543	4.517647	
C	4.427711	6.198795	3.373494		
D	3.75	5.25			
E	5				

Table 26: Summary of expected values from SAS output for example 1.

In the second example, the estimated values from the SAS output are summaried in Table 27 below and the SAS output and code can be found in the appendix.

	Type 1			
Type 2	A'B	A'B'	AB'	AB
FemY	1413	1010.034		2258.966
FemProx		566.9661	346	1268.034

Table 27: Summary of expected values from SAS output for example 2.

It can be observed that the computerized values and the manually calculated values are exactly the same.

4.3 Example 3

The data provided in Table 28 summarises the relationship between radial asymmetry and locular composition in Staphylea.

	Coefficient of Radial Asymmetry						
Locular Composition	0.00	0.47	0.82	0.94	1.25	1.41	1.63
3 even, 0 odd	1001	-	-	263	-	-	1
2 even, 1 odd	-	744	160	-	20	4	-
1 even, 2 odd	-	340	66	-	11	1	-
0 even, 3 odd	76	-	-	17	-	-	0

Table 28: Data indicating alternatives chosen given different conditions, where "-" indicates structural zeros and "0" indicates sampling zeros.

If row 4 is removed then cells $(3even, 0odd, 0.00)$, $(3even, 0odd, 1.25)$, $(3even, 0odd, 1.63)$ form cell isolates and their corresponding columns may be removed too. Hence, cells $(2even, 1odd, 0.47)$, $(2even, 1odd, 0.82)$, $(2even, 1odd, 1.41)$, $(1even, 2odd, 0.94)$ as well as cells $(1even, 2odd, 0.47)$, $(1even, 2odd, 0.82)$, $(1even, 2odd, 1.41)$ and $(1even, 2odd, 0.94)$ are left behind, consider Table 29:

	Coefficient of Radial Asymmetry			
Locular Composition	0.47	0.82	1.41	0.94
2 even, 1 odd	744	160	20	4
1 even, 2 odd	340	66	11	1

Table 29: New table of identification of cell isolates.

The MLEs may be determined directly. The estimates are given in Table 30 below and the SAS code is given in the appendix.

	Coefficient of Radial Asymmetry			
Locular Composition	0.47	0.82	1.41	0.94
2 even, 1 odd	747.364	155.8158	21.37296	3.447251
1 even, 2 odd	336.636	70.18425	9.627043	1.552749

Table 30: Summary of expected values from SAS output for example 3.

The parameter estimates for the loglinear model under quasi independence are given as follows:
$\lambda_1^L = 0.3988$, $\lambda_1^C = 2.6253$, $\lambda_2^C = 1.0574$, $\lambda_3^C = -0.9291$

5 Conclusion

In this paper, the use of loglinear models to model cell counts for contingency tables is introduced . Even though contingency tables are useful for model building, incomplete contingency tables can be a nightmare when expected cell counts must be estimated. This is the problem that is investigated. Many different methods have been proposed for handling such a problem. However, one of the shortfalls is that due to the existence of structural zeros in the table, estimation of the individual cells may be difficult or even impossible. The conditions for the existence of maximum likelihood estimates are discussed in the paper but not much can be done if the estimates just simply do not exist. With the use of the methods for handling incomplete tables that are discussed above, useful models can be built. But, of course with more research, there is always room for improvement.

References

[1] Alan Agresti. *Categorical Data Analysis*. New York: John Wiley & Sons, 2002.

[2] Maurice S Bartlett. Contingency table interactions. *Supplement to the Journal of the Royal Statistical Society*, 2(2):248–252, 1935.

[3] MW Birch. Maximum likelihood in three-way contingency tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 220–233, 1963.

[4] Yvonne MM Bishop, Stephen E Fienberg, and W Paul. *Discrete Multivariate Analysis : Theory and Practice*. The MIT Press, 1975.

[5] George EP Box and Norman Richard Draper. *Empirical Model-building and Response Surfaces*, volume 424. Wiley New York, 1987.

[6] Stephen E Fienberg. Quasi-independence and maximum likelihood estimation in incomplete contingency tables. *Journal of the American Statistical Association*, 65(332):1610–1616, 1970.

[7] Stephen E Fienberg and Alessandro Rinaldo. Three centuries of categorical data analysis: Log-linear models and maximum likelihood estimation. *Journal of Statistical Planning and Inference*, 137(11):3430–3445, 2007.

[8] Shelby J Haberman. Log-linear models for frequency data: Sufficient statistics and likelihood equations. *The Annals of Statistics*, pages 617–632, 1973.

[9] Samarendra N Roy and Sujit K Mitra. An introduction to some non-parametric generalizations of analysis of variance and multivariate analysis. *Biometrika*, pages 361–376, 1956.

[10] SN Roy and Marvin A Kastenbaum. On the hypothesis of no 'interaction' in a multi-way contingency table. *The Annals of Mathematical Statistics*, 27(3):749–757, 1956.

[11] G Udny Yule. On the association of attributes in statistics: with illustrations from the material of the childhood society. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 194:257–319, 1900.

Appendix

5.1 SAS code to example 1

```
data research;
input initial $ final $ count @@ ;
cards;
1 1 11 1 2 23 1 3 12 1 4 15 1 5 8
2 1 9 2 2 10 2 3 4 2 4 1 2 5 .
3 1 6 3 2 4 3 3 4 3 4 . 3 5 .
4 1 4 4 2 5 4 3 . 4 4 . 4 5 .
5 1 5 5 2 . 5 3 . 5 4 . 5 5 .
;
proc catmod data=research;
weight count;
model initial*final=_response_ /
missing=structural zero=sampling
         freq pred=freq noparm oneway;
   loglin initial final;
```

The SAS System

The CATMOD Procedure

Data Summary			
Response	initial*final	Response Levels	15
Weight Variable	count	Populations	1
Data Set	RESEARCH	Total Frequency	121
Frequency Missing	0	Observations	15

One-Way Frequencies		
Variable	Value	Frequency
initial	1	69
	2	24
	3	14
	4	9
	5	5
final	1	35
	2	42
	3	20
	4	16
	5	8

Population Profiles	
Sample	Sample Size
1	121

Response Profiles		
Response	initial	final
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	2	1
7	2	2
8	2	3
9	2	4
10	3	1
11	3	2
12	3	3
13	4	1
14	4	2
15	5	1

Maximum Likelihood Analysis
Maximum likelihood computations converged.

Maximum Likelihood Analysis of Variance			
Source	DF	Chi-Square	Pr > ChiSq
initial	4	35.92	<.0001
final	4	10.60	0.0314
Likelihood Ratio	6	9.60	0.1427

Maximum Likelihood Predicted Values for Response Functions					
Function Number	Observed		Predicted		Residual
	Function	Standard Error	Function	Standard Error	
1	0.788457	0.53936	1.141717	0.492335	-0.35326
2	1.526056	0.493435	1.478189	0.482566	0.047867
3	0.875469	0.532291	0.869783	0.507027	0.005686
4	1.098612	0.516398	0.831373	0.516848	0.267239
5	0.470004	0.570088	0.470004	0.570076	-8.34E-9
6	0.587787	0.557773	0.208897	0.517366	0.37889
7	0.693147	0.547723	0.545369	0.508078	0.147778
8	-0.22314	0.67082	-0.06304	0.531366	-0.16011
9	-1.60944	1.095445	-0.10145	0.540745	-1.50799
10	0.182322	0.60553	-0.12156	0.542863	0.303877
11	-0.22314	0.67082	0.214917	0.534019	-0.43806
12	-0.22314	0.67082	-0.39349	0.556222	0.170345
13	-0.22314	0.67082	-0.28768	0.574942	0.064539
14	0	0.632456	0.04879	0.566599	-0.04879

Maximum Likelihood Predicted Values for Frequencies						
		Observed		Predicted		
initial	final	Frequency	Standard Error	Frequency	Standard Error	Residual
1	1	11	3.162278	15.6607	2.893058	-4.6607
1	2	23	4.316028	21.92498	3.438984	1.075018
1	3	12	3.287844	11.93196	2.636105	0.068037
1	4	15	3.624982	11.48235	2.785858	3.517647
1	5	8	2.733327	8	2.733162	-6.26E-8
2	1	9	2.886274	6.161588	1.501709	2.838412
2	2	10	3.028787	8.626223	1.926491	1.373777
2	3	4	1.966664	4.694543	1.277767	-0.69454
2	4	1	0.995859	4.517647	1.310416	-3.51765
3	1	6	2.387986	4.427711	1.301729	1.572289
3	2	4	1.966664	6.198795	1.719087	-2.1988
3	3	4	1.966664	3.373494	1.07273	0.626506
4	1	4	1.966664	3.75	1.311381	0.25
4	2	5	2.189381	5.25	1.762995	-0.25
5	1	5	2.189381	5	2.189395	2.61E-9

Figure 1: Output from SAS containing the expected frequency results for example 1

5.2 SAS code to example 2

```
data research2;
input female $ male $ count @@;
cards;
1 1 1413 1 2 1029 1 3 . 1 4 2240
2 1 . 2 2 548 2 3 346 2 4 1287
;
proc catmod data=research2;
weight count;
model female*male=_response_/
missing=structural zero=sampling
          freq pred=freq noparm oneway;
```

```
loglin female male;
```

The SAS System

The CATMOD Procedure

Data Summary			
Response	female*male	Response Levels	6
Weight Variable	count	Populations	1
Data Set	RESEARCH2	Total Frequency	6863
Frequency Missing	0	Observations	6

One-Way Frequencies		
Variable	Value	Frequency
female	1	4682
	2	2181
male	1	1413
	2	1577
	3	346
	4	3527

Population Profiles	
Sample	Sample Size
1	6863

Response Profiles		
Response	female	male
1	1	1
2	1	2
3	1	4
4	2	2
5	2	3
6	2	4

Response Frequencies						
	Response Number					
Sample	1	2	3	4	5	6
1	1413	1029	2240	548	346	1287

Maximum Likelihood Analysis
Maximum likelihood computations converged.

Maximum Likelihood Analysis of Variance			
Source	DF	Chi-Square	Pr > ChiSq
female	1	391.88	<.0001
male	3	1075.01	<.0001
Likelihood Ratio	1	1.44	0.2305

Maximum Likelihood Predicted Values for Response Functions					
Function Number	Observed		Predicted		Residual
	Function	Standard Error	Function	Standard Error	
1	0.093401	0.038532	0.108248	0.03661	-0.01485
2	-0.22373	0.041819	-0.22748	0.042054	0.003757
3	0.554162	0.034978	0.57744	0.02917	-0.02328
4	-0.85379	0.051008	-0.80492	0.030293	-0.04887
5	-1.31363	0.060557	-1.29878	0.059353	-0.01485

Maximum Likelihood Predicted Values for Frequencies						
female	male	Observed		Predicted		Residual
		Frequency	Standard Error	Frequency	Standard Error	
1	1	1413	33.4975	1413	33.4975	0
1	2	1029	29.57562	1010.034	24.70737	18.96611
1	4	2240	38.84445	2258.966	35.5597	-18.9661
2	2	548	22.45536	566.9661	16.40719	-18.9661
2	3	346	18.12612	346	18.12612	0
2	4	1287	32.33655	1268.034	27.97883	18.96611

Figure 2: Output from SAS containing the expected frequency results for example 2.

5.3 SAS code to example 3

```
data research3;
input Locular $ coefficient $ count @@;
cards;

1 1 744 1 2 160 1 3 20 1 4 4
2 1 340 2 2 66 2 3 11 2 4 1
;
proc catmod data=research3;
weight count;
model Locular*coefficient=_response_/
missing=structural zero=sampling
```

```
      freq pred=freq oneway;
loglin Locular coefficient;
```

The SAS System

The CATMOD Procedure

Data Summary			
Response	Locular*coefficient	Response Levels	8
Weight Variable	count	Populations	1
Data Set	RESEARCH3	Total Frequency	1346
Frequency Missing	0	Observations	8

One-Way Frequencies		
Variable	Value	Frequency
Locular	1	928
	2	418
coefficient	1	1084
	2	226
	3	31
	4	5

Population Profiles	
Sample	Sample Size
1	1346

Response Profiles		
Response	Locular	coefficient
1	1	1
2	1	2
3	1	3
4	1	4
5	2	1
6	2	2
7	2	3
8	2	4

Response Frequencies								
	Response Number							
Sample	1	2	3	4	5	6	7	8
1	744	160	20	4	340	66	11	1

Maximum Likelihood Analysis
Maximum likelihood computations converged.

Maximum Likelihood Analysis of Variance			
Source	DF	Chi-Square	Pr > ChiSq
Locular	1	183.31	<.0001
coefficient	3	912.47	<.0001
Likelihood Ratio	3	1.00	0.8009

Analysis of Maximum Likelihood Estimates					
Parameter		Estimate	Standard Error	Chi-Square	Pr > ChiSq
Locular	1	0.3988	0.0295	183.31	<.0001
coefficient	1	2.6253	0.1237	450.14	<.0001
	2	1.0574	0.1306	65.53	<.0001
	3	-0.9291	0.1760	27.87	<.0001

Maximum Likelihood Predicted Values for Response Functions					
	Observed		Predicted		
Function Number	Function	Standard Error	Function	Standard Error	Residual
1	6.612041	1.000672	6.176526	0.452095	0.435515
2	5.075174	1.00312	4.608647	0.455952	0.466526
3	2.995732	1.024695	2.6221	0.485516	0.373633
4	1.386294	1.118034	0.79755	0.058906	0.588744
5	5.828946	1.00147	5.378975	0.448241	0.44997
6	4.189655	1.007547	3.811097	0.452131	0.378558
7	2.397895	1.044466	1.824549	0.481929	0.573346

Maximum Likelihood Predicted Values for Frequencies						
		Observed		Predicted		
Locular	coefficient	Frequency	Standard Error	Frequency	Standard Error	Residual
1	1	744	18.24157	747.364	16.9474	-3.36404
1	2	160	11.87353	155.8158	9.874934	4.18425
1	3	20	4.438786	21.37296	3.814325	-1.37296
1	4	4	1.997026	3.447251	1.540075	0.552749
2	1	340	15.94101	336.636	14.39673	3.364042
2	2	66	7.922357	70.18425	5.124527	-4.18425
2	3	11	3.303045	9.627043	1.753193	1.372957
2	4	1	0.999628	1.552749	0.695979	-0.55275

Figure 3: Output from SAS containing the expected frequency results for example 3.

The use of multilevel modeling to assess the mathematics achievement of Grade 9 learners in South Africa for the TIMSS 2015 data

Tshidiso Thebe 12370861

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr Gretel Crafford

Department of Statistics, University of Pretoria

30 October 2017 (final)

Abstract

In this paper, the reader is familiarised with the concept of hierarchical data and statistical models which arise from data with such a structure, i.e. multilevel models. Multilevel models treat the regression coefficients as random variables which act as a "link" between the different levels of the hierarchical structured data, [13, 10]. Different types of models can be formulated from the basic multilevel model, namely the random-intercept model the and random varying slope model. Furthermore, the application of the multilevel model, namely the level-2 model, will be illustrated/fitted using the recent Trends in International Mathematics and Science Study (TIMSS) 2015 data to analyze the performance of individual students in mathematics, using the SAS statistical package.

Declaration

I, Tshidiso Thebe, declare that this essay, submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics, at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____

Tshidiso Thebe

_____

Dr Gretel Crafford

_____

30 October 2017

Acknowledgements

I, Tshidiso Thebe, would like to thank the National Research Foundation for making it possible for me to able to pursue my studies, and Dr. Gretel Crafford for her guidance and patience throughout the year.

Contents

List of Figures

List of Tables

1 Introduction

In the social, behavioral and educational sciences data often has a hierarchical structure, with individuals nested within a network in an organized system. In the educational system of South Africa, students are nested within classes, classes are nested within schools, and schools are nested within regions etc. This is a typical example of a hierarchical structure. Data with such a structure occurs naturally within any organization and each level has certain characteristics which may have an influence on the other levels of the hierarchical structure. For instance, mathematics teachers in different schools use different methods of teaching the subject and will therefore have some influence on the performance of the students in the subject. Such an influence is therefore significant and unique to each school, and thus cannot be ignored when modeling the mathematics performance of students (level 1). Multilevel model analysis incorporates influences, from different levels in order to predict correct estimates of the standard errors that influence different hypothesis tests that can be performed.

[2] illustrates that multilevel modeling offers an improvement as compared to the classical regression methods when dealing with hierarchical data and concludes that for prediction, multilevel models can be essential, for casual inference they can be helpful, and useful for data reduction.

Consider a hierarchical data set, with students (level 1 units) nested within schools (level 2 units), the basic 2-level model can be expressed in a number of ways, with the most convenient way being to model both levels separately and then substituting the two to obtain a combined model, [12]. The model at the first level can be expressed as

$$y_{ij} = b_{0i} + b_{1i}x_{ij} + \varepsilon_{ij}, \quad i = 1, 2, ..., N \text{ and } j = 1, 2, ..., n_i, \tag{1}$$

where y_{ij} is the response variable, x_{ij} is the independent variable at the first level and ε_{ij} is the error term associated with the j-th student from the i-th school. The second level model can be expressed as

$$b_{0i} = \beta_0 + \gamma_{01}z_i + u_{0i}, \quad i = 1, 2, ..., N, \tag{2}$$

$$b_{1i} = \beta_1 + \gamma_{11}z_i + u_{1i}, \quad i = 1, 2, ..., N, \tag{3}$$

where b_{0i} is the intercept model of the i-th school, b_{1i} is the model of the slope of the i-th school. The level-2 predictor, z_i, is the i-th school's independent variable. The error terms u_{0i} and u_{1i} are associated with the i-th school for the intercept- and slope models, respectively, denote the random variation of the coefficients. Substituting (2) and (3) into (1), the resulting model called the combined model, and is

expressed as

$$y_{ij} = \beta_0 + \gamma_{01} z_i + \beta_1 x_{ij} + \gamma_{11} z_i x_{ij} + x_{ij} u_{1i} + u_{0i} + \varepsilon_{ij}, \quad i = 1, 2, ..., N \text{ and } j = 1, 2, ..., n_i. \tag{4}$$

Model (4) incorporates predictors from both levels (student level and school level), in order to model the response variable y_{ij}, which is the mathematics achievement for a Grade 9 student. The Trends in International Mathematics and Science Study (TIMSS) 2015 data is a collection of data from schools in 57 countries, and in this sampled data, a hierarchical structure is exhibited. Using TIMSS 2015 data, the South African grade 9 maths marks will be modeled with a 2-level model and subsequently, various models will be used in order to detect the different relationships in the marks. PROC MIXED in SAS will be used in order to fit the various models for statistical inference purposes.

2 Background Theory

2.1 Origin of multilevel models

A hierarchy consists of individuals of a lower level nested within groups or organization at a higher level(s). [7] describe multilevel models, also known as hierarchical linear models, as statistical models that contain variables at different levels of a hierarchy. The term "hierarchical linear models" was coined by [8], when they used the models to analyze data with a hierarchical structure through Bayesian estimation. [11], through his studies in ecological processes was one of the first researchers to recognize the need for multilevel analysis. In 1961, Paul Lazarsfeld and Herbert Menzel developed a topology which described the relations between variables belonging to different levels in a hierarchical structure, which was further enhanced to levels within individuals by Johan Galtung in 1969, [6].

This topology made it easier to distinguish the level to which the measurements belonged, and describes the creation of related variables by aggregation and dis-aggregation. The lowest level in the structure consists of measurements from individuals and is called the micro-level and the corresponding higher levels are called macro-levels, also referred to as groups or contexts, [7]. Although not much progress was made, the interest in analyzing and interpreting hierarchical structured data arose in the 1970's when there was a surge in theoretical and statistical discussions related to educational and sociological research, [6]. [1], expanded the concept of the topology presented Lazarsfeld and Menzel with path diagrams in order to show relationships between variables and units belonging to levels in a hierarchical structures and hence making it simpler to model multilevel models.

Social sciences involve the study of a population, and subsequently the data will possess a naturally occurring hierarchy due to the "clustering" of human populations (similar to educational sciences), hence

the need for multilevel analysis. During the 1970's, Robert Hauser disputed the need for contextual analysis by claiming that the contextual effects were merely "grouped individual effects", and lacked substance to influence response variables. Prior to the development of multilevel models, the statistical methods available to analyze hierarchical structured data were flawed as they ignored the different levels or treated them inadequately, [6]. *Multilevel Models in Educational and Social Sciences* by [4], *Multilevel Statistical Models* by [5] and *Hierarchical Linear Models* by [9], were the initial texts that accelerated the rise in researching multilevel models and have since been used as main references in multilevel analysis.

The model introduced in equation 4 is the basic two-level model with the students being the micro-level and the schools forming the macro-level. [8] used the Bayesian approach in order to estimate the covariance components but their efforts made little progress as the models required estimation of unbalanced data. Various estimation approaches have since been offered including the maximum likelihood estimation (MLE), restricted maximum likelihood estimation (REML) and iterative generalized least-squares estimation, [13]. The birth of computers and subsequently the development of statistical packages HLM and MLwiN by [9] and [5], respectively, for fitting multilevel models gave researchers the comfort of estimating parameters and their corresponding errors. Numerous other statistical programs have since been developed including the PROC MIXED procedure available in SAS. Prior to the development of statistical computer packages, which brought some ease to the analysis of hierarchical structured data, progress was minimal due to the complexity involved in the estimation techniques.

Researchers, equipped with the power of statistical software, have done extensive improvements to multilevel analysis. Multilevel analysis has since become popular as it allows researchers from various fields to model contextual factors that influence the response variables of interest.

2.2 The Model

Consider the basic regression model of the mathematics achievement of the j-th student in the i-th school y_{ij}, for a single school with a single explanatory variable x_{ij},

$$y_{ij} = b_0 + b_1 x_{ij} + \varepsilon_{ij}, \quad i = 1, 2, ..., N \text{ and } j = 1, 2, ..., n_i \tag{5}$$

where

- b_0 is the expected math achievement for the school given that the explanatory variable, x_{ij}, assumes a value of zero.
- b_1 is the slope of the model.

- and ε_{ij} the random error term unique to each student with distributional assumptions $\varepsilon_{ij} \sim iidN(0, \sigma^2)$.

This regression model only includes the student's explanatory variables and does not take into consideration the school level or "external" predictors which may influence the student's performance in mathematics, for instance, influences like the method a certain school uses to teach. Such influences may be significant to the performance of students within schools. Figure 1 shows the regression lines fitted for 5 schools chosen from the TIMSS 2015 data. These regression lines show the relationship between math achievement and the students confidence in mathematics.

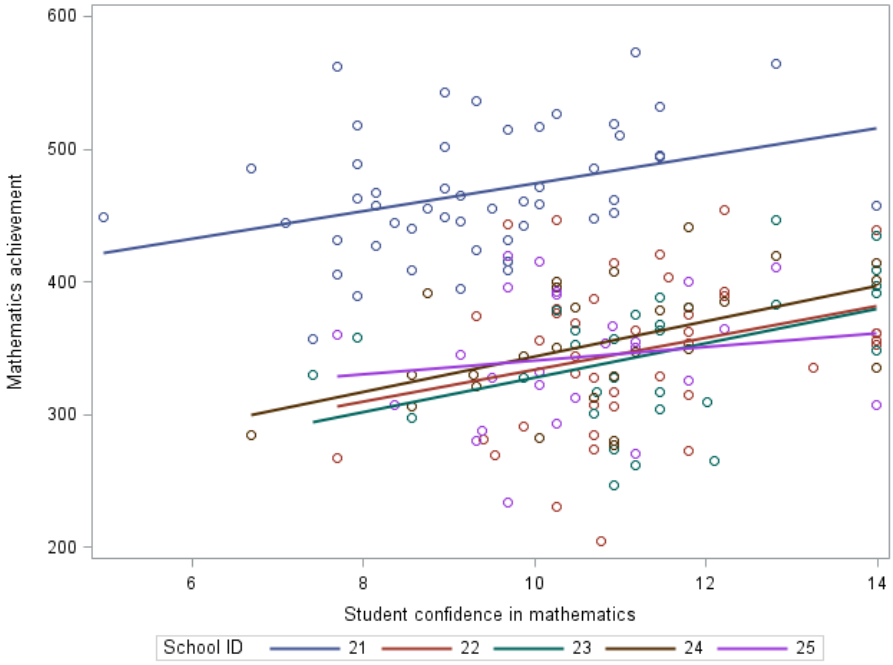


Figure 1: Regression lines for five schools

The graph shows that there is a huge variation in mathematics marks between the schools and fitting a basic regression line for all schools would result in enormous errors in the estimates as there is a huge spread in the data points plotted. For instance, considering school 23 and 24, the regression lines seem to have the same slopes, i.e. the same relationship between math achievement and confidence in math, but students in school 24 are expected to achieve higher marks than a student in school 23 with the same confidence in mathematics (higher intercept), therefore there must be some school level variable that causes the students in school 24 to perform slightly better, but basic regression methods do not account for this type of variation. In order to solve this problem, [9] suggests that this variation between schools can be modeled using multilevel models that include school-level explanatory variables that cause the variation in mathematics achievement between schools.

Now consider that there is now data from N number of schools available. By incorporating school-level

variables the model will now be multilevel, with two levels, due to the students being nested within the schools.[12] states there are at least three different ways to express this multilevel model, firstly by writing separate equations for the different levels; secondly, by writing the separate equations at different levels and then substituting into the equation at the lowest level to end up with a single equation; and lastly, by writing a single equation that specifies the multiple sources of variation. Model (5) can be expressed as a multilevel model with the level-1 model

$$y_{ij} = b_{0i} + b_{1i}x_{ij} + \varepsilon_{ij}, \quad i = 1, 2, ..., N \text{ and } j = 1, 2, ..., n_i, \tag{6}$$

or equivalently in matrix form

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i, \tag{7}$$

where $\boldsymbol{y}_i = \begin{pmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{in_i} \end{pmatrix}$, $\boldsymbol{X}_i = \begin{pmatrix} 1 & x_{i1} \\ 1 & x_{i2} \\ \vdots & \vdots \\ 1 & x_{in_i} \end{pmatrix}$, $\boldsymbol{b}_i = \begin{pmatrix} b_{0i} \\ b_{1i} \end{pmatrix}$ and $\boldsymbol{\epsilon}_i = \begin{pmatrix} \varepsilon_{i1} \\ \varepsilon_{i2} \\ \vdots \\ \varepsilon_{in_i} \end{pmatrix}$

- y_{ij} is the j^{th}student in the i^{th} school, which is the response variable being investigated, i.e mathematics achievement.
- b_{0i} is the intercept of the i^{th} school.
- b_{1i} is the slope of the i^{th} school.
- ε_{ij} is the error term associated with the j^{th} student in the i^{th} school, furthermore $\varepsilon_{ij} \sim iidN(0, \sigma^2)$.

The regression coefficients in (6) are treated as random variables in order to distinguish between the different parameters in which the different schools may take [13], i.e. two different schools may have different mean performance in the subject at hand, thus the intercepts vary across schools. These random coefficients (level 2 units) can be expressed as a level 2 model

$$b_{0i} = \beta_0 + \gamma_{01}z_i + u_{0i}, \quad i = 1, 2, ..., N \tag{8}$$

$$b_{1i} = \beta_1 + \gamma_{11} z_i + u_{1i}, \quad i = 1, 2, ..., N. \tag{9}$$

or equivalently in matrix form

$$\boldsymbol{b}_i = \boldsymbol{Z}_i \boldsymbol{\gamma} + \boldsymbol{u}_i \tag{10}$$

where $\boldsymbol{Z}_i = \begin{pmatrix} 1 & z_i & 0 & 0 \\ 0 & 0 & 1 & z_i \end{pmatrix}$, $\boldsymbol{\gamma} = \begin{pmatrix} \beta_0 \\ \gamma_{01} \\ \beta_1 \\ \gamma_{11} \end{pmatrix}$ and $\boldsymbol{u}_i = \begin{pmatrix} u_{0i} \\ u_{1i} \end{pmatrix}$

- z_i is a level 2 explanatory variable unique to the i^{th} school.
- u_{0i} and u_{1i} are error terms associated with the i^{th} school and they are assumed to have a joint distribution with mean $\mathbf{0}$ and co-variance matrix $\boldsymbol{\Phi}$= $\begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix}$.

Substituting (8) and (9) into (6), the model becomes

$$y_{ij} = \beta_0 + \gamma_{01} z_i + u_{0i} + \beta_1 x_{ij} + \gamma_{11} z_i x_{ij} + x_{ij} u_{1i} + \varepsilon_{ij}, \quad i = 1, 2, ..., N \text{ and } j = 1, 2, ..., n_i \tag{11}$$

or equivalently in matrix form

$$\begin{aligned} \boldsymbol{y}_i &= \boldsymbol{X}_i(\boldsymbol{Z}_i \boldsymbol{\gamma} + \boldsymbol{u}_i) + \boldsymbol{\epsilon}_i \\ &= \boldsymbol{X}_i \boldsymbol{Z}_i \boldsymbol{\gamma} + \boldsymbol{X}_i \boldsymbol{u}_i + \boldsymbol{\epsilon}_i \end{aligned} \tag{12}$$

with

$$E(\boldsymbol{y}_i) = \boldsymbol{X}_i \boldsymbol{Z}_i \boldsymbol{\gamma} \tag{13}$$

and

$$Cov(\boldsymbol{y}_i, \boldsymbol{y'}_i) = \boldsymbol{X}_i \boldsymbol{\Phi} \boldsymbol{X}_i^{'} + \sigma^2 \boldsymbol{I}_{n_i}. \tag{14}$$

Further distributional assumption is that the student level errors and school level errors are uncorrelated, i.e. $Cov(\boldsymbol{\epsilon}_i, \boldsymbol{u}_i^{'}) = \mathbf{0}$. Models (7) and (10) form the basic models at level 1 and level 2, respectively, and substitution will lead to the combined model (12). The first term, $\boldsymbol{X}_i\boldsymbol{Z}_i\boldsymbol{\gamma}$ in (12) is called the fixed component and the second term, $\boldsymbol{X}_i\boldsymbol{u}_i + \boldsymbol{\epsilon}_i$ is called the random component of the model. [10] states that models of higher levels can be expressed using the above logic.

2.2.1 Centering

In multilevel modeling, taking a two-level model for instance, the intercept and slopes in the level-1 model are the outcomes of the level-2 model. The intercept of the level-1 model depends on the location of the level-1 predictor variables, and in order to interpret the relationship between a response variable and a given predictor variable, [10] suggest using a centering technique. Centering enhances the model to be able to correctly depict the level-1 intercept, when the predictor variable takes on a value of zero. There are two types of centering that [10] focus on, namely, group-mean centering and grand-mean centering, to show that the location of the level-1 predictors impacts the interpretation of the models. Group-mean centering involves subtracting the relevant mean of the level-2 group from an observation in that group, therefore the centered observations to be involved in the model are of the form $x_{ij} - \bar{x}_i$, where $\bar{x}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_{ij}$, is the mean of the i^{th} school for the level-1 predictor variable, x.

Consider a basic regression model from one school of the form, $y_{ij} = \beta_0 + \beta_1 x_{ij} + \varepsilon_{ij}$, where y, is the response variable (Math achievement) and x, the predictor variable (student confidence in mathematics). Figure 2, shows the fitted regression line with β_0 and β_1, being the intercept and slope for this school, respectively. The intercept, β_0, is the math achievement of a student given that the student's confidence in mathematics takes a value of zero, but in this case it cannot be interpreted as it is not clear where the intercept lies. The problem that arises can be solved by the technique of centering the data around the school's mean, [10].

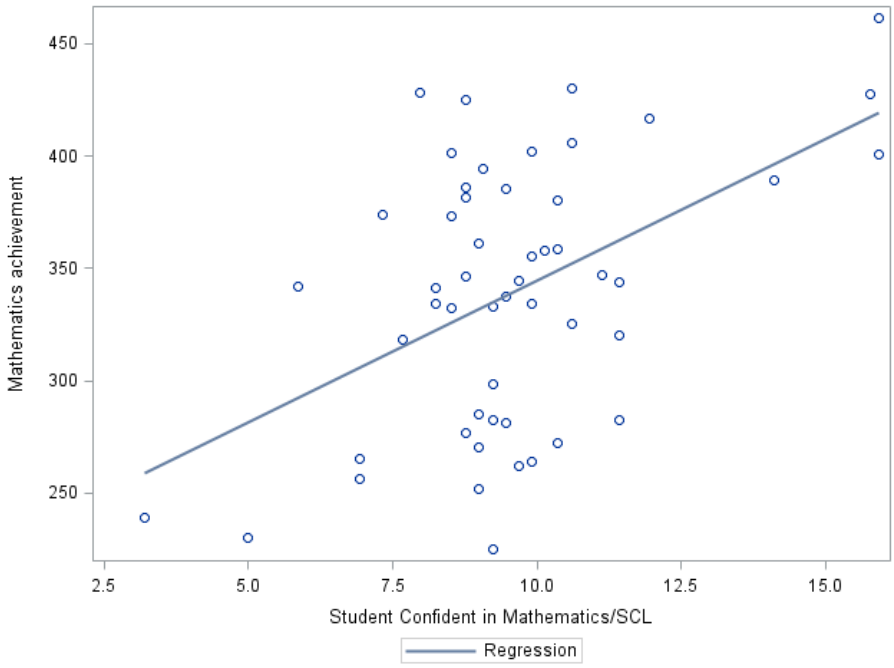


Figure 2: Relationship between math achievement and student confidence in mathematics

Now consider the model described above being centered around the school mean, i.e. $y_{ij} = \beta_0 + \beta_1(x_{ij} - \bar{x}_{i.}) + \varepsilon_{ij}$. Figure 3, now shows that the intercept has shifted, and hence can be interpreted correctly, meaning that given a value of zero (student confidence equal to the school mean), a student will achieve the corresponding intercept value, β_0. Furthermore, it can be noted that the slope of the model, β_1, in Figure 2 and 3 remains the same, hence centering does not influence the relationship between the two variables, it just makes the interpretation simpler.

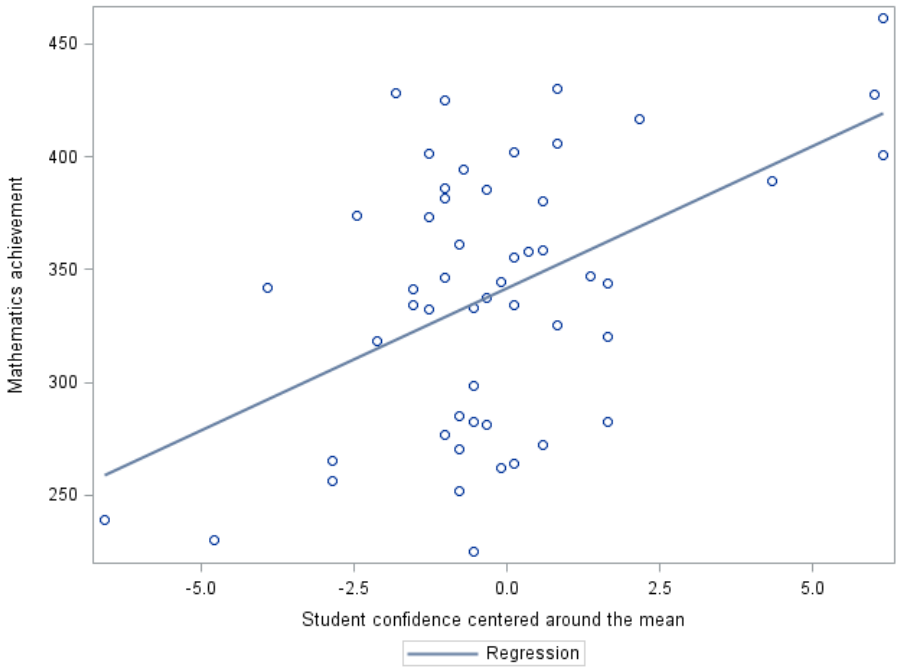


Figure 3: Relationship between math achievement and student confidence in mathematics

Another form of centering is "grand-mean centering" which follows the same logic but subtracts the overall mean of the predictor variable from the individual observations, hence the centered observations included in the model are $x_{ij} - \bar{\bar{x}}$, where $\bar{\bar{x}} = \frac{1}{N}\sum_{i=1}^{N} \bar{x}_i$, is the grand-mean of the predictor variable, x. Centering is therefore an important technique in multilevel modeling because of its ability to let us interpret the intercept of the model. Furthermore, as noted by [12], group mean centering may be applied to level-1 units while grand mean centering is applied to level-2 units. Figure 4 shows the regression lines for the five school with the students confidence centered.

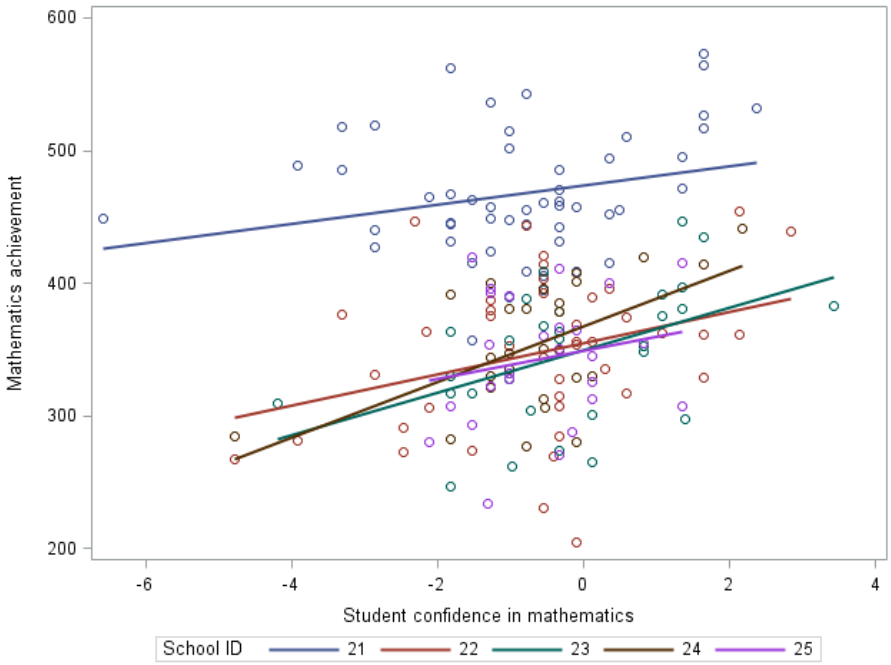


Figure 4: Regression lines for the five schools

The five schools now have a clear intercept and the slopes are still the same as before. Displaying the 292 schools in the TIMSS 2015 data on a graph would result in a similar plot.

2.2.2 Types of models

Different types of models (sub-models) can be formulated by setting certain terms in the basic model equal to zero, depending on the type of tests a researcher is interested in performing. The sub-models fall into two types of categories, namely the random-intercept models and random varying slope models. The random-intercept models include, (i) the one-way analysis of variance (ANOVA) with random effects, (ii) the means-as-outcomes model, (iii) the one-way analysis of co-variance (ANCOVA) model and (iv) the non-randomly varying slopes model ([10]). The randomly varying slope models include, the random-coefficients regression model, as well as, the intercepts- and slopes-as-outcomes model ([10]). These six models will now be explained further in order to explain the basic underlying rationale behind multilevel models.

Model 1: One-way ANOVA with random effects

Considering basic two-level model hierarchical linear model given in (6), (8) and (9), and setting the slope coefficient, b_{1i}, equal to zero for all i, the resulting with the level 1 model is:

$$y_{ij} = b_{0i} + \varepsilon_{ij}, \quad i = 1, 2, ..., N \text{ and } \ j = 1, 2, ..., n_i, \tag{15}$$

with the assumption that each level-1 error term, ε_{ij}, is normally distributed with mean zero and variance, σ^2. The level-2 model, with γ_{01} set equal to zero yields the random intercept model as:

$$b_{0i} = \beta_0 + u_{0i}, \tag{16}$$

and hence substituting (16) into (15), the combined model becomes:

$$y_{ij} = \beta_0 + u_{0i} + \varepsilon_{ij}, \tag{17}$$

with β_0 as the grand mean in the population of schools, u_{0i}, the error term for the i-th school and ε_{ij} the error associated with the j-th student from the i-th school levels. It is also assumed that u_{0i} is normally distributed with mean zero and variance τ_{00}. Model (17) is called the one-way ANOVA with random effects, also referred to as the null model. It can also be noted that the outcome y_{ij}, has the grand-mean, β_0, and variance, $\tau_{00} + \sigma^2$, since

$$E(y_{ij}) = E(\beta_0 + u_{0i} + \varepsilon_{ij}) = \beta_0, \quad \text{and} \tag{18}$$

$$Var(y_{ij}) = Var(\beta_0 + u_{0i} + \varepsilon_{ij}) = Var(u_{0i} + \varepsilon_{ij}) = \tau_{00} + \sigma^2. \tag{19}$$

This combined model is also referred to as fully unconditional as it possesses no predictors both on the first- and second-level models. Another parameter that is quite useful for this model is called the "intraclass correlation coefficient", which measures the proportion of the variance in the outcome that is due to the variance between the schools (level-2 units), and is defined as

$$\rho = \frac{\tau_{00}}{\tau_{00} + \sigma^2}. \tag{20}$$

Model 2: Means-as-outcomes regression model

Considering the basic two-level model with no level-1 predictors, a sub-model called the means-as-outcomes model can be formed by setting the slope model, b_{1i}, equal to zero and including at least one level-2 (school level) predictor, say z_i in the intercept model, the level-2 model then becomes:

$$b_{0i} = \beta_0 + \gamma_{01} z_i + u_{0i}, \tag{21}$$

and the combined model yielded is then expressed as:

$$y_{ij} = \beta_0 + \gamma_{01} z_i + u_{0i} + \varepsilon_{ij}. \tag{22}$$

This model essentially considers the effects that the school level predictor has on the outcome or response variable. This leads to a variance that is conditional due to the inclusion of the predictor, z_i, since u_{0i} is of the form, $u_{0i} = b_{0i} - \beta_0 - \gamma_{01} z_i$, it follows that the variance between schools, τ_{00}, is the variance in b_{0i}, after controlling for the predictor variable z_i.

Model 3: ANCOVA model

As before, considering the basic two-level model, another sub-model can be formed by setting the level-2 coefficients, γ_{01}, γ_{11}, and random effects, u_{1i}, equal to zero, [10]. By including a level-1 predictor centered around the grand mean, the combined model becomes,

$$y_{ij} = \beta_0 + \gamma_{10}(x_{ij} - \bar{\bar{x}}) + u_{0i} + \varepsilon_{ij}. \tag{23}$$

This model is called the one-way ANCOVA with random effects. The variance between students is now conditional, due to the inclusion of the level-1 predictor, x_{ij}.

Model 4: Non-randomly varying slopes model

Consider the basic level-1 model with a predictor centered around the group mean (school level mean), i.e. $y_{ij} = b_{0i} + b_{1i}(x_{ij} - \bar{x}_{i.}) + \varepsilon_{ij}$, with $\bar{x}_{i.} = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij}$ being the average of the predictor in the i-th school. Furthermore, including a level-2 predictor, say z_i, and setting the random effects in the slope model, u_{1i}, equal to zero, results in the following level-2 model:

$$b_{0i} = \beta_0 + \gamma_{01} z_i + u_{0i} \tag{24}$$

$$b_{1i} = \beta_1 + \gamma_{11} z_i. \tag{25}$$

By substituting the level-2 model into the level-1 model yields the following combined model:

$$y_{ij} = \beta_0 + \gamma_{01} z_i + \beta_1(x_{ij} - \bar{x}_{i.}) + \gamma_{11} z_i (x_{ij} - \bar{x}_{i.}) + u_{0i} + \varepsilon_{ij}. \tag{26}$$

This model is called the non-randomly varying slopes model, because even though the slopes may vary from school to school due to the inclusion of the level-2 predictor, their variation is nonrandom due to the constrained school level random effects slope variable, u_{1i}.

Model 5: Random-coefficients regression model

Considering the basic level-1 model with one predictor variable centered around the group mean, i.e. $y_{ij} = b_{0i} + b_{1i}(x_{ij} - \bar{x}_{i.}) + \varepsilon_{ij}$, with $\bar{x}_{i.}$ as described above. Constraining γ_{01} and γ_{11} to be null, the level-2 model can now be expressed as:

$$b_{0i} = \beta_0 + u_{0i}, \tag{27}$$

$$b_{1i} = \beta_1 + u_{1i}, \tag{28}$$

and the combined model yielded is called the random-coefficients model given by,

$$y_{ij} = \beta_0 + \beta_1(x_{ij} - \bar{x}_{i.}) + u_{1i}(x_{ij} - \bar{x}_{i.}) + u_{0i} + \varepsilon_{ij}. \tag{29}$$

[10], also notes that the variation between the schools in this model can be viewed as being unconditional, due to the absence of level-2 predictors, and has the co-variance matrix:

$$Cov(\boldsymbol{u}_i, \boldsymbol{u}'_i) = \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix} = \boldsymbol{\Phi}, \tag{30}$$

where τ_{00} is the unconditional variance in the level-1 intercepts, τ_{11}, the unconditional variance in the level-1 slopes and $\tau_{10} = \tau_{01}$, the unconditional co-variance between level-1 intercepts and slopes.

Model 6: Intercepts and slopes-as-outcomes

This model can be viewed as the full basic model as it includes both level-1 and level-2 predictors. The level-1 and level-2 models are given by

$$y_{ij} = b_{0i} + b_{1i}x_{ij} + \varepsilon_{ij}, \quad i = 1, 2, ..., N \text{ and } j = 1, 2, ..., n_i \tag{31}$$

and

$$b_{0i} = \beta_0 + \gamma_{01}z_i + u_{0i}, \quad i = 1, 2, ..., N, \tag{32}$$

$$b_{1i} = \beta_1 + \gamma_{11}z_i + u_{1i}, \quad i = 1, 2, ..., N. \tag{33}$$

The combined model is then formulated by substituting (32) and (33) into (31), and is expressed as:

$$y_{ij} = \beta_0 + \gamma_{01}z_i + \beta_1 x_{ij} + \gamma_{11}z_i x_{ij} + x_{ij}u_{1i} + u_{0i} + \varepsilon_{ij}, \quad i = 1, 2, ..., N \text{ and } j = 1, 2, ..., n_i. \tag{34}$$

The term $\beta_0 + \gamma_{01}z_i + \beta_1 x_{ij} + \gamma_{11}z_i x_{ij}$, is the fixed component and the second term, $x_{ij}u_{1i} + u_{0i} + \varepsilon_{ij}$, is the random component of the model.

2.3 Estimation theory

This part of the report describes the parameter estimation of the two-level model and is divided into three sections to outline the techniques used to estimate the different components of the model. For a two-level hierarchical model, there are three types of parameters that can be estimated, namely, the fixed effects, the random level-1 coefficients and the co-variance components. [10] mentions that the estimation of each

of the parameters depends on the others. The basic two-level model will be used in order to illustrate the estimation techniques used by [10].

1. **Fixed effects estimation**

Using matrix notation the general level-1 model with Q predictors can be expressed as:

$$\boldsymbol{y}_i = \boldsymbol{X}_i \boldsymbol{b}_i + \boldsymbol{\epsilon}_i, \tag{35}$$

where $\boldsymbol{y}_i$ is an n_i by 1 vector of outcomes, $\boldsymbol{X}_i$ is an n_i by (Q+1) matrix of Q predictor variables with the vector $\mathbf{1}$, in the first column, $\boldsymbol{b}_i$ is a (Q+1) by 1 vector of the unknown parameters and $\boldsymbol{\epsilon}_i$ is an n_i by 1 vector of the random errors. Furthermore, it is assumed that $\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \sigma^2 \boldsymbol{I}_{n_i})$, i.e. the random errors are identically and independently normally distributed with mean 0, and variance of σ^2. Assuming that $\boldsymbol{X}_i$ is full rank, the ordinary least squares (OLS) estimator of $\boldsymbol{b}_i$ is given by

$$\hat{\boldsymbol{b}}_i = (\boldsymbol{X}_i^{'} \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i^{'} \boldsymbol{y}_i, \tag{36}$$

and its dispersion matrix is given by

$$Var(\hat{\boldsymbol{b}}_i) = \sigma^2 (\boldsymbol{X}_i^{'} \boldsymbol{X}_i)^{-1} = \boldsymbol{V}_i. \tag{37}$$

A model for the estimated parameters, $\hat{\boldsymbol{b}}_i$, can be built by pre-multiplying equation 35 with $(\boldsymbol{X}_i^{'} \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i^{'}$. The model for $\hat{\boldsymbol{b}}_i$ is

$$\hat{\boldsymbol{b}}_i = \boldsymbol{b}_i + \boldsymbol{e}_i, \tag{38}$$

where $\boldsymbol{e}_i \sim N(\mathbf{0}, \boldsymbol{V}_i)$ and the dispersion of $\hat{\boldsymbol{b}}_i$ as an estimate of $\boldsymbol{b}_i$ is indicated by the error-variance matrix, $\boldsymbol{V}_i$. The general level-2 model with F predictors is given by

$$\boldsymbol{b}_i = \boldsymbol{Z}_i \boldsymbol{\gamma} + \boldsymbol{u}_i, \qquad \boldsymbol{u}_i \sim N(\mathbf{0}, \boldsymbol{\Phi}), \tag{39}$$

where $\boldsymbol{Z}_i$ is a (Q+1) by F matrix of predictors, $\boldsymbol{\gamma}$ is an F by 1 vector of fixed effects, $\boldsymbol{u}_i$ is a (Q+1) by 1 vector of school-level errors. The combined model for $\hat{\boldsymbol{b}}_i$ can be yielded by the substitution of (39) into (38), i.e.

$$\hat{\boldsymbol{b}}_i = \boldsymbol{Z}_i \boldsymbol{\gamma} + \boldsymbol{u}_i + \boldsymbol{e}_i \tag{40}$$

and the dispersion of $\hat{\boldsymbol{b}}_i$, given $\boldsymbol{Z}_i$, is

$$Var(\hat{\boldsymbol{b}}_i) = Var(\boldsymbol{u}_i + \boldsymbol{e}_i) = \boldsymbol{\Phi} + \boldsymbol{V}_i = \boldsymbol{\Delta}_i \tag{41}$$

which the sum of the parameter dispersion and the error dispersion. These estimates are computed for unbalanced data, i.e. for data with unequal number of students in schools. [10] note that given that the data is unbalanced, the values for $\boldsymbol{\Delta}_i$ will differ from school to school, and then a unique, minimum-variance and unbiased estimator of $\boldsymbol{\gamma}$ is the generalized least squares (GLS) estimator

$$\hat{\boldsymbol{\gamma}} = (\sum_{i=1}^{N} \boldsymbol{Z}_i^{'} \boldsymbol{\Delta}_i^{-1} \boldsymbol{Z}_i)^{-1} \sum_{i=1}^{N} \boldsymbol{Z}_i^{'} \boldsymbol{\Delta}_i^{-1} \hat{\boldsymbol{b}}_i \tag{42}$$

given that each $\boldsymbol{\Delta}_i$ is known.

2. **Estimation of random level-1 coefficients**

As discussed in the estimation of the fixed effects, the OLS estimator of $\boldsymbol{b}_i$ is given by equation 36, based on data from the i-th school. Given the school's characteristics in the matrix $\boldsymbol{Z}_i$, a second estimator of $\boldsymbol{b}_i$ is the predicted value

$$\hat{\hat{\boldsymbol{b}}}_i = \boldsymbol{Z}_i \hat{\boldsymbol{\gamma}}, \tag{43}$$

where $\hat{\boldsymbol{\gamma}}$ is the GLS estimate in equation 42. Using empirical Bayes, the optimal combination of both of these estimators is

$$\boldsymbol{b}_i^* = \boldsymbol{\Lambda}_i \hat{\boldsymbol{b}}_i + (\boldsymbol{I}_{n_i} - \boldsymbol{\Lambda}_i) \boldsymbol{Z}_i \hat{\boldsymbol{\gamma}}, \tag{44}$$

where

$$\boldsymbol{\Lambda}_i = \boldsymbol{\Phi}(\boldsymbol{\Phi} + \boldsymbol{V}_i)^{-1}$$

is the multivariate reliability matrix, i.e. the proportion of the parameter dispersion matrix for $\boldsymbol{b}_i$ in the total dispersion matrix for $\hat{\boldsymbol{b}}_i$. More weight is put on the estimates $\hat{\boldsymbol{b}}_i$, if they are more reliable. The reliability of the estimates are calculated using the proportion of the estimate's variance in the total dispersion.

3. **Estimation of co-variance components**

Estimation of the variance and co-variance components unbalanced data requires iterative numerical procedures in order to obtain efficient estimates. Maximum likelihood methods are commonly used, mainly the full maximum likelihood (MLF) and the restricted maximum likelihood (REML). The basic concept of maximum likelihood is that the estimates of $\boldsymbol{\gamma}$, $\boldsymbol{\Phi}$ and σ^2 are chosen such that the probability of observing the observed values $\boldsymbol{y}$ is a maximum. The distinct difference between the MLF and REML estimation methods can be illustrated by considering the traditional regression model

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + ... + \beta_Q x_{Qi} + r_i, \quad r_i \sim N(0, \sigma^2)$$

for $i = 1, 2, ..., n$. The MLF estimator for σ^2 is

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} r_i}{n} \tag{45}$$

and the corresponding REML estimator for σ^2 is

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} r_i}{n - Q - 1}. \tag{46}$$

The distinct difference between MLF and REML is that the latter corrects for degrees of freedom that is lost in estimating the residual $\hat{r}_i = y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{1i} - \hat{\beta}_2 x_{2i} - ... - \hat{\beta}_Q x_{Qi}$, where each $\hat{\beta}_q$ is the OLS estimate ([10]). Furthermore, [3] proved that iterative generalized least squares will produce the same estimates as REML.

3 Application

The TIMSS 2015 data contains a sample of 12514 students from 292 schools in South Africa. Using the TIMSS 2015 data, the theory on multilevel modeling mentioned above is used, in particular the two-level model where the learners are regarded as level-1 units and the schools as level-2 units, to analyze the mathematics marks for grade 9 students in South Africa. The response variable, y, in all the models in the application is the mathematics achievement of grade 9 learners in South Africa. The international benchmark for mathematics is 500 units, and in this section it will be determined how South African students performed and what influences their performance. PROC MIXED in the SAS statistical package is used to fit the different models in order to perform hypothesis tests that will allow interpretation and analysis of the data. Table 1 and 2 give a description of the data used in the analysis of the TIMSS 2015 data set. A sub-sample of 8040 students from 196 schools was used in order to perform analyses using data with no missing values. Furthermore, the student- and school-level variables chosen for the application of the theory of the 2-level model have been proven to be significant for modeling the mathematics achievement of grade 9 students and the results are displayed in Figure 18 and 21 of the appendix.

Variable code	Description	Type	
SLTest	Student speaks language of test at home	Ordinal	1 : Always 0 : Sometimes -1: Never
GCSLM	Student likes mathematics (grand-mean centered)	Continuous	
GCSCM	Student confidence in mathematics (grand-mean centered)	Continuous	

Table 1: Student-level variables

Variable	Description	Type	
SCHEcoDisadv	Students background : Economic disadvantage	Ordinal	$1:$ if $0-25\%$ $0:$ if $26-50\%$ $-1:$ if $>50\%$
CLTest	Percentage of students speaking language of test	Ordinal	$-1:$ if $<25\%$ $0:$ if $26-75\%$ $1:$ if $>76\%$
MeanDisc	Schools discipline centered around the grand mean	Continuous	

Table 2: School-level variables

3.1 The null model

Model 1: The two-level model that can be used to estimate the average mathematics achievement for grade 9 learners in South Africa is the one-way ANOVA with random effects which can be defined by the level-1 model

$$y_{ij} = b_{0i} + \varepsilon_{ij}, \tag{47}$$

where $\varepsilon_{ij} \sim N(0, \sigma^2)$, and the level-2 model as:

$$b_{0i} = \beta_0 + u_{0i}, \tag{48}$$

where $u_{0i} \sim N(0, \tau_{00})$, with the intercepts b_{0i}, varying across schools. The null model includes no predictors at both levels and the combined model is expressed as,

$$y_{ij} = \beta_0 + u_{0i} + \varepsilon_{ij} \tag{49}$$

with β_0 as the grand mean of grade 9 mathematics marks across South Africa, u_{0i} the random effects of the i^{th} school in South Africa and ε_{ij}, the random effects of the j^{th} student in the i^{th} school in South Africa. The results of the fixed and random effects of the model are displayed in Table 3 and Table 4, respectively.

Effect	Estimate	Standard Error	p-value
Intercept	384.20	4.6745	<0.0001

Table 3: Model 1 fixed effects

Variance components	Subject	Estimate
Intercept	IDSCHOOL	4189.67
Residual		3225.04

Table 4: Model 1 random effects

The intercept estimated, which is the grand mean of the grade 9 mathematics marks, in Table 3 has a standard error of 4.6745, and a p-value of <0.0001, and at a 1% significance level, it can be concluded that

the average mathematics achievement in South Africa is significantly different from zero. It can therefore be concluded that the estimated value of $\hat{\beta}_0 = 384.20$, is the grand mean of grade 9 mathematics marks across schools in South Africa, giving all schools equal weight. Furthermore, the variation in mathematics marks between schools is 4189.67, and the variation of mathematics achievement of students within schools is 3225.04. This suggests that there is a higher variation of mathematics marks between schools than within schools, which shows that there are schools performing way better than others in South Africa. The proportion of the variance that is due to the variation between schools is determined by

$$\hat{\rho} = \frac{\hat{\tau}_{00}}{\hat{\tau}_{00} + \hat{\sigma}^2} = \frac{4189.67}{4189.67 + 3225.04} = 0.5650,$$

hence 56.5% of the total variation is explained by the variation between schools. Therefore there is a vast difference between school performances in South Africa and students within schools perform similar (not as much of a difference).

3.2 Models with school-level predictors only

Model 2.1: Due to the large variation in the mathematics marks between schools, level-2 predictors can be added to the null model in order to quantify the effects or variation between schools average math achievement that is explained by the relevant school-level predictors. The level-1 model considered in this part of the application remains the same as (47) and the level-2 model becomes,

$$b_{0i} = \beta_0 + \gamma_{01} MeanDisc_i + u_{0i}, \tag{50}$$

where $MeanDisc_i$ is the $i-th$ school's discipline rating centered around the grand mean, in order to make the intercept more interpret-able. Table 5 and 6 present the fixed and random components, respectively, for the combined model,

$$y_{ij} = \beta_0 + \gamma_{01} MeanDisc_i + u_{0i} + \varepsilon_{ij}, \tag{51}$$

which is obtained by substituting (50) into (47).

Effect	Estimate	Standard Error	p-value
Intercept	381.75	4.3519	<.0001
MeanDisc	16.4250	2.8597	<.0001

Table 5: Model 2-1 fixed effects

Variance Component	Subject	Estimate
Intercept	IDSCHOOL	3583.85
Residual		3225.09

Table 6: Model 2.1 random effects

Table 5 shows the fixed effect estimates of the fitted model, the intercept β_0, is estimated to be 381.75, and the coefficient of $MeanDisc$, is estimated to be $\hat{\gamma}_{01} = 16.4250$, both being significant to the model as the p-values of both estimates are less than the 5% level of significance. Therefore, there is a relationship between math achievement and $MeanDisc$, and a 1-unit increase in $MeanDisc$, will result in an increase of 16.4250 in math achievement.

The variance components estimated, as shown in Table 6 are 3583.85 and 3225.09 for the between school τ_{00}, and the students within schools, σ^2, respectively. In comparing with the threshold estimates of the null model, there is a reduction in the variation between school average math achievement, from 4189.67 to 3583.85, which shows us that a portion of the variation between schools math achievement is explained by the school's discipline, i.e. the variation is conditional upon the school-level predictor, $MeanDisc$. In addition, the proportion in variation between school average math achievement that's explained by the schools discipline is

$$\frac{4189.67 - 3583.85}{4189.67} = 0.1446,$$

which means that 14.46% of the variation between school average math achievement is due to the schools discipline. This is a significantly large percentage, which shows that a school's discipline plays an important role in how well a school will perform, but the model can be enhanced in order to detect more variables that explain the variation between schools.

Model 2.2: In order to detect more causes of variation, the model (51) is enhanced by adding a few categorical variables from the school level predictors in Table 2. The combined model to be fitted is,

$$y_{ij} = \beta_0 + \gamma_{01} MeanDisc_i + \gamma_{02} CLTest_i + \gamma_{03} SCHEcoDisadv_i + u_{0i} + \varepsilon_{ij}, \tag{52}$$

where all variables are detailed in Table 2. Table 7 shows that all the predictors are significant to the model (all p-values <0.0001).

Effect	Estimate	Standard Error	p-value
Intercept	417.25	4.5520	<0.0001
MeanDisc	6.8826	2.3715	0.0041
CLTest	21.5238	4.3509	<0.0001
SCHEcoDisadv	43.3937	5.2032	<0.0001

Table 7: Model 2.2 fixed effects

Variance component	Estimate	p-value
UN(1,1)	2088.44	<0.0001
Residual	3225.24	<0.0001

Table 8: Model 2.2 random effects

The within schools variation, $\hat{\sigma^2}$, remains unconditional due to no student-level predictors in the model and has not changed much as compared to the estimate of the null model. The proportion of the total variation in average mathematics marks between schools that is explained by the predictors included in model (52) is

$$\frac{4189.67 - 2088.44}{4189.67} = 0.5015,$$

which is higher than that of model (51), and is therefore a better model in to explain the variation between schools. The inclusion of only three level-2 predictors caused the between school variation to drop drastically. The same logic of including more school-level predictors to the model can be used in order to explain a higher proportion of variation between schools.

Furthermore, the fitted model reveals that schools that have less than 25% of their school population from an economically disadvantaged background and more than 76% of their students speaking the language that the tests are given, will have an expected mathematics of 482.1675, which is 97.9675 units higher than the South African average math mark.

3.3 Models with student-level predictors only

Model 3.1: In order to detect causes of variation in mathematics marks within schools, level-1 predictors are included to the null model. The effect of how much a student likes mathematics has on their results is modeled by including the student-level predictor, $GCSLM$ to the null model (47), yields the level-1 model,

$$y_{ij} = b_{0i} + b_{1i}GCSLM_{ij} + \varepsilon_{ij}, \tag{53}$$

where $\varepsilon_{ij} \sim N(0, \sigma^2)$, for $i = 1, 2, ..., 252$ and $j = 1, 2, ..., n_i$. The level-2 model is expressed as,

$$b_{0i} = \beta_0 + u_{0i}, \tag{54}$$

$$b_{1i} = \beta_1 + u_{1i}, \tag{55}$$

where $\begin{pmatrix} u_{0i} \\ u_{1i} \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix}\right]$, and substitution of the level-2 model into the level-1 yields the combined model,

$$y_{ij} = \beta_0 + \beta_1 GCSLM_{ij} + u_{1i}GCSLM_{ij} + u_{0i} + \varepsilon_{ij} \tag{56}$$

with the fixed component represented by $\beta_0 + \beta_1 GCSLM_{ij}$, and the last three terms in (56) representing the random component of the model.

Effect	Estimate	Standard Error	p-value
Intercept	384.56	4.8472	<0.0001
GCSLM	9.0395	0.3737	<0.0001

Table 9: Model 3.1 fixed effects

Variance component	Subject	Estimate	Standard error	p-value
UN(1,1)	IDSCHOOL	4518.48	468.70	<0.0001
UN(2,1)	IDSCHOOL	-44.4711	27.8149	0.1099
UN(2,2)	IDSCHOOL	1.5522	2.7310	0.2849
Residual		2977.83	48.1310	<0.0001

Table 10: Model 3.1 random effects

The estimates of the fixed effects of the model (56) are displayed in Table 9 and indicate that the average school mean for mathematics after controlling for student's preference of mathematics is 384.56, which does not differ greatly from the results of the null model. The estimate for the coefficient of $GCSLM$ is 9.0395, which means that a 1-unit increase in a student's preference for maths will result in an increase of 9.0395 in their mathematics marks and subsequent increases in the student's preference of mathematics will lead to further increments of their math marks in multiples of 9.0395. The low p-values for the estimates of the intercept, $\hat{\beta}_0$ and of $\hat{GCSLM}$ are low (both less than a 5% significance level), which leads us to conclude that there is a statistically significant relationship between a student's preference of maths and their maths marks.

The co-variance parameter estimates from Table 10, can be written in matrix form,

$$\begin{pmatrix} \hat{\tau}_{00} & \hat{\tau}_{01} \\ \hat{\tau}_{10} & \hat{\tau}_{11} \end{pmatrix} = \begin{pmatrix} 4518.48 & -44.4711 \\ -44.4711 & 1.5522 \end{pmatrix}, \tag{57}$$

with $\hat{\tau}_{00}$ showing that the variation between the intercepts is 4518.48 and $\hat{\tau}_{11}$ showing that the variation between slopes is 1.5522 across schools. The co-variance between the intercepts and slopes across schools is -44.4711. The estimate $\hat{\sigma^2}$ is 2977.83, which is conditional after controlling for the student's preference of mathematics. The proportion of variation in the maths marks within schools that is explained by the inclusion of the student's preference of mathematics is

$$\frac{\hat{\sigma}^2_{model1} - \hat{\sigma}^2_{model3.1}}{\hat{\sigma}^2_{model1}} = \frac{(3225.04 - 2977.83)}{3225.04} = 0.0767,$$

which means that 7.67% of the variation in the maths marks within schools is explained by how much a student likes mathematics.

Model 3.2: Consider the level-1 model,

$$y_{ij} = b_{0i} + b_{1i}GCSCM_{ij} + b_{2i}SLTest_{ij} + \varepsilon_{ij} \tag{58}$$

where $GCSCM_{ij}$ is the $j-th$ student's confidence in mathematics in the $i-th$ school, $SLTest_{ij}$ is how often a particular student speaks the language of the test at home, and $\varepsilon_{ij} \sim N(0, \sigma^2)$. The level-2 model is given by,

$$b_{0i} = \beta_0 + u_{0i}, \tag{59}$$

$$b_{1i} = \beta_1 + u_{1i}, \tag{60}$$

$$b_{2i} = \beta_2 + u_{2i}, \tag{61}$$

and the substitution of (59), (60) and (61) into (58) leads to the combined model,

$$y_{ij} = \beta_0 + \beta_1 GCSCM_{ij} + u_{1i}GCSCM_{ij} + \beta_2 SLTest_{ij} + u_{2i}SLTest_{ij} + u_{0i} + \varepsilon_{ij} \tag{62}$$

where $Cov \begin{pmatrix} u_{0i} \\ u_{1i} \\ u_{2i} \end{pmatrix} = \mathbf{\Phi} = \begin{pmatrix} \tau_{00} & \tau_{01} & \tau_{02} \\ \tau_{10} & \tau_{11} & \tau_{12} \\ \tau_{20} & \tau_{21} & \tau_{22} \end{pmatrix}$. The model (62) represents the relationship between the students average mathematics marks and how confident a student is in the subject and how frequent the student speaks the language that math tests are given.

Table 11 and 12 display the SAS results from fitting model (62) with PROC MIXED and certain improvements compared to model (56) be seen.

Effect	Estimate	Standard error	p-value
Intercept	380.03	4.4522	<0.0001
GCSCM	10.1710	0.3778	<0.0001
SLTest	13.3226	1.5269	<0.0001

Table 11: Model 3.2 fixed effects

Variance component	Subject	Estimate	Standard error	p-value
UN(1,1)	IDSCHOOL	3749.45	409.13	<0.0001
UN(2,1)	IDSCHOOL	38.3745	23.9714	0.1094
UN(2,2)	IDSCHOOL	3.6747	2.3564	0.0594
UN(3,1)	IDSCHOOL	-52.1667	105.83	0.6221
UN(3,2)	IDSCHOOL	-3.6150	7.6045	0.6345
UN(3,3)	IDSCHOOL	84.6014	39.2017	0.0155
Residual		2835.39	46.0649	<0.0001

Table 12: Model 3.2 random effects

Firstly, from Table 12, the estimate $\hat{\sigma^2}$ is 2835.39, which is the variation in average mathematics

marks within schools after controlling for the students confidence in mathematics and how frequent the student speaks the language the test is given. The proportion of variation in mathematics marks within schools with the inclusion of the level-1 predictors, $GCSCM$ and $SLTest$, is

$$\frac{\hat{\sigma^2}_{model1} - \hat{\sigma^2}_{model3.2}}{\hat{\sigma^2}_{model1}} = \frac{(3225.04 - 2835.39)}{3225.04} = 0.1208,$$

hence 12.08% of the variation in average mathematics marks within schools is explained by the student's confidence in mathematics and how often the student speaks the language that the tests are written. Furthermore, Table 12 shows that only the variation between schools intercepts ($\hat{\tau}_{00} = 3749.45$ with p-value <0.0001) and the variation between school slopes ($\hat{\tau}_{22} = 84.6014$ with p-value $= 0.0155$) are statistically significant to the model, when testing for significance at a 5% level of significance.

The estimates of the fixed effects are displayed in Table 11, with the intercept, $\hat{\beta}_0 = 380.03$ being the expected mathematics marks for schools, the coefficients,$\hat{\beta}_1 = 10.1710$ and $\hat{\beta}_2 = 13.3226$ for the student's confidence in mathematics ($GCSCM$) and frequency of speaking the language of the test ($SLTest$), respectively. Hence the fitted model is,

$$\hat{y}_{ij} = 380.03 + 10.1710GCSCM_{ij} + 13.3226SLTest_{ij}, \tag{63}$$

which shows that for a 1-unit increase in a student's confidence will result in an increase of 10.171 in the mathematics marks from the grand mean of 380.03. Subsequently, the three fitted models will be,

$$\hat{y}_{ij} = 380.03 + 10.1710GCSCM_{ij} + 13.3226, \tag{64}$$

$$\hat{y}_{ij} = 380.03 + 10.1710GCSCM_{ij}, \tag{65}$$

and

$$\hat{y}_{ij} = 380.03 + 10.1710GCSCM_{ij} - 13.3226, \tag{66}$$

for students who always speak the language, sometimes speak the language and who never speak the language of test, respectively. Therefore, students who always speak the language that the math test is written have an expected math achievement of 393.3526 units, while students who never speak the language will get a score of 366.7074. From these results, it is evident that the option of writing the tests in your home language (combined with confidence in mathematics) will lead to a substantial increase in the performance of South Africa in the TIMSS ratings.

3.4 Models with both student- and school-level predictors

Model 4.1: In order to show the effects of both the level-1 and level-2 predictors on the grade 9 mathematics results, models can be formulated including the specified predictors from the two level. The student-level predictor, $GCSLM$, and the school-level predictor, $MeanDisc$, are included in the model in order to analysis the significance of the combination of a school's discipline and how much a student within the school likes mathematics has on the overall mathematics marks of South African students. The level-1 model is expressed as,

$$y_{ij} = b_{0i} + b_{1i}GCSLM_{ij} + \varepsilon_{ij}, \tag{67}$$

where $\varepsilon_{ij} \sim N(0, \sigma^2)$, for $i = 1, 2, ..., 252$ and $j = 1, 2, ..., n_i$. The level-2 model is,

$$b_{0i} = \beta_0 + \gamma_{01}MeanDisc_i + u_{0i}, \tag{68}$$

$$b_{1i} = \beta_1 + \gamma_{11}MeanDisc_i + u_{1i}, \tag{69}$$

where $\begin{pmatrix} u_{0i} \\ u_{1i} \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix}\right]$, for $i = 1, 2, ..., 252$. The combined model becomes,

$$y_{ij} = \beta_0 + \beta_1 GCSLM_{ij} + \gamma_{01}MeanDisc_i + \gamma_{11}(GCSLM_{ij} * MeanDisc_i) + u_{1i}GCSLM_{ij} + u_{0i} + \varepsilon_{ij}. \tag{70}$$

The results for the fixed components and random components of model (70) are displayed in Table 13 and Table 14, respectively.

Effect	Estimate	Standard Error	p-value
Intercept	382.02	4.5365	<0.0001
GCSLM	9.0235	0.3666	<0.0001
MeanDisc	16.7711	2.9801	<0.0001
GCSLM*MeanDisc	0.5843	0.2142	0.0228

Table 13: Model 4.1 fixed effects

Variance component	Estimate	p-value
UN(1,1)	3909.18	<0.0001
UN(2,1)	-61.4018	0.0146
UN(2,2)	0.7833	0.3844
Residual	2977.82	<0.0001

Table 14: Model 4.1 random effects

The SAS PROC MIXED results of the model (70) show that at a 5% significant level, all the coefficients

are significant to the model. The fitted model can therefore be written as,

$$\hat{y}_{ij} = 382.02 + 9.0235GCSLM_{ij} + 16.7711MeanDisc_i + 0.5843GCSLM_{ij} * MeanDisc_i, \tag{71}$$

which shows that there is a positive relationship between math achievement and student-level variable, $GCSLM$, and also between math achievement and school-level variable, $MeanDisc$, while keeping the other variable constant. It can be further noted that both $GCSLM$ and $MeanDisc$ have a mean value of zero, since they are centered around the grand mean of the whole sample of schools, therefore the average mathematics mark is 382.02 (which does not differ greatly from the null model estimate).

The variance parameter estimate of τ_{11} in Table 14, has a p-value greater than the 5% significance level, which means that the null hypothesis that the slopes do not vary across schools cannot be rejected. The variance components of the intercepts, however, remains significantly different from zero (p-value <0.0001) and the co-variance between intercepts and slopes is also significantly different from zero, with the estimates being $\hat{\tau}_{00} = 3909.18$ and $\hat{\tau}_{01} = -61.4018$. Therefore there is a negative relationship between the intercepts and the slopes.

Model 4.2: Consider the level-1 model,

$$y_{ij} = b_{0i} + b_{1i}GCSLM_{ij} + \varepsilon_{ij}, \tag{72}$$

where $GCSLM_{ij}$ is a level-1 predictor that measures how much the j-th student in the i-th school likes mathematics and it is centered around the grand-mean, in order to be able to interpret the intercept and $\varepsilon_{ij} \sim N(0, \sigma^2)$. The level-2 model to be considered in this section is,

$$b_{0i} = \beta_0 + \gamma_{01}MeanDisc_i + \gamma_{02}CLTest_i + u_{0i}, \tag{73}$$

$$b_{1i} = \beta_1 + \gamma_{11}MeanDisc_i + \gamma_{12}CLTest_i + u_{1i}, \tag{74}$$

where the level-2 predictor $MeanDisc_i$ is the i-th school's discipline rating centered around the grand-mean and $CLTest_i$ is the dummy variable that depicts the percentage of students in the i-th school that speak the language that the math test is written as described in Table 2. Furthermore, the error terms are $\begin{pmatrix} u_{0i} \\ u_{1i} \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix}\right]$ distributed. Table 15 and 16 display the fixed and random components for the combined model.

Effect	Estimate	Standard error	p-value
Intercept	399.01	4.6175	<0.0001
GCSLM	8.8070	0.4327	<0.0001
MeanDisc	13.5654	2.6692	<0.0001
CLTest	36.6461	4.9275	<0.0001
GCSLM*MeanDisc	0.6391	0.2576	0.0131
GCSLM*CLTest	-0.5568	0.4711	0.2373

Table 15: Model 4.2 fixed effects

Variance component	Subject	Estimate	Standard error	p-value
UN(1,1)	IDSCHOOL	3033.00	319.00	<0.0001
UN(2,1)	IDSCHOOL	-46.2466	22.2432	0.0376
UN(2,2)	IDSCHOOL	0.5344	2.6583	0.4203
Residual		2878.21	48.1351	<0.001

Table 16: Model 4.2 random effects

From the results in Table 15, the average math achievement among schools is estimated to be $\hat{\beta}_0 = 399.01$, and the estimates for the coefficients of $GCSLM$, $MeanDisc$, and $CLTest$ are 8.8070, 13.5654 and 36.6461, respectively. These estimates are significant to the model with p-values <0.0001 and the fitted model follows as,

$$\hat{y}_{ij} = 399.01 + 8.807GCSLM_{ij} + 13.5654MeanDisc_i + 36.6461CLTest_i + 0.6391GCSLM_{ij} * MeanDisc_i. \tag{75}$$

Using the fact that the variable $CLTest$ takes on the values -1,0, or 1 for schools with the percentage of students speaking the language less than 25%, between 26 and 75%, and above 76% respectively, three models can be fitted for the three categories. The three models are,

$$\hat{y}_{ij} = 362.3639 + 8.807GCSLM_{ij} + 13.5654MeanDisc_i + 0.6391GCSLM_{ij} * MeanDisc_i, \tag{76}$$

for schools that have less than 25% of their students speaking the language of the test,

$$\hat{y}_{ij} = 399.01 + 8.807GCSLM_{ij} + 13.5654MeanDisc_i + 0.6391GCSLM_{ij} * MeanDisc_i. \tag{77}$$

for schools that have between 26 and 75% of their students speaking the language of the test, and

$$\hat{y}_{ij} = 435.6561 + 8.807GCSLM_{ij} + 13.5654MeanDisc_i + 0.6391GCSLM_{ij} * MeanDisc_i. \tag{78}$$

for schools that have more than 76% of their students speaking the language of the test. The coefficient of the cross-product, $GCSLM_{ij} * CLTest_i$ is not significant to the model as it has a p-value equal to 0.2373, and therefore the null hypothesis that the coefficient is equal to zero cannot be rejected on a 5%

level of significance. From (76) and (78) it can be seen that schools that have more than 76% of their students speaking the language that mathematics is tested are expected to perform 73.2922 units better than schools with less than 25% of their students speaking the language.

Table 16 displays the results for the random component of the model. The within school variance estimate, $\hat{\sigma}^2 = 2878.21$ (with p-value <0.0001) is conditional after controlling for student's like of mathematics. The proportion of variation in mathematics marks within schools that is due to the inclusion of the level-1 predictor $GCSLM$ is

$$\frac{\hat{\sigma^2}_{model1} - \hat{\sigma^2}_{model4.2}}{\hat{\sigma^2}_{model1}} = \frac{(3225.04 - 2878.21)}{3225.04} = 0.1075,$$

which means that almost 11% of the mathematics marks within schools can be explained by how much the students like maths. The variation in mathematics marks between school intercepts, $\hat{\tau}_{00} = 3303.00$ (with p-value <0.0001) is significantly different from zero, the proportion of variation in mathematics marks between school intercepts that is due to the level-2 predictors in the model is,

$$\frac{4189.67 - 3303.00}{4189.67} = 0.2116,$$

hence 21.16% of the variation. The co-variance between the intercepts and slopes of school, $\hat{\tau}_{01} = -46.2466$, has a p-value equal to 0.0376 and is therefore significantly different from zero at a 5% level of significance test. Furthermore, it can be concluded that the variance of slopes between schools for this model is zero, due to the fact that at a 5% level of significance, the null hypothesis that the variation of slopes is equal to zero cannot be rejected, as the estimate $\hat{\tau}_{11} = 0.5344$, has a p-value equal to 0.4203.

4 Conclusion

This research project is on multilevel models, which are models that are applied on data that has a distinct hierarchical structure. As stated by [2], multilevel models provide more accurate estimates for data with a hierarchical structure and are therefore more reliable than using traditional regression methods. The theory on a 2-level model is extensively covered, also illustrating how to build models of a higher order. There are different types of sub-models that can be constructed from the basic model, depending on the type of tests a researcher is interested in performing. Estimation of the model is rigorous, due to the fact that the model has fixed and random components which need to be estimated. Restricted maximum likelihood estimation is used for the estimation of the random component and least squares estimation for the fixed component of the multilevel model. The power of statistical software makes it easier to estimate the models and hence form a huge role in the development of studying hierarchical data sets.

Multilevel models can be used as a tool in the study of variation.

The 2-level model is applied to the grade 9 TIMSS 2015 data for the mathematics marks in order to find possible causes the variation in mathematics marks within and between South African schools. The data used has the students serving as level-1 units and schools as level-2 units. The null model is used to estimate the average mathematics mark across schools in South Africa for grade 9 students. Student-level predictors are included in the model in order to find causes of variation within schools, and school-level predictors are included in order to find the between school variation.

The variation in mathematics marks between schools over-weighs the within school variation and most of this variation is caused by the schools discipline and the percentage of students that speak the language in which the test is written. More school-level predictors can be used in order to find more causes of variation, and in turn certain intervention measures can be taken to address the poor performance of students. It can also be noted that schools with a higher percentage of students from a disadvantaged background perform in mathematics worse than schools with a lesser percentage of disadvantaged students.

The literature on multilevel models is quite extensive and cannot see how it can be improved, but the literature on programming models of higher orders can be improved.

References

[1] P.J Curran and D.J Bauer. Building path diagrams for multilevel models. *Psychological Methods*, 12(3):283–297, 2007.

[2] A. Gelman. Multilevel (hierarchical) modeling: what it can and cannot do. *Technometrics*, 48(3), August 2006.

[3] H. Goldstein. Multilevel mixed linear model analysis using iterative generalised least squares. *Biometrika*, 73(1):43–56, 1986.

[4] H. Goldstein. *Multilevel Models in Educational and Social Research*. Charles Griffin & Company Ltd, 1987.

[5] H. Goldstein. *Multilevel Statistical Models*. Edward Arnold, 1995.

[6] J.J. Hox and J.K Roberts. *Handbook of Advanced Multilevel Analysis*. Taylor & Francis, 2011.

[7] I.G Kreft and J de Leeuw. *Introducing Multilevel Modeling*. SAGE Publications, Ltd, 2011.

[8] D.V. Lindley and A.F.M Smith. Bayes estimation for the linear model. *Journal of the Royal Statistical Society*, 43:1–41, 1972.

[9] A.S. Bryk & S.W. Raudenbush. *Hierarchical Linear Models*. Sage Publications Inc., 1992.

[10] Stephen W Raudenbush and Anthony S Bryk. *Hierarchical Linear Models: Applications and Data Analysis Methods*. Sage, 2nd edition, 2002.

[11] W.S. Robinson. Ecological correlations and the behavior of individuals. *Sociological Review*, 15:351–357, 1950.

[12] J.D. Singer. Using sas proc mixed to fit multilevel models, hierarchical models, and individual growth models. *Journal of Educational and Behavioral Statistics*, 23(4):323–355, 1998.

[13] S.H.C. Du Toit. *Analysis of multilevel models.Part 1:Theoretical Aspects*. Human Science Research Council, 1993.

Appendix

SAS Code:

```
proc glm data=timss15.ss9;
class IDSCHOOL;
model Mathach= GC_SLM GC_SCM S_LTest / solution;
run;
```

SAS Output:

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	8455553.48	2818517.83	498.06	<0.0001
Error	8036	45475666.17	5658.99		
Corrected Total	8039	53931219.65			

R-Square	Coeff Var	Root MSE	Mathach Mean
0.156784	19.84343	75.22628	379.0992

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	365.2535407	0.97242631	375.61	<0.0001
GCSLM	-3.0896085	0.59398087	-5.20	<0.0001
GCSCM	13.2303152	0.59449621	22.25	<0.0001
SLTest	43.3460529	1.51499778	28.61	<0.0001

Table 18: PROC GLM output for student-level variables

SAS Code:

```
proc glm data=timss15.ss9;
class IDSCHOOL;
model Mathach = MeanDisc SCHEcoDisadv C_LTest / solution;
run;
```

SAS output:

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	13576844.58	4525614.86	901.21	<0.0001
Error	8036	40354375.07	5021.70		
Corrected Total	8039	53931219.65			

R-Square	Coeff Var	Root MSE	Mathach Mean
0.251744	18.69272	70.86395	379.0992

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	416.6266400	1.13754191	366.25	<0.0001
MeanDisc	5.4269782	0.58486100	9.28	<0.0001
SCHEcoDisadv	43.8800238	1.28019144	34.28	<0.0001
CLTest	20.5585124	1.08572713	18.94	<0.0001

Table 21: PROC GLM output for school-level variables

Figure 1 SAS Code:

```
proc sgplot data=timss15.ss9;
reg x=BSBGSCM y=mathach / group = IDSCHOOL;
xaxis LABEL="Student confidence in mathematics";
yaxis LABEL="Mathematics achievement";
where IDSCHOOL < 20;
run;
```

Figure 2 SAS Code:

```
data timss15.ss8;
set timss15.ss7;
where IDSCHOOL in (2);
run;


proc sgplot data=timss15.ss8;
reg x=BSBGSCM y=mathach;
yaxis LABEL="Mathematics achievement";
run;
```

Figure 3 SAS Code:

```
proc sgplot data=timss15.ss8;
reg x=GC_SCM y=mathach;
xaxis LABEL="Student confidence centered around the mean";
yaxis LABEL="Mathematics achievement";
```

```
run;
```

3.1 The null model

```
proc mixed data=timss15.ss9;
class IDSCHOOL;
model Mathach = /solution;
random intercept / sub = IDSCHOOL;
run;
```

3.2 Models with school-level predictors only

```
proc mixed data=timss15.ss9;
class IDSCHOOL;
model Mathach = MeanDisc / solution ddfm = bw;
random intercept / sub =IDSCHOOL;
run;


proc mixed data=timss15.ss9 noclprint covtest noitprint;
class IDSCHOOL;
model Mathach = MeanDisc C_LTest SCHEcoDisadv / solution ddfm=bw notest;
random intercept / sub=IDSCHOOL type=un;
run;
```

3.3 Models with student-level predictors only

```
proc mixed data=timss15.ss9 noclprint covtest noitprint;
class IDSCHOOL;
model Mathach = GC_SLM / solution ddfm=bw notest;
random intercept GC_SLM / sub=IDSCHOOL type=un;
run;


proc mixed data=timss15.ss9 noclprint covtest noitprint;
class IDSCHOOL;
model Mathach = GC_SCM S_LTest  / solution ddfm=bw notest;
random intercept GC_SCM S_LTest / sub=IDSCHOOL type=un;
run;
```

3.4 Models with both student- and school-level predictors

```
proc mixed data=timss15.ss9 noclprint covtest noitprint;
class IDSCHOOL;
model Mathach = GC_SLM MeanDisc GC_SLM*MeanDisc / solution ddfm=bw notest;
random intercept GC_SLM / sub=IDSCHOOL type=un;
run;

proc mixed data=timss15.ss7 noclprint covtest noitprint;
class IDSCHOOL;
model Mathach = GC_SLM GCNTD C_LTest GC_SLM*GCNTD GC_SLM*C_LTest / solution ddfm=bw notest;
random intercept GC_SLM / sub=IDSCHOOL type=un;
run;
```

Uncertainties in the extraction of informal roads from remote sensing images

Renate Thiede 14288941

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr. I.N. Fabris-Rotelli, Co-supervisors: Prof. A. Stein, Prof. P. Debba

Department of Statistics, University of Pretoria

31 October 2017 (final version)

Abstract

The presence of undocumented shacks and informal roads in municipalities places great stress on the infrastructure and may lead to unsuitable development or policy decisions if the municipal authorities are unaware of their existence or location. This is a common problem in many South African municipalities. Information on shacks and informal roads is critical for town planners wishing to perform accessibility analyses in order to determine how many people have access to which facilities, such as clinics and schools. This in turn will enable the municipality to decide where to build new or upgrade existing roads and facilities, leading to sustainable city growth and maintenance.

In order to estimate the number and location of informal roads within an area to improve the municipality's understanding of its inhabitants, the number and location of informal roads should first be estimable. In this project, a first step towards this process, namely the detection of informal roads, is addressed. The state of the art region-based urban road extraction algorithm proposed in [14] is used. The method relies on the hierarchical representation of the study area in a Binary Partition Tree (BPT). Regions in the image are modelled using two geometrical features, namely region elongation and compactness, and two structural features, respectively utilising orientation histograms and path-based morphological profiles. The method is applied to extract roads from satellite images of two areas of Mabopane, Gauteng Province, South Africa. Good results are produced in the case of broad, straight unpaved roads ($12-15$m wide including road shoulders and sidewalks), but imperfect results are found for narrower informal roads in areas with more prevalent vegetation.

Declaration

I, *Renate Nicole Thiede,* declare that this research report, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Renate Nicole Thiede

_ _

Supervisors: Inger Fabris-Rotelli, Alfred Stein, Pravesh Debba

_ _

Date

Acknowledgements

I would like to thank my supervisors, Dr. Inger Fabris-Rotelli, Prof. Alfred Stein and Prof. Pravesh Debba for their guidance and support. I would also like to thank Dr. Mengmeng Li of the Faculty of GeoInformation Science and Earth Observation, University of Twente for his assistance and for the use of his code. The financial support of Statomet, Department of Statistics and financial support from the Center for AI Research, Meraka Institute, CSIR is acknowledged for this research. The VHR images used in the study were provided by the CSIR.

Contents

List of Figures

List of Tables

1 Introduction

There are many undocumented shack areas throughout South Africa. Unlike government-provided low income housing, shacks are informal dwellings constructed by residents without the knowledge of the local municipality, often outside of residential zones. These informal settlements, also known as townships, can grow to eventually accommodate a large number of residents, who may create some informal infrastructure inside the settlement, such as ad hoc unpaved roads. Without proper knowledge of the number and location of shacks and informal roads, problems such as overpopulation or stress on the infrastructure in an area may go unnoticed for too long, and no adequate solutions can be provided. On the other hand, having this information can empower municipal authorities to do far more than merely solving problems as they arise. Instead, it will enable them to proactively plan and assess the impacts of policies and developments.

One way in which this information can be incorporated into decision-making is via accessibility analyses. An accessibility analysis is a multiple-step process for determining which destinations can be reached from given origins, in what ways and with how much effort [3, 4, 15]. Accessibility is a comprehensive measure that can give an indication of the quality of the land use and transportation systems in an area. It is essential for developing sound policies for transport and land use that will support sustainable development [6, 4]. Since accessibility analyses provide information on how many people from given areas have access to which facilities, such as schools, clinics, government offices and places of potential employment, they also enable town planners and developers to decide where to build new or upgrade existing roads and facilities. Van Eck and De Jong [35] provide examples of service location planning, specifically planning the location of new shops, using accessibility information.

Three of the measures that should be minimised in order to increase accessibility are travel distance, which may be measured as straight line distance or distance along a transport network, travel time and travel cost [6, 13, 15, 35]. Data on informal unpaved roads is critical to accurately estimate all three. Travel distance and travel time between an origin and a destination may be less than expected if there is an informal unpaved road connecting them; however, it is also important to note that travel time on such a road will differ from that on a paved road. The quality of a paved road may decrease the travel time, but informal roads may be less liable to suffer from traffic congestion, which may compensate for the poor road surface quality in terms of travel time. Travel costs will also be influenced by the presence of informal unpaved roads. The cost of travel via such roads may be lower than the cost of travel between the same origin and destination via a formal road if travel time or distance is significantly reduced by such roads. It may also be reduced if the unpaved roads can be navigated by an informal mode of transport that is less expensive or otherwise more convenient than using the formal transport system, such as minibus taxis. Only transport by automobile will be considered; walking will not be accounted

Figure 1: Low-income formal housing next to informal dwellings.
Imagery ©2017 Google, Map data ©2017 AfriGIS (Pty) Ltd
Coordinates: 28°03'01.2"E 25°46'59.5"S

for since commuters do not necessarily walk only on roads.

Another important use for information on the number and location of informal roads is for the process of formalising informal infrastructure. The Tshwane municipality's Project Tirane aims to upgrade townships to formal settelements[1]. Knowing the extent of the informal road network in a township will enable the municipality to decide which informal roads to formalise and pave in order to optimise the local transport network.

Other services may also find the data on informal roads valuable, such as emergency services needing to navigate informal settlements more effectively.

Figure 1 shows images of an area of Atteridgeville in Tshwane where the contrast between government housing and informal settlements is clear. In the government-provided residential area, to the north of Maunde road, the houses are larger, have yards and are placed in a regular pattern. The roads are wide as well as generally straight and paved. The informal settlement to the south is characterised by small, irregularly-placed dwellings and narrow dirt roads that tend to be curved or winding.

Figure 2 shows images from the same area, displaying roads detected by Google Maps and roads not detected by Google Maps. In these images, all of the paved roads were detected, but not all of the dirt roads. In particular, the narrower, more curved roads have not been captured.

Given an accurate and efficient road extraction method, detecting informal unpaved roads from remote sensing images could be more cost and time effective than land surveying or manual digitising, allowing for the information to be updated more frequently. However, very little work has been done on the automatic extraction of informal roads from remote sensing images. The difficulties associated with informal unpaved roads include the irregular patterns followed by many informal road networks, as well as the fact that land cover is highly heterogeneous at the scale of objects in informally settled areas

[1] As reported on the official website of the Democratic Alliance, the South African political party of which the mayor of Tshwane is a member.
Link: https://www.da.org.za/2017/07/da-led-tshwane-selling-mayoral-mansion-bring-better-services/

Figure 2: Roads captured by Google Maps.
Imagery ©2017 Google, Map data ©2017 AfriGIS (Pty) Ltd
Coordinates (left-hand image): 28°02'49.5"E 25°47'04.7"S
Coordinates (right-hand image): 28°03'01.2"E 25°46'59.5"S

in Very High Resolution (VHR) images [21]. The object-based approach employed in [21] is one of the few works to address the problem of informal road extraction and highlights these and other difficulties. There is still a need to develop a method for efficiently extracting such informal roads from remote sensing imagery.

In [38] road features are classified and a wide variety of methods summarised and compared for extracting roads from remote sensing images, describing which methods make use of which road features. It is mentioned that the application of topological and functional features to real-world data is problematic; the implication is that geometric, photometric or texture features should be considered instead. Mena [18] provides a comprehensive summary of road extraction methods along with an extensive bibliography. Textures are based on the homogeneity of image regions [38]. Land cover may be highly heterogeneous in VHR images, and roads may be partially covered by dust and hence appear heterogeneous. Detecting roads by identifying them as homogeneous regions may therefore be problematic. Methods relying on the photometric characteristics of images, such as many classification-based methods, have the disadvantage of misclassification due to the spectral similarity between roads and other features. Informal roads are generally unpaved and will therefore be spectrally identical to non-road dirt areas and to building roofs made from local clay. Since roads have distinct elongated shapes, there is little risk of confusion with other features, especially buildings or yards, if the geometric properties are considered. Any long linear shape in the context of an informal settlement is likely to be a road, although it is important to distinguish roads from streams or railways. A great variety of techniques are based on geometric features. Graph

theory has the disadvantage of being complex, while dynamic programming requires prior knowledge of the data and can be greatly influenced by the values of preset parameters [38]. The Hough transform is often used to detect linear features [12], but is computationally intensive. Liu [16] proposes a more time-effective and less complex variant on the Hough transform method, but this has so far only been applied to straight lines. Active contour models are very sensitive to the preselected seed points. Mathematical morphology is compared to most of the abovementioned geometric methods, as well as some classification-based methods in [38] and has distinct advantages among these.

Mathematical morphology may be described as the mathematical study of shape [27]. Serra [26] and Soille [29] are two seminal works on the theory of mathematical morphology. Zhang et al. [39] applies mathematical morphology to automatically detect and extract road features. Amini et al. [1] uses mathematical morphology to extract roads in two stages, first by extracting straight lines and then by extracting a road skeleton. The line segment matching method is applied in [28] in order to form a road network by detecting straight-line segments. A significant disadvantage of morphological techniques is that a reference shape must be predetermined [32], e.g. straight lines in order to detect roads or rectangular shapes in order to detect buildings. In many cases, curved roads are approximated by short straight line segments, which leads to a loss in accuracy. Valero et al. [32] provides a possible solution with relation to road extraction by considering the morphological operators path openings and path closings. These operators are independent of the reference shape and fit both rectilinear and curvilinear structures. Path operators are clearly more flexible than morphological operations by straight lines; they are also more restricted than area operators in that they are constrained to detect only linear structures. Heijmans et al. [11] establishes the theory of path opening and closings. Morphological opening and closing are defined in terms of adjacency and paths, and algorithms for computing path openings for binary and grey tone images are provided. In [30], computationally efficient algorithms for applying path openings and closings are supplied.

The algorithm proposed in [14] will be applied to the problem. This algorithm is based on binary partition trees (BPT). Binary partition trees provide a hierarchical representation of an image at various scales which allows for the fast execution of complex image processing techniques [23]. BPT approaches are region-based and allow for the employment of various characteristics of a region. In [34], a BPT approach is shown to outperform the traditional pixel-based approach, where only spectral information can be used. Salembier and Garrido [23] provide a comprehensive introduction to the use of BPTs for image processing. Vilaplana et al. [36] applied BPTs to the problem of object detection. A review of the literature and a thorough discussion of BPTs may be found in [33], as well as an algorithm for constructing a BPT.

The algorithm in [14] considers the compactness and elongation of regions, as well as two features

Study Area 1
Coordinates: 28°4'7"E 25°30'53"S

Study Area 2
Coordinates: 28°1'26"E 25°31'25"S

Figure 3: The study areas.

based on path-based morphological profiles and orientation histograms. In this project, the algorithm is applied to Very High Resolution (VHR) satellite images of two areas of Mabopane, Tshwane Municipality, Gauteng Province, South Africa, shown in Figure 3. The project aims to ascertain which types of roads may be identified, and explores the uncertainties associated with the extraction of roads as spatial objects from spatial big data in the form of remote sensing images.

2 Background Theory

2.1 Mathematical Morphology

Mathematical morphology is a theory for the analysis of the shape of objects. It is based on rigorous mathematical theory, including lattice algebra and set theory [27]. The idea behind mathematical morphology is that the concept of the shape of an object exists neither exclusively in the physical world, nor is it an entirely subjective or abstract human perception. Rather, it is something between the two [26], an abstract concept that is evident in and severely constrained by physical reality. Mathematical morphology seeks to quantify the abstract interpretive aspect of the definition of shape.

2.1.1 Images and their Representations

This research report will consider grey tone and binary images in two dimensions. An image can be thought of as a two-dimensional grid where each element is called a pixel [29].

We will denote the set of all images by $\mathcal{F}$ and the set of all binary images by $\mathcal{F}_B$, where $\mathcal{F}_B \subset \mathcal{F}$.

Definition 1. Mathematically, an *image* f is a positive upper semi-continuous function [26] bounded by a value $t \in \mathbb{R}$ which maps some subset $\mathcal{D}_f$ of $\mathbb{Z}^2$ onto a subset of consecutive nonnegative integers starting at zero [29], namely

$$f : \mathcal{D}_f \subset Z^2 \to \{0, 1, ..., t\} \subset N_0,$$

where $\mathcal{D}_f$ is a rectangular frame known as the definition domain of f [29].

For a binary image f, $t = 1$, i.e. the value of each pixel of f can be either 0 or 1. The values are assigned in the following way:

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if the pixel } \mathbf{x} \text{ is part of the foreground} \\ 0 & \text{if the pixel } \mathbf{x} \text{ is part of the background} \end{cases}$$

The binary image f can be represented by the set of all image pixels.

Remark 2. Binary images may either consist of white foreground pixels on a black background, or black foreground pixels on a white background.

Example 3. Figure 4 shows a binary image f. In this case, the image pixels are black while the background is white. Letting the origin $(0,0)$ be the upper left corner, the set representation of f is

$$X = \{(0,0),(0,1),(0,2),(1,0),(1,1),(1,2),(2,0),(2,1),(2,2)\}.$$

Furthermore, $f(\mathbf{x}) = 1$ if $\mathbf{x} \in \{(0,1),(1,0),(1,1),(1,2),(2,0),(2,1)\}$. □

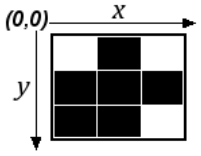


Figure 4: A binary image f

Let $\mathcal{G}$ be the set of all grey tone images. For a grey tone image f, t can take on any finite value. When $t = 255$, the image is called a greyscale image. Grey tone images can be represented as objects in $\mathbb{Z}^3$ [10], where the (x, y, z) coordinates of each pixel are given by the x and y axes and the grey value $f(x, y)$ is given by the z axis. Soille [29] represents grey tone images as digital elevation models where the elevation corresponds to the grey value. This representation allows us to envision grey tone images as a stack of t binary images, one for each grey value, with corresponding set representations $X_{(i)} = \{\mathbf{x} | f(\mathbf{x}) = i\}$, $i = 0, 1, ..., t$. Grey tone images can be decomposed into these binary images using the threshold operator T [29]:

$$[T_{[i,j]}(f)](\mathbf{x}) = \begin{cases} 1 & i \leq f(\mathbf{x}) \leq j \\ 0 & \text{elsewhere} \end{cases}$$

$CS_i(f)$ is called the cross section of the grey tone image f and is the set of all pixels $\mathbf{x}$ of f such that $f(\mathbf{x}) \geq i$, i.e.

$$[CS_i(f)](\mathbf{x}) = \{\mathbf{x} | i \leq f(\mathbf{x}) \leq t\} = [T_{[i,t]}(f)](\mathbf{x}).$$

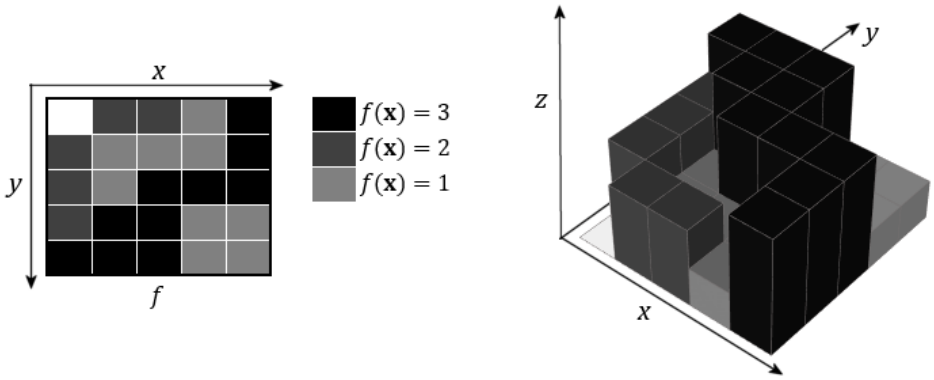


Figure 5: A grey tone image f with its DEM representation

The grey tone image may be obtained by summing the cross sections CS_i where $i = 0, 1, 2, ...$ [29]:

$$f = \sum_{i=0}^{t} CS_i(f).$$

The value of $f(\mathbf{x})$ is the maximum value of i such that $[CS_i(f)](\mathbf{x}) \geq 0$ [29], i.e.

$$f(\mathbf{x}) = \max\{i | [CS_i(f)](\mathbf{x}) \geq 0\}.$$

Example 4. Figure 5 shows an image f with its representation as a digital elevation model (DEM). Suppose the pixels of f take on values in the following way:

$$f(\mathbf{x}) = \begin{cases} 3 \text{ if the pixel } \mathbf{x} \text{ is black} \\ 2 \text{ if the pixel } \mathbf{x} \text{ is medium grey} \\ 1 \text{ if the pixel } \mathbf{x} \text{ is light grey} \\ 0 \text{ if the pixel } \mathbf{x} \text{ is white i.e. part of the background} \end{cases}$$

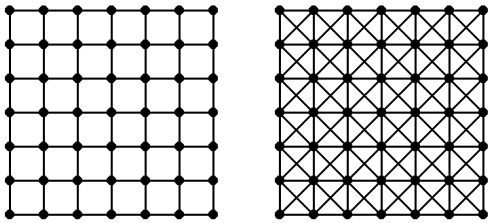
The cross sections are as follows:

$CS_0 = \{\mathbf{x} | f(\mathbf{x}) = 0\} = \{(0,0)\}$

$CS_1 = \{\mathbf{x} | 1 \leq f(\mathbf{x}) \leq 3\} = \mathcal{D}_f \setminus \{(0,0)\}$

$CS_2 = \{\mathbf{x} | 2 \leq f(\mathbf{x}) \leq 3\} = \{(0,1),(0,2),(0,4),(1,0),(1,4),(2,0),(2,2),(2,3),(2,4),(3,0),(3,1)(3,2),(4,0),(4,1),(4,2)\}$ So, for example, $[CS_1(f)](0,0) = 1 > 0$, $[CS_2(f)](0,0) = 1 > 0$ and $[CS_3(f)](0,0) = 1 > 0$. Therefore $f(0,0) = \max\{i | [CS_i(f)](0,0) > 0\} = \max\{1,2,3\} = 3$. Similarly, $f(1,4) = \max\{i | [CS_i(f)](1,4) > 0\} = \max\{1,2\} = 2$ and $f(3,0) = \max\{i | [CS_i(f)](3,0) > 0\} = \max\{1\} = 1$, etc. $\square$

$CS_3 = \{\mathbf{x} | 3 \leq f(\mathbf{x}) \leq 3\} = \{(0,0),(1,0),(1,1),(2,0),(2,1),(2,2),(3,2),(4,2),(4,3),(4,4)\}$



a) b)

Figure 6: Non-oriented graphs. 4-connectivitity is shown in a) while the graph in b) is 8-connected.

2.1.2 Graphs, Connectivity and Adjacency

The concepts and definitions in this section are compiled from [11], [30] and [32]. The theory in this section is relevant to the sections 2.1.8, 2.1.9 and 2.1.11.

An image may be conceptualised as a set of pixels or points. However, some thought must be given to the relationships between pixels, such as adjacency. The definitions in this section provide a way of accounting for such relationships.

Definition 5. A *non-oriented graph* G is a set of points connected by vertices [29].

The points in a graph may be in any pattern. Since an image is typically represented as a matrix of pixels, the pattern relevant for image analysis is a square grid.

Definition 6. *Connectivity* determines which points or pixels can be said to be adjacent to which other points or pixels.

There are different types of connectivity. The two most common examples in the case of square grids are 8-connectivity and 4-connectivity. In the case of 8-connectivity, all points surrounding a point **x** are connected to **x**, while in the case of 4-connectivity, only those points directly above, below, to the left or the right of **x** are connected [29]. In the context of spatial raster data, 4-connectivity is sometimes referred to as rook's connectivity, referring to the moves of a rook in a chess game, while 8-connectivity is sometimes called queen's or king's connectivity [9].

Example 7. Figure 6 shows two non-oriented graphs. The points in these graphs are connected by lines or vertices demonstrating 4-connectivity and 8-connectivity. □

Another way of thinking of connectivity in non-oriented graphs is to imagine walking from point to point, using the vertices as paths. Starting from a point x, one can only walk to those points sharing a vertex with x. However, graphs are not always non-oriented. It may be useful to restrict the direction of movement between points. Consider two connected points x and y and suppose that one is allowed to walk from x to y, but not from y to x. This is the concept behind adjacency relations.

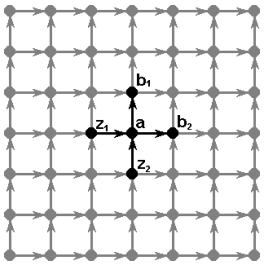


Figure 7: A directional graph showing the predecessors and successors of the point a.

Definition 8. The adjacency relation $x \mapsto y$ signifies that the point y may be directly reached from the point x, but not vice versa. The point y is called a *successor* of x while x is called a *predecessor* of y. This relation is not in general symmetric or reflexive[11].

Definition 9. A *directional graph* is defined by a set of points and vertices as well as an adjacency relation "$\mapsto$".

Example 10. A directional graph is shown in Figure 7. The point a may be reached either from the point z_1 or z_2, so they are the predecessors of a. The points b_1 and b_2 are the successors of a since they can be reached from a. $\square$

2.1.3 Properties of Operators

Let f and g be images with the same domain $\mathcal{D}$. Let ψ and θ be operators that map elements of $\mathcal{D}$ onto $\mathbb{Z}$, i.e. $\psi : \mathcal{D} \to \mathbb{Z}$ and $\theta : \mathcal{D} \to \mathbb{Z}$. Then some of the possible properties of these operators are:

1. Increasingness: ψ is *increasing* if $f \leq g \;\Rightarrow\; \psi(f) \leq \psi(g)$
2. Exensivity: ψ is *extensive* if $\psi(f) \geq f$
3. Anti-extensivity: ψ is *anti-extensive* if $\psi(f) \leq f$
4. Idempotence: ψ is *idempotent* if $\psi\psi(f) = \psi(f) \quad \forall f \in \mathcal{F}, \quad i.e. \quad \psi\psi = \psi$
5. Duality: ψ and θ are *dual operators* if $\psi(f^c) = [\theta(f)]^c$ where c indicates the complement.

2.1.4 Dilation and Erosion

Definition 11. The *Minkowski addition* of a set A by a set $\{b\}$ is denoted by A_b and is defined as follows:

$$A_b = A \oplus \{b\} = \{a + b | a \in A\}$$

Definition 12. In [8], the *Minkowski addition* of two sets A and B is defined:

$$A \oplus B = \{a + b | a \in A,\ b \in B\}$$

The Minkowski addition of the sets is therefore obtained by taking the union of the Minkowski addition of A with every element of B:

$$A \oplus B = \bigcup_{b \in B} A_b$$

The*Minkowski decomposition or subtraction* of two sets A and B is defined as the set C such that $A = B \oplus C$, which can be written as follows [29]:

$$A \ominus B = \{a | \forall b \in B,\ a + b \in A\} = \bigcap_{b \in B} A_b$$

Minkowski addition and subtraction are closely related to the operators erosion and dilation [17, 26].

Definition 13. The *dilation* of a set X by a set B is given by the Minkowski addition of X and $\check{B}$ where $\check{B} = \bigcup_{b \in B} \{-b\}$ is called the transpose of set B, i.e.:

$$\delta_B(X) = X \oplus \check{B} = \{x | B_x \cap X \neq \varnothing\}$$

Definition 14. The *erosion* of a set X by a set B is given by the Minkowski subtraction of $\check{B}$ from X , i.e.

$$\varepsilon_B(X) = X \ominus \check{B} = \{x | B_x \subset X\}$$

Definition 15. The set B is called the *structuring element (SE).*

Remark 16. Note that, in general, erosion and dilation do not undo one another's effects, i.e. $\delta\varepsilon(f) \neq \varepsilon\delta(f)$.

Intuitively speaking, a binary image is dilated by placing the origin of the SE on some pixel within the image and seeing if the SE hits any foreground pixels. If it does, the pixel at the origin of the SE is included in the dilated image. This is repeated until the origin of the SE has been placed on every pixel in the image. In effect, for each position of the SE, all the pixels in the SE will be foreground pixels in the dilated image. Similarly, a binary image is eroded by placing the SE on some pixel within the image and seeing if it is completely contained within an area of foreground pixels. If it does, those foreground pixels will be included in the eroded image; otherwise, they will become background pixels. This process is also repeated until the origin of the SE has been placed on every image pixel.

Example 17. Figure 8 shows the erosion of a binary image f by a symmetrical (in the origin) vertical linear SE. The eroded image contains mostly vertical linear structures. The SE in Figure 9 is horizontal

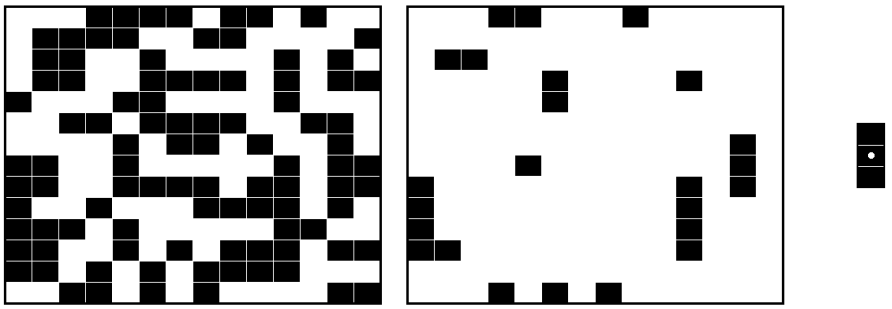


Figure 8: Erosion of a binary image f by a vertical linear SE

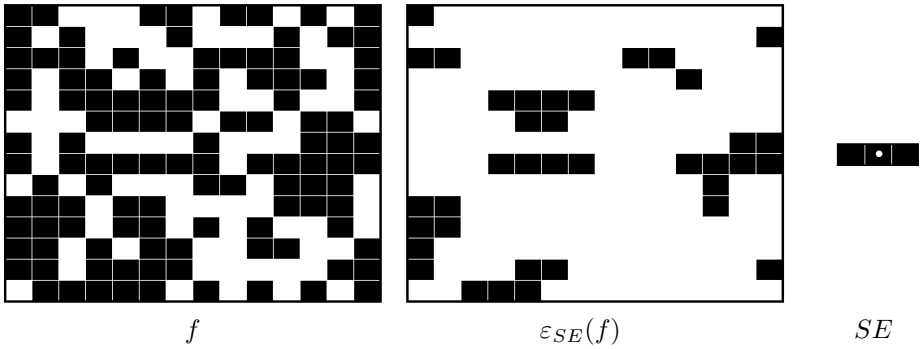


Figure 9: Erosion of a binary image f by a horizontal linear SE

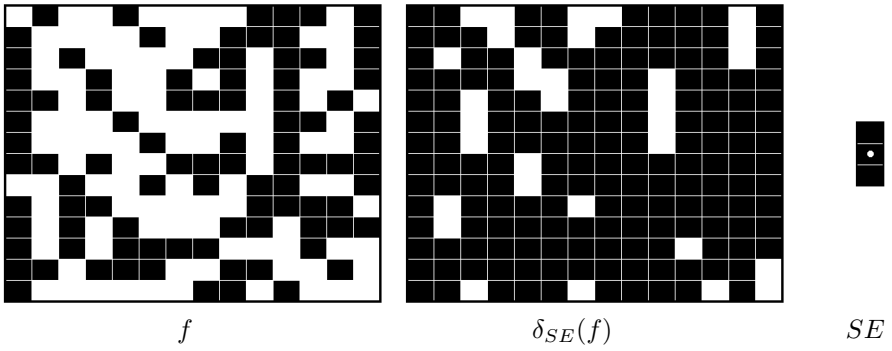


Figure 10: Dilation of a binary image f by a vertical linear SE

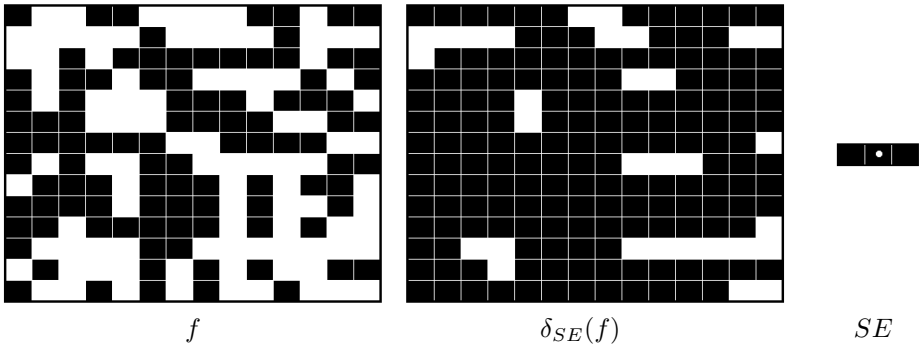


Figure 11: Dilation of a binary image f by a horizontal linear SE

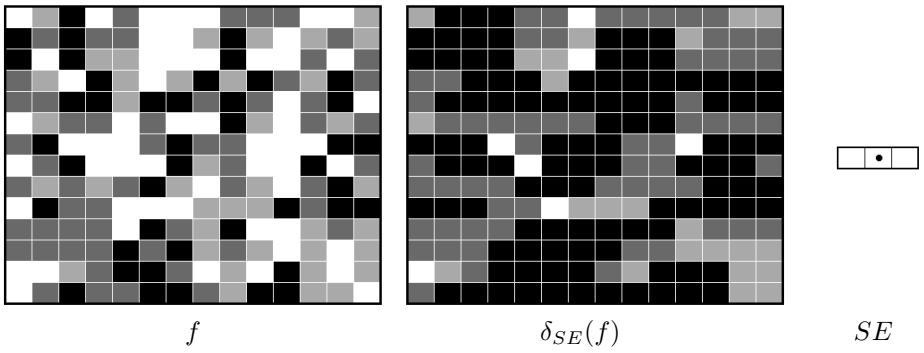


Figure 12: Dilation of a grey tone image f by a horizontal linear SE

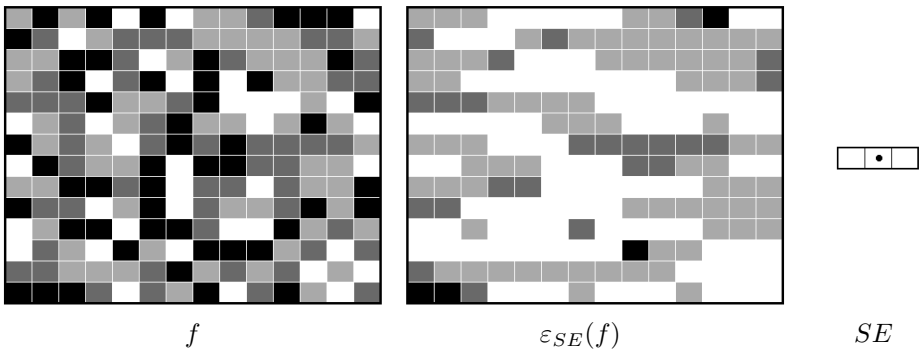


Figure 13: Erosion of a grey tone image f by a horizontal linear SE

linear, and the erosion contains primarily horizontal linear structures. In general, the features preserved by the erosion will be of the same kind of structure as the SE. The dilation by a vertical linear SE shown in Figure 10 preserves mainly vertical linear background shapes, while Figure 11 shows dilation by a horizontal linear SE, which contains principally horizontal linear background areas. In general, the shapes of the background areas preserved by the dilation are dictated by the structure of the SE. □

Dilation and erosion can be generalised to grey tone images by making use of the operators $\vee$ (supremum/maximum) and $\wedge$ (infimum/minimum). Let f be a grey tone image and B be a structuring element, and let $f_{\mathbf{b}}$ be the translation of f by the vector $\mathbf{b}$, $\mathbf{b} \in B$. Then the following hold [29]:

1. $\delta_B(f) = \bigvee_{\mathbf{b}\in B} (f_{\text{-}\mathbf{b}})$
2. $[\delta_B(f)](\mathbf{x}) = \max_{\mathbf{b}\in B} f(\mathbf{x}+\mathbf{b})$
3. $\varepsilon_B(f) = \bigwedge_{\mathbf{b}\in B} (f_{\text{-}\mathbf{b}})[\varepsilon_B(f)]$
4. $[\varepsilon_B](f)(\mathbf{x}) = \min_{\mathbf{b}\in B} f(\mathbf{x}+\mathbf{b})$

Example 18. Figure 12 shows the dilation of grey tone image f by a horizontal linear SE, while 13 shows the erosion of a different grey tone image f by the same SE. Just as in the binary case, dilation preserves and enlarges dark-coloured horizontal linear structures, while erosion preserves and enlarges light-coloured horizontal linear structures. □

Remark 19. The previous two examples clearly show that the image structures preserved by erosion and

dilation are determined by the form of the SE. The advantage of this is that the choice of the SE allows the image analyst to decide the kind of structure to be extracted. For example, linear SEs can be used to extract straight line features, such as roads in remote sensing images, while square SEs can be used to extract square features such as buildings. Serra [26] puts it as follows: "[T]he notion of a geometrical structure ... does not exist in the phenomenon itself, nor in the observer, but somewhere in between the two. Mathematical morphology quantifies this intuition by introducing the concept of structuring elements." The disadvantage is that SEs can provide too rigid constraints. Roads, for example, may be rectilinear or curvilinear, and so cannot be described by one SE structure; specifiying a straight line SE may lead to curved roads being omitted, while an SE trying to imitate the form of a curved road would not lead to the extraction of the straight roads (as well as omitting roads that curve in a different direction, or are more curved or straighter than the SE). This notion is discussed further, and a possible solution with regard to road extraction is provided in Section 2.1.11.

2.1.5 Opening and Closing

In this section, two particular compositions of erosions and dilations are discussed.

Definition 20. The opening of an image f by a set B is defined as an erosion by B followed by a dilation by $\check{B}$, i.e. $\gamma_B(f) = [\delta_{\check{B}} \circ \varepsilon_B](f)$

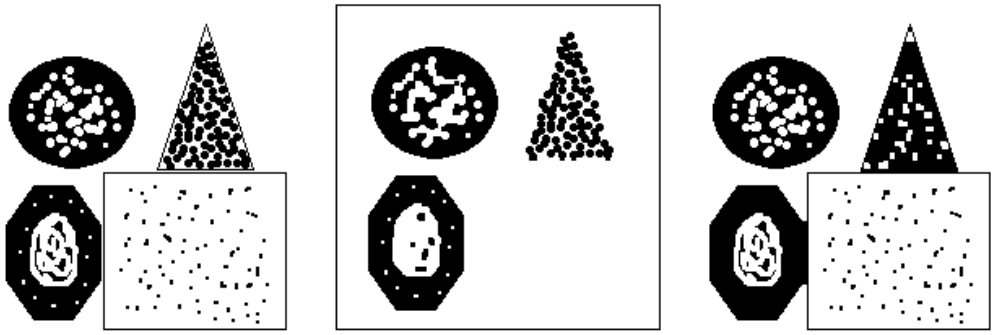
The opening removes all structures in the image that do not contain the SE B [29]. In this way it removes small protrusions and opens up holes and cavities in structures in the image. It entirely removes structures that are too small to contain the SE.

Definition 21. The closing of an image f by a set B is defined as a dilation by B followed by an erosion by $\check{B}$, i.e. $\phi_B(f) = [\varepsilon_{\check{B}} \circ \delta_B](f)$

The closing fills up cavities inside structures and bridges small gaps between structures in the image, wherever the cavities and gaps are too small to fit the SE B [29].

Example 22. The opening and closing of a binary image by a 3×3 square SE are shown in Figure 14. The black pixels were considered part of objects while the background was white. Note that the opening has entirely removed all black objects smaller than the SE, including the boundaries of the triangle and the rectangle. It has also joined the white spots inside the ellipse. On the other hand, the closing has joined the black spots inside the triangle, and has removed the white background areas smaller than the SE, including the white spots in the octagon and the thin background slivers between some of the large objects.

Figure 15 shows the opening and closing of a greyscale image, also by a 3×3 square SE. In this case, the dark areas were preserved by the opening while the bright areas were preserved by the closing.



a) b) c)

Figure 14: Opening and closing of binary image: a) Original image b) Binary opening c) Binary closing

a) b) c)

Figure 15: Opening and closing of greyscale image: a) Original image b) Grey tone opening c) Grey tone closing

Most strikingly, the closing removed building shadows. The roads are clear in both the opening and the closing, but appear to be more sharply delineated by the closing. □

Properties of opening and closing

1. Opening and closing are dual operators.
2. The opening is (a) increasing, (b) idempotent, and (c) anti-extensive.
3. The closing is (a) increasing, (b) idempotent, and (c) extensive.

2.1.6 Geodesic Dilation and Geodesic Erosion

This section refers to [29]. Geodesic dilation and erosion involve the use of a control image or mask g to limit the expansion and shrinking of the objects in an image f caused by dilation and erosion respectively. The images g and f must have the same definition domain and g must be greater than or equal to f, i.e. the requirements are:

1. $\mathcal{D}_f = \mathcal{D}_g = \mathcal{D}$ and

2. $g \geq f$

where $g \geq f \quad \Leftrightarrow \quad g(\mathbf{x}) \geq f(\mathbf{x}) \ \ \forall \mathbf{x} \in \mathcal{D}$.

Definition 23. The geodesic dilation of size 1 of an image f using a mask image g is obtained by taking, at each pixel, the minimum of g and the dilation of f, i.e.:

$$\delta_g^{(1)}(f) = \delta(f) \wedge g = \min\{\delta(f), g\}$$

Definition 24. The geodesic erosion of size 1 of an image f using a mask image g is obtained by taking, at each pixel, the maximum of g and the erosion of f, i.e.:

$$\varepsilon_g^{(1)}(f) = \delta(f) \vee g = \max\{\varepsilon(f), g\}$$

Definition 25. The geodesic dilation or erosion of size n of an image f using a mask image g is obtained by applying the geodesic dilation or erosion operator to f n times:

$\delta_g^{(n)}(f) = \delta_g^{(1)}[\delta_g^{(n-1)}(f)]$

$\varepsilon_g^{(n)}(f) = \varepsilon_g^{(1)}[\varepsilon_g^{(n-1)}(f)]$

Remark 26. Geodesic dilation and geodesic erosion are dual operators.

Example 27. Figure 16 illustrates the geodesic erosion and dilation of a binary image. The original image is shown in a) while the masks used in the geodesic operations are shown in b) and c). The erosion and dilation of the original image are shown in d) and e) respectively. The geodesic erosion by the white mask is shown in f) while the geodesic dilation by the black mask is shown in g). □

Remark 28. It is worth noting that geodesic dilation and geodesic erosion both eventually converge to a stable result. This is discussed further in Section 2.1.7.

2.1.7 Morphological Reconstruction

Repeatedly applying geodesic erosion or geodesic dilation to an image will eventually lead to a stable result. This can be expressed as follows:

$$\lim_{n\to\infty} \delta_g^{(n)}(f) = h_\delta \text{ for some image } \ h_\delta \in \mathcal{F}$$

$$\lim_{n\to\infty} \varepsilon_g^{(n)}(f) = h_\varepsilon \text{ for some image } \ h_\varepsilon \in \mathcal{F}$$

where $\delta_g^{(n)}(f) = [\delta_g^{(n-1)}(\delta_g^{(1)})](f)$ and $\varepsilon_g^{(n)}(f) = [\varepsilon_g^{(n-1)}(\varepsilon_g^{(1)})](f)$.

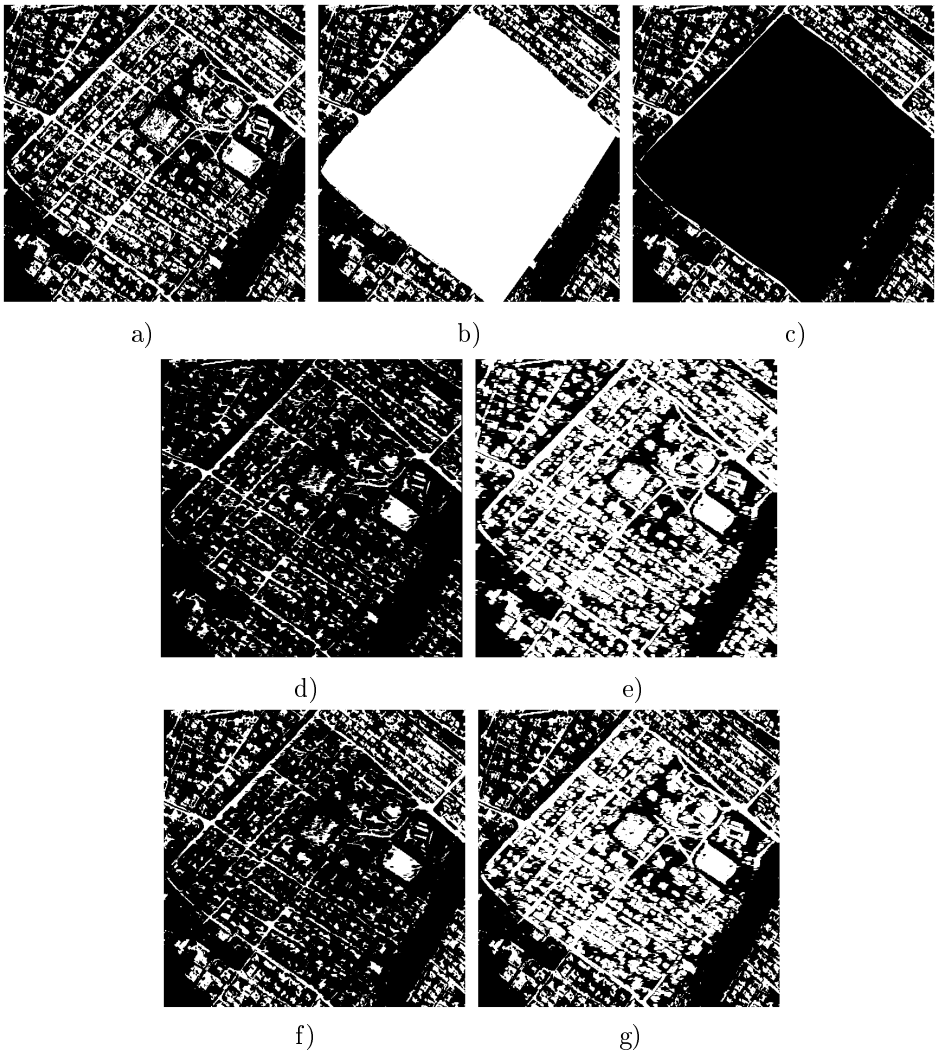


Figure 16: Geodesic dilation and erosion of an image: a) Original image b) White mask c) Black mask d) Erosion by a linear SE e) Dilation by the same SE f) Geodesic erosion using the black mask g) Geodesic dilation using the white mask

Definition 29. The image obtained by performing geodesic erosion on an image f by a mask image g until convergence is called the *reconstruction by erosion* of a mask image g from an image f and is denoted by $R_g^{(\varepsilon)}(f)$ [29], i.e.:

$$R_g^{(\varepsilon)}(f) = \varepsilon_g^{(n)}(f),$$

where $n \geq N$ and N is such that $\varepsilon_g^{(n)}(f) = \varepsilon_g^{(n+1)}(f)$ i.e. convergence has been reached.

Definition 30. The image obtained by performing geodesic dilation on an image f by a mask image g until convergence is called the *reconstruction by dilation* of a mask image g from an image f and is denoted by $R_g^{(\delta)}(f)$ [29], i.e.:

$$R_g^{(\delta)}(f) = \delta_g^{(n)}(f),$$

where $n \geq N$ and N is such that $\delta_g^{(n)}(f) = \delta_g^{(n+1)}(f)$.

Note that $R_g^{(\delta)} = h_\delta$ and $R_g^{(\varepsilon)} = h_\varepsilon$ with h_δ and h_ε as defined previously.

Definition 31. The *opening by reconstruction* of an image f by a structuring element B is denoted by Γ_f and is obtained by first performing erosion on the image, and then performing geodesic dilation on the eroded image $\varepsilon(f)$ using the original image f as a mask, i.e.:

$$\Gamma_f = R_f^{(\delta)}(\varepsilon_B(f)),$$

Opening by reconstruction first entirely removes all structures from the image that are too small to contain the SE through the erosion. The erosion also shrinks the other structures; these are completely restored to their respective original sizes and shapes by the geodesic dilation [37].

Definition 32. The *closing by reconstruction* of an image f by a structuring element B is denoted by Φ_f and is obtained by first performing dilation on the image, and then performing geodesic erosion on the dilated image $\delta(f)$ using the original image f as a mask, i.e.:

$$\Gamma_f = R_f^{(\varepsilon)}(\delta_B(f)).$$

Example 33. Figure 17 shows reconstruction by dilation and reconstruction by erosion of the image in Figure 16. The geodesic dilation and erosion, respectively, after first 5 and then 10 iterations are shown, along with the final converged results. Note that the final result of the reconstruction by erosion is in this case the same as the white mask given in Figure 17 while the result of the reconstruction by dilation is the same as the black mask. □

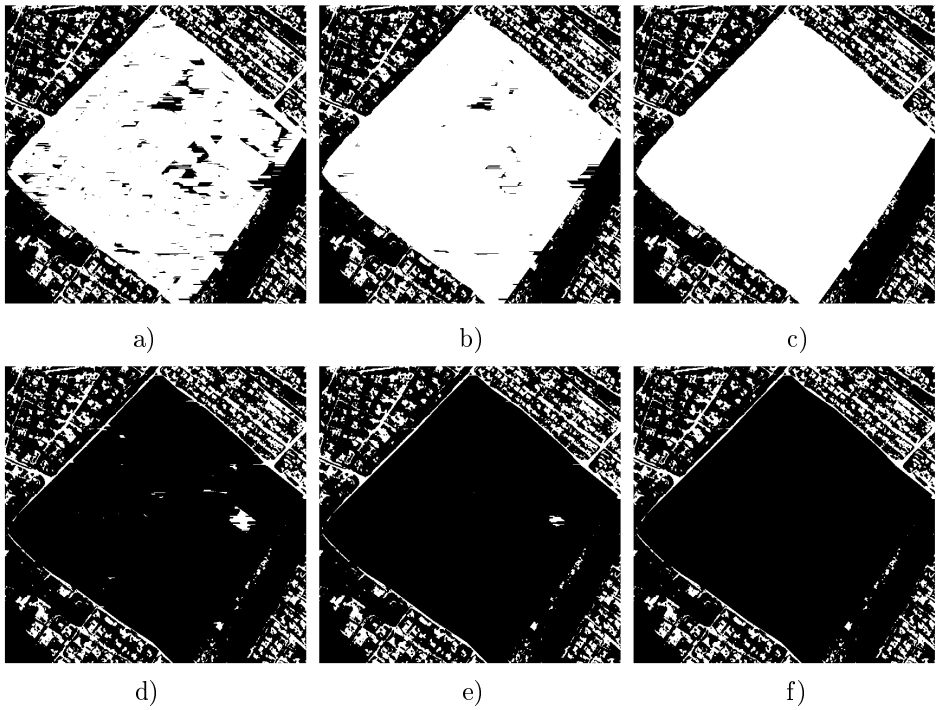


Figure 17: Morphological reconstruction of the binary image shown in Figure 16. Reconstruction by dilation is shown in a)-c): a) Geodesic dilation after 5 iterations b) Geodesic dilation after 10 iterations c) Reconstruction by dilation. Reconstruction by erosion is shown in d)-f): d) Geodesic erosion after 5 iterations e) Geodesic erosion after10 iterations f) Reconstruction by dilation.

2.1.8 Elementary SEs

This section defines the concept of elementary SEs, a concept crucial to understanding Section 2.1.9. Recall from Section 2.1.2 that connectivity determines which pixels are immediate neighbours to which other pixels and that two of the most common kinds of connectivity are 8-connectivity and 4-connectivity.

Recall that in the case of 8-connectivity, all pixels surrounding a pixel $\mathbf{x}$ are the 1st-order neighbours of $\mathbf{x}$. The pixels surrounding $\mathbf{x}$ with its 1st-order neighbours are the 2nd-order neighbours of $\mathbf{x}$, etc. In the case of 4-connectivity, only those pixels adjacent to $\mathbf{x}$, i.e. sharing a side with $\mathbf{x}$, are considered 1st-order neighbours. Those pixels sharing sides with the 1st-order neighbours of $\mathbf{x}$ are the 2nd-order neighbours of $\mathbf{x}$, etc.

Definition 34. The *elementary SE* $B^\star$ is the set containing a pixel and its immediate neighbours, where the neighbours are defined by the underlying neighbourhood graph [29].

Example 35. Figure 18a) shows the first- to fourth-order neighbours of a pixel $\mathbf{x}$ along with the elementary SE in the case of an underlying 8-connectivity. Figure 18b) shows the same for an underlying 4-connectivity. □

Remark 36. For the remainder of the theory section, we will assume that the underlying neighbourhood graph is 8-connected unless otherwise stated. This simply means that the immediate neighbours of a

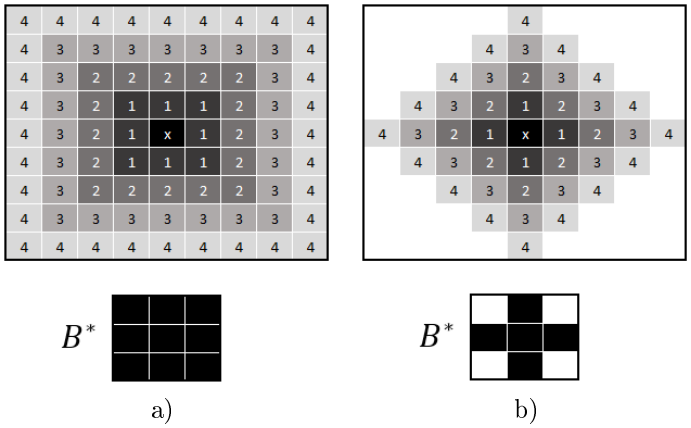


Figure 18: Neighbours and elementary SEs with respect to different underlying connectivity. The 8-connected case is shown in a) while b) shows the 4-connected case.

certain pixel **x** are all those pixels surrounding **x**.

2.1.9 Morphological Gradients and Top Hats

It is assumed that the edges of objects or structures in an image will be found in areas where the grey values vary greatly [22, 29]. A morphological gradient is an operator that enhances the contrast in pixel intensity in a specified neighbourhood [22]. Since they enhance contrast, morphological gradients can be used to detect edges.

The concept and function of morphological gradients will be made clearer by considering examples of specific gradients. In the definitions of gradients and filters in this section, erosion, dilation, opening and closing are done with respect to the elementary SE, defined in Section 2.1.8.

The following definitions can be found in [29].

Definition 37. The *thin* or *Beucher gradient* of an image f is defined as the difference between the dilation and erosion of the image:

$$\rho_{B^\star}(f) = \delta_{B^\star}(f) - \varepsilon_{B^\star}(f).$$

Definition 38. The *thick gradient* of an image f is defined as the difference between the dilation and erosion of the image where the size of the SE is greater than 1:

$$\rho^{(n)}_{B^\star}(f) = \delta^{(n)}_{B^\star}(f) - \varepsilon^{(n)}_{B^\star}(f).$$

Example 39. Figure 19 shows a binary image along with its gradients. The dimensions underneath each gradient shows the dimensions of the SE. The underlying connectivity is 8-connectivity, so the elementary SE is a square. The gradient by the elementary SE (i.e. the 3×3 square) is the Beucher or thin gradient, while the others are thick gradients. □

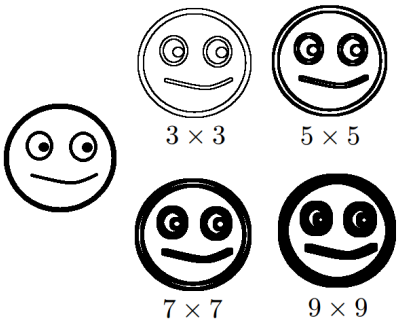


Figure 19: Thin and thick gradients of a binary image

Definition 40. The *half-gradient by dilation* or *outer gradient* of an image f is defined as the difference between the dilation of the image and the original image:

$$\rho^{+}_{B^\star}(f) = \delta_{B^\star}(f) - f.$$

Definition 41. The *half-gradient by erosion* or *inner gradient* of an image f is defined as the difference between the original image and the erosion of the image:

$$\rho^{-}_{B^\star}(f) = f - \varepsilon_{B^\star}(f).$$

Example 42. Figure 20 shows a grey tone satellite image with its thin gradient (by a square SE), a thick gradient (a gradient by a 5×5 square SE) and its inner and outer gradients. Note that the outer gradient gives a clearer delineation of linear structures than the inner gradient. □

Just as erosion and dilation can be used to define certain gradients, opening and closing can be used to define top hat filters. The idea behind top hat filters is to extract certain structures from an image by firstly removing the desired image structures using a morphological operation by an SE that does not fit the structures, and by secondly considering the difference between the original image and the result of the operation [29].

Let B be an SE.

Definition 43. The *white top hat* of an image f is the difference between the original image and the opening of the image:

$$WTH(f) = f - \gamma_B(f).$$

Since the opening removes light-coloured groups of pixels that do not fit the SE, the white top hat extracts these light-coloured image structures.

Definition 44. The *black top hat* of an image f is the difference between the closing of the image and

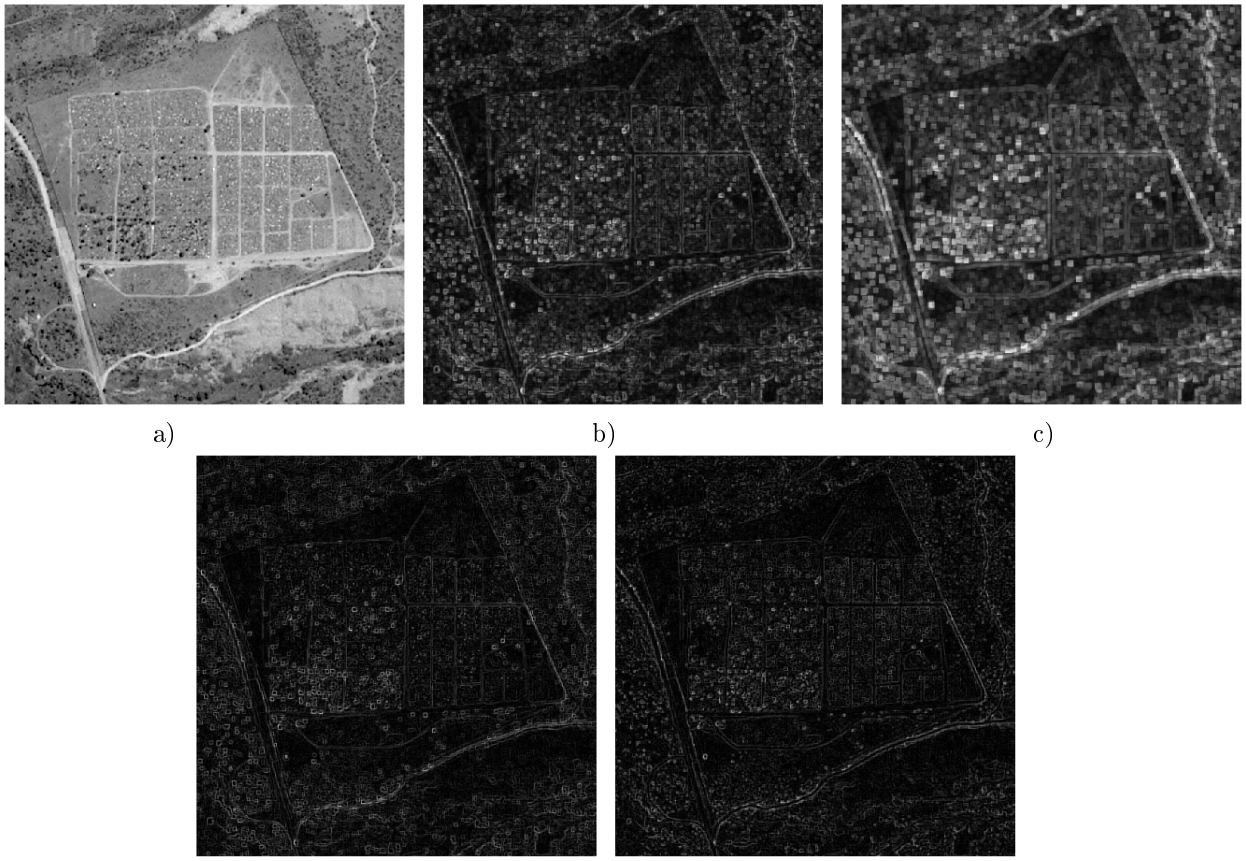


Figure 20: Greyscale satellite image with its gradients a) Original grey tone satellite image b) Thin gradient c) Thick gradient d) Inner gradient e) Outer gradient

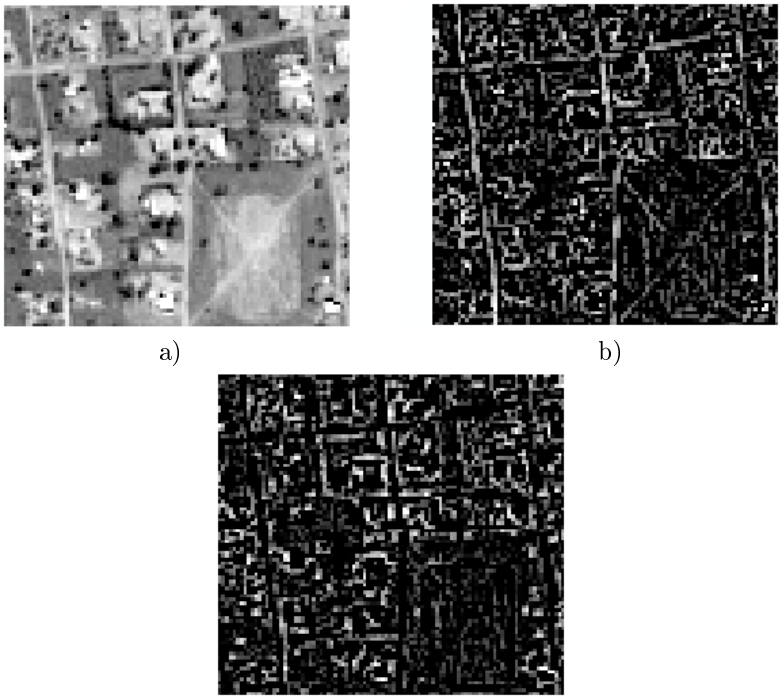


Figure 21: Greyscale satellite image shown in a) with b) White top hat c) Black top hat

the original image:

$$BTH(f) = \phi_B(f) - f.$$

The black top hat extracts the dark-coloured image structures that are removed by the closing.

Example 45. Figure 21 shows a grey tone satellite image along with the results of the white top hat and the black top hat (a cross-shaped SE was used). In this case, the dark pixels were considered part of the background and the lighter pixels were considered to be feature pixels. Note that the white top hat has extracted the lighter grey values (feature pixels) removed by the opening, namely the roads and the lines on the field, while the black top hat has extracted the darker areas (background areas) that were removed by the closing, namely the sides of the roads and lines. In both cases, the locations of the roads and lines are fairly clear, although in this example the results of the white top hat may be more appropriate for locating roads. □

Example 46. Top hats are now applied to the problem of detecting roads in an informal settlement. Figure 22 shows an image with both formal and informal settlements, with formal development occurring in the bottom right part and informal settlements in the top left area. Alongside this image are the white top hats of the image using the corresponding SEs. It is clear from these images that linear SEs are too constrained to detect curved roads, or even straight roads at different orientations. This motivates the use of path operators, which will be discussed in Section 2.1.11. □

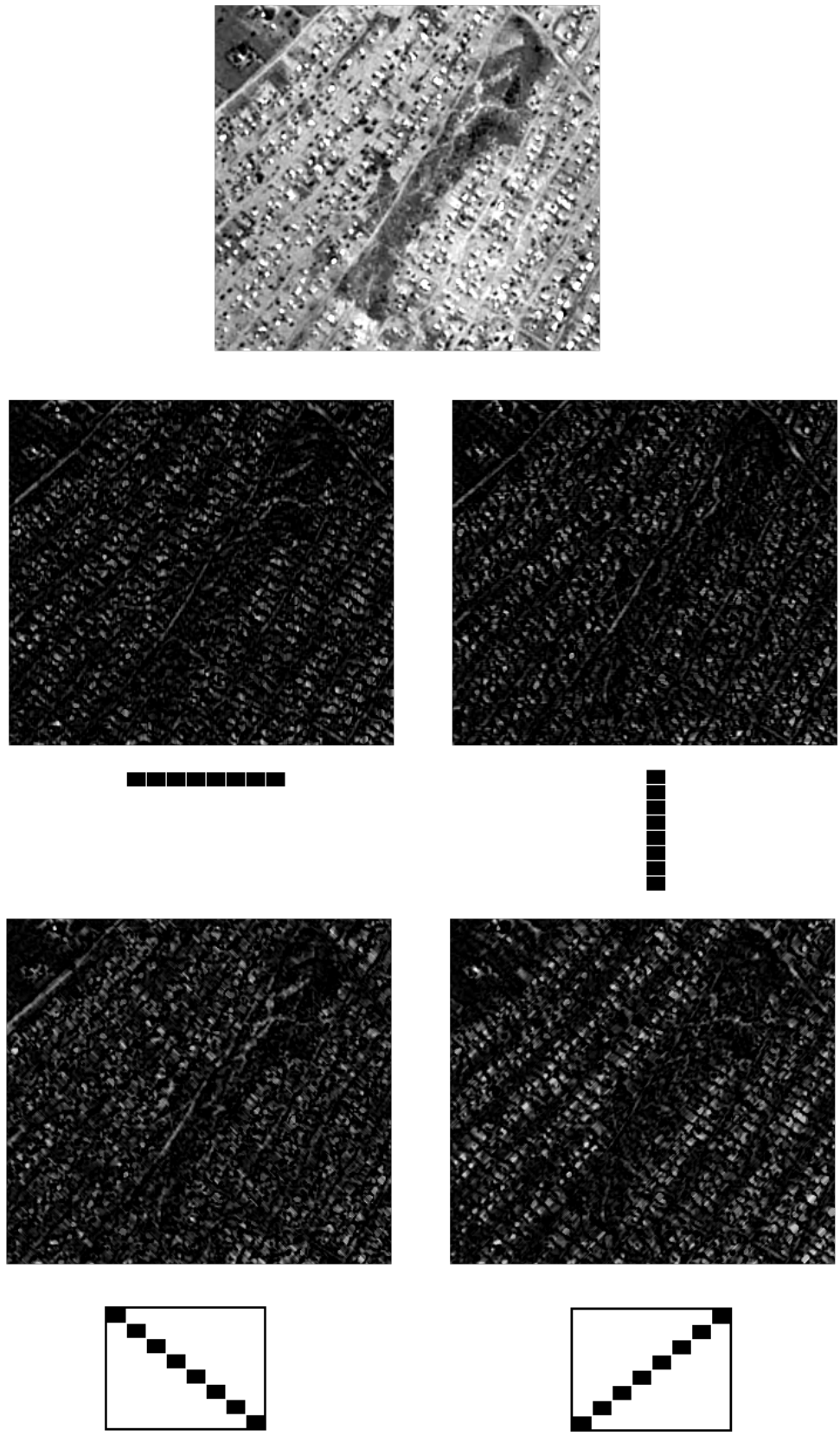


Figure 22: White top hats applied to an image of urban settlements.

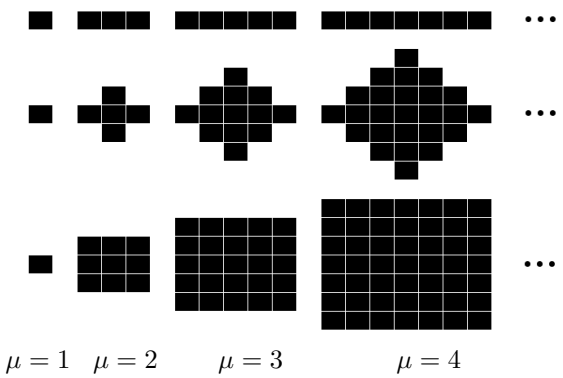


Figure 23: Structuring elements of size μ

2.1.10 Granulometries

It is often desirable to measure the distributions of the sizes of objects in an image. Size distributions, also called granulometries, can be used to obtain and represent how the sizes of objects in an image are distributed. Before granulometries can be further discussed, the absorption property must first be defined.

Definition 47. An operator ψ is *absorbing* if the operator has the following property:

$$\psi_\mu \psi_\lambda = \psi_\lambda \psi_\mu = \psi_{\max(\mu,\lambda)},$$

where ψ_μ refers to the operator at size μ, i.e. acting on an image by an SE of size μ. Note that this property implies idempotence: $\mu = \lambda \rightarrow \psi_\mu \psi_\mu = \psi_\mu \psi_\mu$.

Figure 23 shows SEs of different sizes, illustrating what is meant by an SE of size μ.

Definition 48. A *granulometry with size parameter* μ is an operator ψ that is (a) increasing, (b) anti-extensive, (c) absorbing, and that (d) generates a family of operators $\{\psi_\mu\}$.

Recall that a grey tone image can be interpreted as a digital elevation model having volume, where the volume of an image object is determined by the surface area of the object as well as the grey values. A granulometric curve or pattern spectrum is formed by plotting the loss of volume (or surface area in the case of a binary image) between ψ_μ and $\psi_{\mu+1}$ against $\mu + 1$. A peak in the pattern spectrum indicates that a large number of objects can be found at that particular size [29]. A binary image and its granulometry are shown in Figure 24.

2.1.11 Path-Based Morphology

This section refers back to Section 2.1.2 for preliminary theory on connectivity and adjacency graphs.

Let E be the domain of an image, i.e. $E \subset \mathbb{Z}^2$ and let $E^\star$ be a directed graph on the points in E, defined by an adjacency relation "$\mapsto$". Let $\mathcal{F}(E^\star)$ and $\mathcal{G}(E^\star)$ be the set of all binary and grey tone

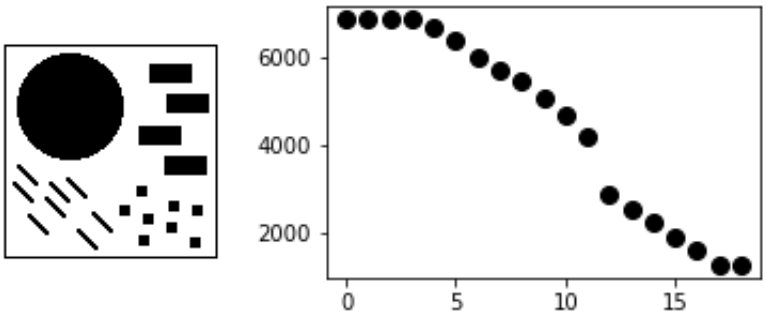


Figure 24: A binary image with its granulometry.

images, respectively, defined on $E^\star$. For every point $\mathbf{x}$ in $E^\star$ we have the set of predecessors as well as the set of successors of $\mathbf{x}$, a fact which makes it possible to state the following definition.

Definition 49. For each point $\mathbf{x}$ in $E^\star$, the *dilation* $\delta(\{\mathbf{x}\})$ is defined as the set of successors of $\mathbf{x}$, i.e.

$$\delta(\{\mathbf{x}\}) = \{\mathbf{y} \in E^\star : \mathbf{x} \mapsto \mathbf{y}\}.$$

Let X be an arbitrary subset of $E^\star$. Then the dilation on X, $\delta(\{X\})$ is defined as the set of *successors* of every point $\mathbf{x}$ in X, i.e.

$$\delta(\{X\}) = \{\mathbf{y} \in E^\star : \mathbf{x} \mapsto \mathbf{y} \text{ for some } \mathbf{x} \in X\}.$$

Similarly, we can define the set $\check{\delta}(\mathbf{x})$ as the set of all *predecessors* of $\mathbf{x}$, i.e.

$$\check{\delta}(\{\mathbf{x}\}) = \{\mathbf{x} : \mathbf{x} \mapsto \mathbf{y} \text{ for some } \mathbf{y} \in E^\star\}.$$

$\check{\delta}(X)$ is then the set of all points having a successor in X, i.e.

$$\check{\delta}(\{X\}) = \{\mathbf{x} \in E^\star : \mathbf{x} \mapsto \mathbf{y} \text{ for some } \mathbf{y} \in X\}.$$

The dilation operator is used to define a δ-path, a fundamental concept in path-based morphology.

Definition 50. A *δ-path of length L* is an L-tuple $\mathbf{a} = (a_1, a_2, ..., a_L)$ such that $a_{k+1} = \delta(\{a_k\})$ for $k = 1, 2, ..., L-1$, i.e. if $a_k \mapsto a_{k+1}, \qquad k = 1, 2, ..., L-1$. This means that each element of the tuple is a predecessor of the next element.

The *reverse path* $\check{\mathbf{a}}$ is an L-tuple given by $\check{\mathbf{a}} = (a_L, a_{L-1}, ..., a_1)$. Each element is therefore a successor of the next element. $\check{\mathbf{a}}$ is called a *$\check{\delta}$-path of length L* since $\check{a}_{k+1} = \check{\delta}(\{\check{a}_k\})$ for $k = 1, 2, ..., L-1$ where $\check{a}_k = a_{L+1-k}, \qquad k = 1, 2, ..., L$.

The set of all δ-paths of length L is denoted by Π_L while the set of all $\check{\delta}$-paths of length L is denoted

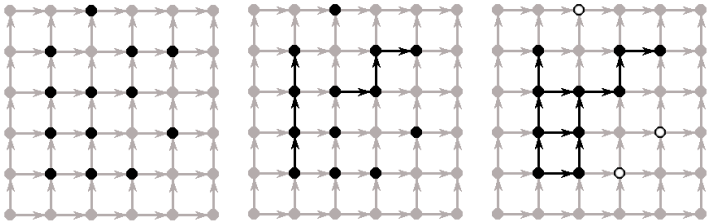


Figure 25: Example of path opening. The subset X is given in a), two possible paths are displayed in b) and c) shows the path opening. The white dots with black borders have been excluded.

by $\check{\Pi}_L$.

Definition 51. For any path $\mathbf{a} = (a_1, a_2, ..., a_L)$ in $E^\star$, the set of all the elements of $\mathbf{a}$ is given by

$$\sigma(\mathbf{a}) = \sigma(a_1, a_2, ..., a_L) = \{a_1, a_2, ..., a_L\}.$$

The set of all δ-paths of length L completely contained in a subset X of $E^\star$ is given by

$$\Pi_L(X) = \{\mathbf{a} \in \Pi_L : \sigma(\mathbf{a}) \subseteq X\}.$$

Definition 52. The *path opening* α_L with respect to the integer L and the subset X of $E^\star$ is defined by

$$\alpha_L(X) = \bigcup\{\sigma(\mathbf{a}) : \mathbf{a} \in \Pi_L(X)\},$$

namely it is the union of all δ-paths of length L in X.

The *path closing* β_L is obtained in the same way as the path opening, but using a subset X of the background of the image instead of the foreground, having the same adjacency relation [32].

Remark 53. Note that the path opening and path closing have all the properties of an opening and a closing respectively, i.e. both are increasing and idempotent, the path opening is anti-extensive and the path closing is extensive [11].

Example 54. The path opening of a subset of points on a directional graph is shown in figure 25. The graph itself, $E^\star$, is shown in light grey while the subset of points X is shown by the black dots. In figure 25 b), two possible paths of length $L = 3$ are shown. The path opening, the union of all δ-paths of length 3 in X, is shown in c). The white dots with black borders could not be connected to the other dots by δ-paths of length 3 and were therefore excluded by the opening. □

Determining path openings can be computationally intensive. Figure 26 shows four graphs with the associated number of δ-paths of length $L = 3$. The first graph is fairly small and simple with only 4 rows and 3 columns, and northward (N) adjacency. There are 3 possible δ-paths of length 3. The graph

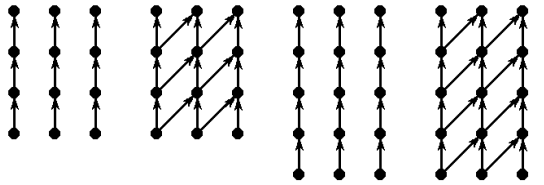


Figure 26: Demonstration of the increase in the number of δ-paths as graph size and adjacency complexity increases. 4×3 graphs with N and NNE adjacency, respectively, are shown in a) and b) while 5×3 graphs with the same respective adjacencies are shown in c) and d).

can be made more complex by increasing the size, as in c), where the addition of one row increased the number of δ-paths to 6, or by introducing more complex adjacency, as in b), where the northeast (NE) adjacency was added. This resulted in a graph of the same size, but with northward and northeastward (NNE) adjacency, which increased the number of δ-paths to 12. Finally, d) shows a 5×3 graph with NNE adjacency, which had 24 possible δ-paths. The number of δ-paths for a large image with a complex adjacency rule will be very great. A decomposition algorithm which greatly reduces the computational effort required is provided in [11].

It may sometimes be necessary to work with long paths (large values of L). However, long paths have the tendency to include noise [11]. To circumvent this problem and reduce the amount of noise, as well as to create a more flexible operator that is not excessively constrained by the exact number of points in the path, the concept of the more flexible incomplete path opening is introduced. The idea behind the incomplete path opening is that a few points are allowed to lie outside of X.

Let $\Pi_L^k(X)$ be the set of all paths of length L inside $E^\star$ with at least $L-k$ points inside X, $0 \leq k \leq L$.

Definition 55. The incomplete path opening $\alpha_L^k(X)$ is defined as the union of all paths in $E^\star$ such that at least $L-k$ points of the path lies inside X:

$$\alpha_L^k(X) = \bigcup \{\sigma(\mathbf{a}) \cap X : \mathbf{a} \in \Pi_L^k(X)\}, \qquad k = 1, 2, ..., L-1.$$

The path opening operator, incomplete path opening and the decomposition algorithm can be generalised to the grey tone case using threshold decomposition, a concept discussed in the section **Images and their Representations** [30]. A new paramater ν, representing grey value, is introduced for this purpose. Let g be a grey tone image and let $X_\nu(g)$ be the *cross section* of g at ν, i.e. $X_\nu(g) = \{\mathbf{x} \in E^\star : \nu \leq g(\mathbf{x}) \leq t\}$ where t is the maximum value of g on the domain $X \subseteq E^\star$.

The set of all δ-paths of length L at grey value ν is given by

$$\Pi_L^\nu(g) = \{a \in \Pi_L : g(a_k) \geq \nu, k = 1, 2, ..., L\}.$$

a) b) c)

Figure 27: An image with path openings: a) The original image. b) An incomplete path opening with parameters $L = 400$ and $k = 10$. c) A path opening of length $L = 400$.

The following relation holds [11]: $\Pi_L^\nu(g) = \Pi_L(X_\nu(g))$.

Definition 56. Let V be the domain of g. Then the grey scale opening is defined as:

$$[\mathcal{A}_L(g)](a) = \max\{\nu \in V : a \in \sigma(\mathbf{a}) \text{ for some } \mathbf{a} \in \Pi_L^\nu(g)\}.$$

Example 57. Figure 27 shows an image with the results of a complete and an incomplete path opening. In both of the results, the houses and patches of bare soil that are not connected to the roads have become fainter. In the incomplete path opening shown in b), a maximum of 10 pixels are allowed to interrupt the path. Many of the bare soil patches that are connected to the roads are still bright. In the complete path opening shown in c), however, most of the non-linear patches of bare soil have become faint. □

2.1.12 Path-Based Morphological Profiles

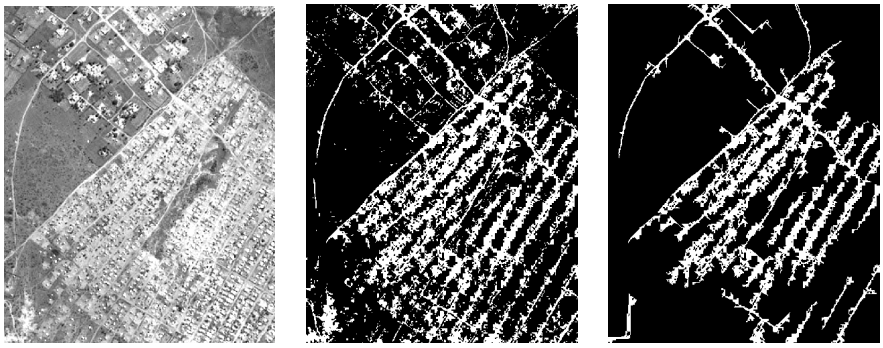
A morphological profile (MP) function is a "fuzzy membership function related to a set of morphological characteristics of the connected components in the image" and is useful for segmenting satellite images [20]. Ghamisi [7] states that MPs can be used to model spatial information in images. MPs are constructed by performing a sequence of openings and closings by an SE of increasing size on an image [7]; a more detailed discussion on the construction of MPs in the case of classical SE-based morphology may be found in [20].

As has been discussed previously in this report, classical morphology has the disadvantage of being severely constrained by the choice of the SE or SEs used. A solution to this problem is to construct MPs using path operators [14].

Definition 58. The path-based morphological profile $MP(\mathbf{x})$ of a pixel $\mathbf{x}$ is defined as

$$\text{MP}(\mathbf{x}) = \{\Pi_L(\mathbf{x})\}, L = 1, 2, ..., L_{max},$$

where L_{max} is the length of the longest path applied [14]. A path-based MP of a subset X, MP(X) is



a) b) c)

Figure 28: A satellite image with a binary thresholded version of the image, and a path opening. a) The original satellite image. b) The binary thresholded image. c) A complete path opening with path length $L = 100$ applied to the binary image in b).

thus created by applying a sequence of path openings of increasing length to the set of points or pixels X.

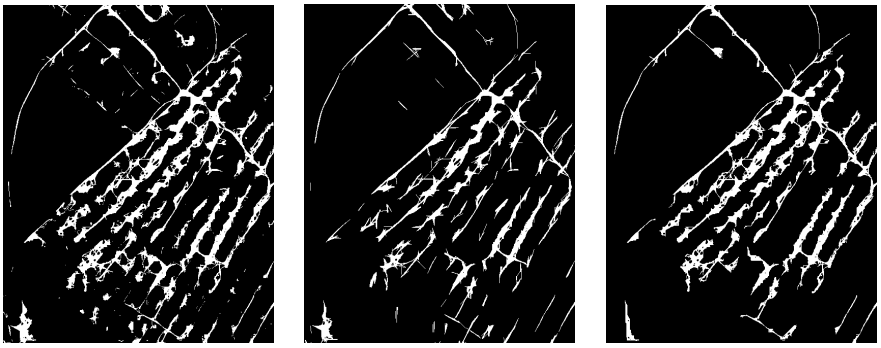
2.1.13 Comparison of Classical Openings and Path Openings Applied to Binary Images

This section compares the results of classical and path-based openings on the same image, shown in Figure 28b).

Applying a path opening directly to Figure 28b) results in the image shown in Figure 28c). In this case, the non-directionality of the path opening was a disadvantage, since the white areas were mostly connected. In order to separate the connected areas, the image was eroded using 28 SEs, obtained by rotating a 15-pixel straight line at all possible angles given a 15×15 grid. Refer to figure 37 in the Appendix for the SEs and Figures 38 and 39 in the Appendix for the erosions of the image using the different SEs.

The result of these erosions is given in Figure 29a). An opening using the same 28 SEs was applied to this eroded image, as was a complete path opening. The results are shown in Figure 29b) and c) respectively. In this case, the path opening was less noisy and preserved curvilinear structures better than the traditional opening.

In conclusion, binary path openings may not be effective when too many of the image objects are connected. However, provided there is enough separation between the objects, path openings will give a less noisy result with better preservation of curvilinear structures, as well as being simpler to calculate than classical openings. This is due to the fact that a vast number of SEs must be used to avoid the directional nature of the results of classical openings.



a) b) c)

Figure 29: Comparison of the results of a classical opening and a path opening. a) The original image. b) A classical opening using 28 SEs. c) A complete path opening with length $L = 100$ performed on the erosion from the classical morphology shown in a).

2.2 Binary Partition Trees

2.2.1 Connected Components and Operators

This section presents some definitions critical to the understanding of binary partition trees. It refers back to the sections **Graphs, Connectivity and Adjacency** and **Path-Based Morphology**.

Definition 59. A connected component of a set X with adjacency rule $\mapsto$ is the set of points that can be connected by a path inside X [24].

Definition 60. A flat zone FZ is the largest connected component of an area in an image where the value of the image is constant [23].

Definition 61. Suppose an image f is defined on a set E with adjacency rule $\mapsto$. Now suppose $\{R_i\},\ i = 1, 2, ..., n$ for some $n > 0$ is a set of connected components of E that are mutually disjoint. Then $\{R_i\}$ is a *partition* of E if $E = \bigcup_{i=1}^{n} R_i$, i.e. the union of the R_i^s makes up E. The R_i^s are called the *regions* of f [24].

Suppose an image f has n flat zones. When f is divided into its flat zones, these flat zones create a partition. Let $FZ = \{FZ_i,\ i = 1, 2, ..., n\}$ be the set of these flat zones. Then the partition of f is the union of all the flat zones, i.e. $\bigcup_{i=1}^{n} FZ_i$ [23].

Definition 62. Let $P_A = \bigcup_{i=1}^{n} R_i^A$ and $P_B = \bigcup_{j=1}^{m} R_j^B$ be two partitions of an image. Then P_A is said to be *finer* than P_B if the following holds:

$$\mathbf{p}_1, \mathbf{p}_2 \in R_i^A \ \Rightarrow\ \exists j\ \ni\ \mathbf{p}_1, \mathbf{p}_2 \in R_j^B \qquad \forall i = 1, 2, ..., n,\ \mathbf{p}_1, \mathbf{p}_2.$$

This means that any two pixels in the same region of partition P_A are also in one unique region of partition P_B [23, 24].

Figure 30: Two partitions P_B and P_A of the same image. P_A is finer than P_B.

Example 63. Two partitions P_B and P_A of the same image are shown in Figure 30. P_A has been created by subdividing the regions of P_B and hence any two pixels in one region of P_A will also be in one region of P_B. P_A is therefore clearly finer than P_B. □

Definition 64. Let f be a grey tone image. A *connected operator* ψ acting on images is an operator that merges flat zones in an image, i.e. the partition of the flat zones of f is finer than the partition of the flat zones of $\psi(f)$ [23, 24].

2.2.2 Binary Partition Trees

Definition 65. A Binary Partition Tree (BPT) is a hierarchical tree-based representation of the regions that can be obtained from an initial partition of an image [23, 34].

The hierarchy represents the image at different scales and can be obtained by storing the steps in some region merging algorithm [33, 23]. A BPT provides a natural representation of images that is suited for a variety of applications [34] and allows for the quick application of complex image processing techniques [23]. It also leads to a significant reduction in computation time since not all the relationships between regions need to be stored or analysed.

Refer to Figure 31. The leaves of the tree, i.e. the nodes found at the bottom of the BPT structure, are the regions resulting from an initial partition. These regions are merged iteratively based on a *region model* and a *merging order* until some stopping criterion is satisfied. In the case of Figure 31, the stopping criterion is that there should be only one region, i.e. all the regions should be merged.

Definition 66. The region model and merging order are now defined.

- The *region model* M_R is a vector containing the values of the properties of interest of a region.
- The *merging order* $O(R_i, R_j)$ is a measure of similarity between two regions R_i and R_j. The exact definition of $O(R_i, R_j)$ depends on the problem.

Some merging criteria are statistical in nature, such as the MSE. This and other examples of merging criteria and region models may be found in [36].

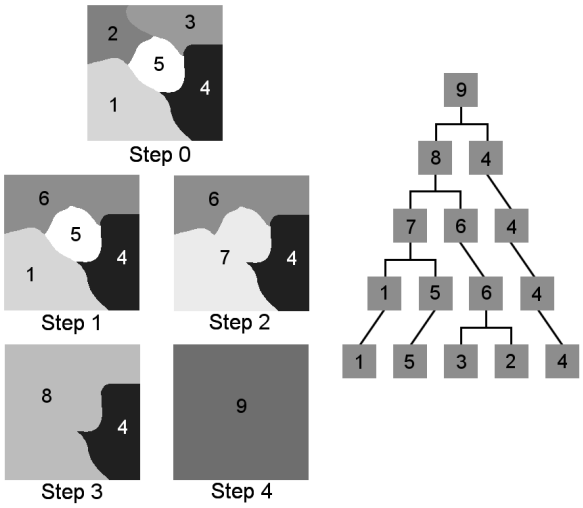


Figure 31: Example of a binary partition tree constructed by keeping track of the merging steps in a region merging algorithm. The initial partition is shown in step 0 while steps 1 to 4 show the merging steps.

Example 67. The step-by-step results of a region merging algorithm with associated BPT are shown in figure 31. The grey values were represented by the Hue-Saturation-Intensity (HSI) model. Since all the grey values had the same hue (namely 255) and saturation (namely 0), the region model M_R was an integer between 0 and 255 representing a region's intensity or grey value. The merging criterion $O(R_i, R_j)$ was defined as the average of the grey values of the two regions R_i and R_j, i.e.

$$O(R_i, R_j) = \frac{(M_{R_i} + M_{R_j})}{2}.$$

Step 0 shows the initial partition, which is the result of segmentation. Table 1 gives the absolute differences in the intensity between the five initial regions. The two regions with the smallest absolute difference, namely regions 2 and 3, were merged to form region 6 (with an intensity of $\frac{128+154}{2} = 141$). In step 2, the absolute differences between regions 1, 4, 5 and 6 were compared and regions 1 and 5 were merged to form region 7. This process was repeated until there was only one region, namely region 9. The steps followed are schematically represented in the BPT in figure 31. Note that the only relationships stored between regions are the relationships between each parent node and its child or children. There is no need to store all the relationships between the regions or all the possible ways in which the regions could have been merged. This leads to a reduction in computation time and effort, as well as storage space. □

Remark 68. Note that the initial partition could be the pixels of the image [33], the set of flat zones or any other initial segmentation [23]. A major advantage of using a segmentation that creates image objects over simply using the pixels is the relationship between image objects and real-world objects [2].

	Region	1	2	3	4	5
	Intensity	214	128	154	32	255
Region	Intensity	Absolute difference in intensity between regions				
1	214	0	86	60	182	41
2	128	86	0	26	96	127
3	154	60	26	0	122	101
4	32	182	96	122	0	223
5	255	41	127	101	223	0

Table 1: Absolute differences in intensity between regions. These differences are used to determine the order in which regions should be merged.

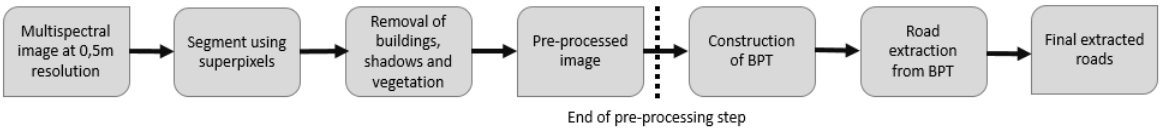


Figure 32: The application process.

There is no inherent relationship between image pixels and the real-world objects they represent.

A typical algorithm for BPT construction can be found in Valero et al. [33]. Np refers to the number of regions in the initial partition.

```
Get O(Ri,Rj) values between all connecting regions
i=0
While i < Np - 1:
    1. Rank O(Ri,Rj) values so that O(Ri,Rj) of most similar region pair is in first position
    2. Merge regions Ri and Rj to form Rij
    3. Update RAG by removing edge between Ri and Rj and creating new edges of Rij
    4. Update list of O(Ri,Rj)s by adding O(Rij,Rk) for all k
    5. i += 1
End while
```

3 Application

An overview of the method is given in Figure 32. The preprocessing, construction of the BPT and road extraction were all done in Matlab using code provided by Dr. Mengmeng Li of the Faculty of GeoInformation Science and Earth Observation, University of Twente. The quality assessment was done using Python. The code for the quality assessment is given in the Appendix.

3.1 Problem Statement and Study Area

The study areas are both located in Mabopane, Tshwane Municipality, Gauteng Province, South Africa. The informal settlements in this area are in many cases starting to formalise and roads are beginning to

a)
Coordinates: 28°4'7"E 25°30'53"S

b)
Coordinates: 28°1'26"E 25°31'25"S

Figure 33: The study areas, located in Mabopane, Tshwane Municipality, Gauteng Province, South Africa: a) Study area 1. b) Study area 2. The images are from The images are from Pléiades-1B.

take on a grid-like structure. The data used were two very high resolution (VHR) multispectral images, with a spatial resolution of 0.5m, from the Pléiades-1B satellite. Each image consisted of 4 bands: 3 bands in the visible spectrum, namely Red, Green and Blue (RGB) and a Near-Infrared (NIR) band. The study areas are shown in Figure 33. Due to the visually indistinct nature of many of the informal roads, the expectation of this project was to determine which kinds of roads could be identified by the proposed method, rather than to achieve a high accuracy.

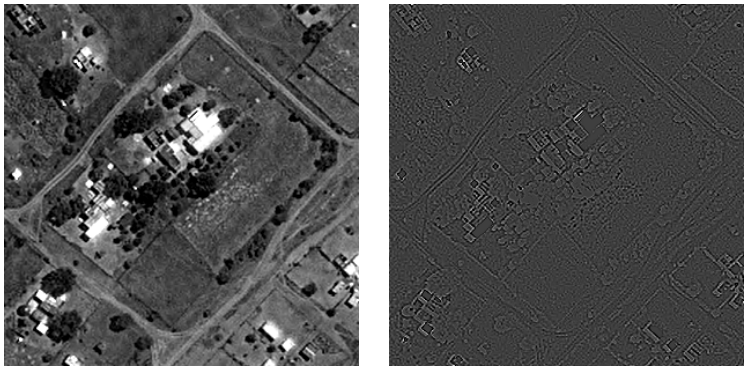
3.2 Data Preprocessing

Each image was segmented by applying the superpixels technique with 10000 segments to the RGB bands. For each image, the Normalised Difference Vegetation Index (NDVI) was computed using the formula

$$\text{NDVI}=\frac{\text{NIR}-\text{R}}{\text{NIR}+\text{R}}.$$

Vegetation removal was performed by applying a threshold to the NDVI computed from the images. Li et al. [14] applied Otsu's threshold [19]. However, this thresholding method delivered sub-optimal results for both regions in this paper, as this eliminated much of the bare soil areas, making the detection of dirt roads impossible. In order to include the bare soil, an experimentally determined threshold of 0.3 was used, which had to be higher than Otsu's threshold.

The shadow regions were removed by converting the NIR-RGB colour space to a a Hue-Saturation-Intensity (HSI) colour space and applying Otsu's threshold to an index computed from this new colour space, namely $\frac{\text{Saturation - Intensity}}{\text{Saturation + Intensity}}$, as proposed by [31]. To avoid misdetection of dark objects as shadows, each image was subset into smaller parts, as in [14], and tree shadows were removed by considering the directional relationship between trees and their shadows. The directional relationship between buildings and their shadows was considered in order to remove buildings, as in [14]. The segments that were not classified as trees, buildings or shadows were used to construct the BPTs.



a) b)

Figure 34: An image with its intensity gradient: a) An image showing some buildings and informal roads within the study area. b) The Laplacian intensity gradient of the image.

3.3 Construction of the Binary Partition Tree

For this project, the region model M_R consisted of the following components: two geometrical features, the elongation and compactness of a region, and two structural features based on orientation histograms and morphological profiles respectively.

The elongation and compactness of a region R_i are respectively obtained as follows:

$$\text{elongation}_i = \frac{\text{area}_i}{\text{length}_i^2}$$

$$\text{compactness}_i = \frac{2\sqrt{\pi}}{\text{perimeter}_i},$$

where area_i, length_i, and perimeter_i are the area, length and perimeter respectively of region R_i.

The orientation histogram of region R_i is denoted by HOG_i and describes the distribution of the directions of local intensity gradients in the region [5]. Intensity (grey value) gradients can be used to identify edges, where the change in intensity is great, as shown in Figure 34. The directions of such edges are binned to create an orientation histogram. The orientation histograms are computed as in [5], with the difference that the minimum local neighbourhood is enforced for every HOG_i. The orientation histogram HOG_i is computed over the bounding box of the region R_i, however, this region may be very small. To ensure that the neighbourhood is not too small, a minimum cell size w_{cell} is introduced. If either the height or the width of the bounding box is smaller than w_{cell}, the bounding box is buffered. This method was proposed and implemented in [14].

Recall from Section 2.1.12 that the path-based morphological profile $MP(\mathbf{x})$ of a pixel $\mathbf{x}$ is defined as

$$\text{MP}(\mathbf{x}) = \{\Pi_L(\mathbf{x})\},\ L = 1, 2, ..., L_{max},$$

where L_{max} is the length of the longest path applied [14].

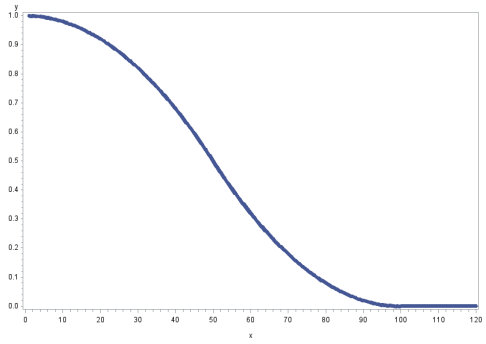


Figure 35: A Z-shaped fuzzy membership function with parameters $a = 0$ and $b = 100$.

A path-based MP of a subset X, MP(X) is created by applying a sequence of path openings of increasing length to the set of points or pixels X. The connectivity feature is defined as

$$confeature_i = \frac{1}{n_i} \sum_{p=1}^{n_i} \sum_{L=0}^{L_{max}} w_L \Pi_L(x_p),\ x_p \in R_i,$$

where w_L is the weight of a path opening with length L. The weights can be calculated using a linear or a Gaussian function.

The merging order $O(R_i, R_j) = O_{ij}$ was computed using the parameters $\mu_{1,i}$, $\mu_{2,i}$, $\mu_{3,i}$ and $\mu_{4,i}$. Let the subscript ij denote region R_{ij}, which is the union of adjacent regions R_i and R_j.

$\mu_{1,ij}$ and $\mu_{2,ij}$ are the membership values of the compactness and elongation of regions R_{ij} as defined by a Z-shaped fuzzy membership function, i.e.

$$\mu_{1,ij} = f(compactness_{ij}; 0, b_1)$$
$$\mu_{2,ij} = f(elongation_{ij}; 0, b_2),$$

where

$$f(x; a, b) = \begin{cases} 1 & x \leq a \\ 1 - 2(\frac{x-a}{b-a})^2 & a \leq x \leq \frac{a+b}{2} \\ 2(\frac{x-b}{b-a})^2 & \frac{a+b}{2} \leq x \leq b \\ 0 & x \geq b \end{cases}$$

is the membership function. A visual representation of such a function is shown in Figure 35.

The orientation similarity between regions R_i and R_j is given by

$$\mu_{3,ij} = \max\{f_{bin_i}(R_i) - f_{bin_i}(R_j), f_{bin_j}(R_i) - f_{bin_j}(R_i)\},$$

where $f_{bin_k}(R_l)$ is the normalised frequency of HOG_l at the bin location $bin_k,\ k, l \in i, j$. Note that the histogram is normalised by dividing it by its maximum value, therefore $f_{bin_k}(R_k) = 1,\ k \in i, j$.

Finally, $\mu_{4,ij}$ is given by

$$\mu_{4,ij} = \frac{\text{area}_i \text{confeature}_i + \text{area}_j \text{confeature}_j}{\text{area}_i + \text{area}_j}.$$

The merging order O_{ij} is now defined as

$$O_{ij} = \alpha\sqrt{(\mu_{1,ij}\mu_{2,ij})} + (1-\alpha)\sqrt{(\mu_{3,ij}\mu_{4,ij})} + \varepsilon,$$

where

$$\varepsilon = \frac{|width_{ij} - \max\{width_i, width_j\}|}{width_{ij}}$$

limits the width of the merged regions and α is a predefined weight. The higher the value of α, the higher the relative importance attached to the geometrical features and the lower the importance of the structural features, and vice versa. In this project, the value of α was 0.5.

The parameter values used to construct the BPTs are given in Table 2. The parameters not given in this table were assigned the same values as in [14]. All other parameters were assigned the same values as in [14].

Parameter	Value for Area 1	Value for Area 2	Use
b_1	0.7	0.8	Used to define the Z-shaped fuzzy membership functions.
b_2	0.5	0.7	

Table 2: Parameters used to construct the BPT.

3.4 Road Extraction

The method of [25] for automatically extracting objects from VHR images was applied to the BPT representation. The membership degree of region R_i being a road was based on the geometric features $\mu_{1,i}$ and $\mu_{2,i}$. For each object, a possibility measure that the object was a road or a non-road, as well as a necessity measure that the object was a road or a non-road was calculated.

$$\begin{aligned}
\text{Possibility measure that } R_i \text{ is a road:} &\quad \Pi(R_i) = \max\{\mu_{1,i}, \mu_{2,i}\} \\
\text{Possibility measure that } R_i \text{ is a non-road:} &\quad \Pi(\bar{R}_i) = \max\{1-\mu_{1,i}, 1-\mu_{2,i}\} \\
\text{Necessity measure that } R_i \text{ is a road:} &\quad N(R_i) = 1 - \Pi(\bar{R}_i) \\
\text{Necessity measure that } R_i \text{ is a non-road:} &\quad N(\bar{R}_i) = 1 - \Pi(R_i)
\end{aligned}$$

R_i was classified as a road if $\Pi(R_i) > \Pi(\bar{R}_i)$ and $N(R_i) > N(\bar{R}_i)$.

a) b)

Figure 36: Results. a) Results for the first study area. b) Results for the second study area.

Quality Measure	Value for Area 1	Value for Area 1 (Unpaved Roads Only)	Value for Area 2	Value for Area 2 (Unpaved Roads Only)
Completeness = $\frac{TP}{TP+FN}$	89%	86%	74%	61%
Correctness = $\frac{TP}{TP+FP}$	20%	21%	13%	6%
Quality = $\frac{TP}{TP+FP+FN}$	33%	53%	30%	26%
Commission = $\frac{FP}{TP+FN}$	167%	61%	142%	130%
Omission = $\frac{FN}{TP+FN}$	11%	14%	26%	40%

Table 3: Assessment of the quality of the results.
(Notation: Number of pixels exhibiting TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative)

3.5 Results

The obtained results are given in Figure 36, with corresponding quality assessment in Table 3. The method proved robust with regards to the presence of cars on the road, and produced good results in the areas with broad, unpaved roads ($12-15$m wide including road shoulders and sidewalks). The unpaved areas around paved roads were detected as roads, thereby increasing the number of false positives. This occurrence is a natural result of considering both types of surfaces, however, and does not reflect badly on the method. If only paved roads were to be considered, the NDVI threshold could be made lower in the preprocessing step, which would filter out more of the bare soil. A great number of false positives was due to other misdetections. Most notable among these was the large patch of bare soil in the centre of the first study area which was detected as road. The problem of false positives was less prevalent around the unpaved roads in the first study area, suggesting that the method may be suitable for detecting broad, unpaved roads in an urban context. Fewer false positives were detected in the second study area, but more false negatives were detected. Masking out the formal roads resulted in a high omission rate leading to a very low correctness score. Despite this, the commission rate remains significantly higher than the omission rate, which agrees with the problem of false positive detection experienced in [21].

4 Conclusion

On a theoretical level, this project discussed the theory of mathematical morphology, the study of shape. It was highlighted that the dependence of classical morphology on a preset structural element is a significant disadvantage, which may be overcome by using path-based morphology. However, path-based binary openings may not produce useful results if too many of the image components are connected.

On a practical level, this project showed contributions in the modelling of the uncertainty of spatial objects as extracted from spatial big data, namely the identification of unpaved, informal roads. This is a topic of great practical importance which has not yet been widely addressed.

The state of the art road detection method proposed in [14] was applied in a South African context. The method produced satisfactory results in the detection of broad, straight, unpaved roads, and proved robust with regards to the presence of cars on roads. It may therefore be a suitable method for the detection of such roads. In the case of less formal, narrower unpaved roads, the method had a high number of false negatives and may not be appropriate.

Future research is required to accurately detect narrower, more winding informal roads, as well as to precisely determine the boundaries of roads and reduce the number of false positives. The ideal NDVI thresholds for identifying bare soil roads as well as for identifying paved roads have yet to be determined. LiDAR data may also be taken into account in future, since it would provide height information which would be invaluable in differentiating tall objects (such as buildings) from roads.

The misdetection of unpaved areas as roads, especially those areas that may be used for navigation, raises an interesting question: what precisely is a road and when and how do we differentiate a road from a pathway? When venturing into the area of informal or unpaved roads, created by citizens on an ad hoc, purely pragmatic basis, what exactly constitutes a road becomes unclear. This question invites future discussion, and how it is answered must undergird any future attempt at the detection of informal roads.

References

[1] J Amini, MR Saradjian, JAR Blais, C Lucas, and A Azizi. Automatic road-side extraction from large scale imagemaps. *International Journal of Applied Earth Observation and Geoinformation*, 4(2):95–107, 2002.

[2] Ursula C Benz, Peter Hofmann, Gregor Willhauck, Iris Lingenfelder, and Markus Heynen. Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. *ISPRS Journal of photogrammetry and remote sensing*, 58(3):239–258, 2004.

[3] Luca Bertolini, Frank le Clercq, and Loek Kapoen. Sustainable accessibility: a conceptual framework to integrate transport and land use plan-making. Two test-applications in the Netherlands and a reflection on the way forward. *Transport Policy*, 12(3):207–220, 2005.

[4] Geneviève Boisjoly and Ahmed M El-Geneidy. How to get there? A critical assessment of accessibility objectives and indicators in metropolitan transportation plans. *Transport Policy*, 55:38–50, 2017.

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[6] Karst T Geurs and Bert van Wee. Accessibility evaluation of land-use and transport strategies: review and research directions. *Journal of Transport Geography*, 12(2):127–140, 2004.

[7] Pedram Ghamisi, Mauro Dalla Mura, and Jon Atli Benediktsson. A survey on spectral–spatial classification techniques based on attribute profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5):2335–2353, 2015.

[8] Pijush K Ghosh. A mathematical model for shape description using Minkowski operators. *Computer Vision, Graphics, and Image Processing*, 44(3):239–269, 1988.

[9] Daniel A Griffith. *Spatial Autocorrelation and Spatial Filtering: Gaining Understanding through Theory and Scientific Visualization*. Springer Science & Business Media, 2013.

[10] Robert M Haralick, Stanley R Sternberg, and Xinhua Zhuang. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4):532–550, 1987.

[11] Henk Heijmans, Michael Buckley, and Hugues Talbot. Path openings and closings. *Journal of Mathematical Imaging and Vision*, 22(2):107–119, 2005.

[12] Arnon Karnieli, Amnon Meisels, Leonid Fisher, and Yaacov Arkin. Automatic extraction and evaluation of geological linear features from digital remote sensing data using a Hough transform. *Photogrammetric Engineering and Remote Sensing*, 62(5):525–531, 1996.

[13] Jonathan Levine, Louis Merlin, and Joe Grengs. Project-level accessibility analysis for land-use planning. *Transport Policy*, 53:107–119, 2017.

[14] Mengmeng Li, Alfred Stein, Wietske Bijker, and Qingming Zhan. Region-based urban road extraction from VHR satellite images using binary partition tree. *International Journal of Applied Earth Observation and Geoinformation*, 44:217–225, 2016.

[15] Suxia Liu and Xuan Zhu. An integrated GIS approach to accessibility analysis. *Transactions in GIS*, 8(1):45–62, 2004.

[16] Weifeng Liu, Zhenqing Zhang, Shuying Li, and Dapeng Tao. Road detection by using a generalized Hough transform. *Remote Sensing*, 9(6):590, 2017.

[17] Petros Maragos and Ronald Schafer. Morphological filters–part I: their set-theoretic analysis and relations to linear shift-invariant filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(8):1153–1169, 1987.

[18] Juan B Mena. State of the art on automatic road extraction for GIS update: a novel classification. *Pattern Recognition Letters*, 24(16):3037–3058, 2003.

[19] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[20] Martino Pesaresi and Jon Atli Benediktsson. A new approach for the morphological segmentation of high-resolution satellite imagery. *IEEE transactions on Geoscience and Remote Sensing*, 39(2):309–320, 2001.

[21] JA Quintanilha RAA Nobrega, CG O'Hara. Detecting road in informal settlements surrounding Sao Paulo city by using object-based classification. In *1st International Conference on Object-Based Image Analysis (OBIA)*, 2006.

[22] Jean-Francois Rivest, Pierre Soille, and Serge Beucher. Morphological gradients. In *SPIE/IS&T 1992 Symposium on Electronic Imaging: Science and Technology*, pages 139–150. International Society for Optics and Photonics, 1992.

[23] Philippe Salembier and Luis Garrido. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE Transactions on Image Processing*, 9(4):561–576, 2000.

[24] Philippe Salembier and Jean Serra. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on image processing*, 4(8):1153–1160, 1995.

[25] Imane Sebari and Dong-Chen He. Automatic fuzzy object-based analysis of vhsr images for urban objects extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 79:171–184, 2013.

[26] J Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.

[27] Jean Serra and Pierre Soille. *Mathematical Morphology and its Applications to Image Processing*, volume 2. Springer Science & Business Media, 2012.

[28] Wenzhong Shi and Changqing Zhu. The line segment match method for extracting road network from high-resolution satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, 40(2):511–514, 2002.

[29] Pierre Soille. *Morphological Image Analysis: Principles and Applications*. Springer, 1999.

[30] Hugues Talbot and Ben Appleton. Efficient complete and incomplete path openings and closings. *Image and Vision Computing*, 25(4):416–425, 2007.

[31] Mustafa Teke, Emre Başeski, Ali Ok, Barış Yüksel, and Çağlar Şenaras. Multi-spectral false color shadow detection. *Photogrammetric Image Analysis*, pages 109–119, 2011.

[32] Silvia Valero, Jocelyn Chanussot, Jon Atli Benediktsson, Hugues Talbot, and Björn Waske. Advanced directional mathematical morphology for the detection of the road network in very high resolution remote sensing images. *Pattern Recognition Letters*, 31(10):1120–1127, 2010.

[33] Silvia Valero, Philippe Salembier, and Jocelyn Chanussot. Hyperspectral image representation and processing with binary partition trees. *IEEE Transactions on Image Processing*, 22(4):1430–1443, 2013.

[34] Silvia Valero, Philippe Salembier, Jocelyn Chanussot, and Carles M Cuadras. Improved binary partition tree construction for hyperspectral images: application to object detection. In *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*, pages 2515–2518. IEEE, 2011.

[35] JR Ritsema Van Eck and Tom de Jong. Accessibility analysis and spatial competition effects in the context of GIS-supported service location planning. *Computers, Environment and Urban Systems*, 23(2):75–89, 1999.

[36] Veronica Vilaplana, Ferran Marques, and Philippe Salembier. Binary partition trees for object detection. *IEEE Transactions on Image Processing*, 17(11):2201–2216, 2008.

[37] Luc Vincent. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2(2):176–201, 1993.

[38] Weixing Wang, Nan Yang, Yi Zhang, Fengping Wang, Ting Cao, and Patrik Eklund. A review of road extraction from remote sensing images. *Journal of Traffic and Transportation Engineering (English Edition)*, 3(3):271–282, 2016.

[39] Chunsun Zhang, Shunji Murai, and Emmanuel Baltsavias. Road network detection by mathematical morphology. In *ISPRS Workshop on 3DGeospatial Data Production: Meeting Application Requirements*, pages 185–200, 1999.

Appendix

Structuring Elements and Intermediate Results for Section 2.1.13

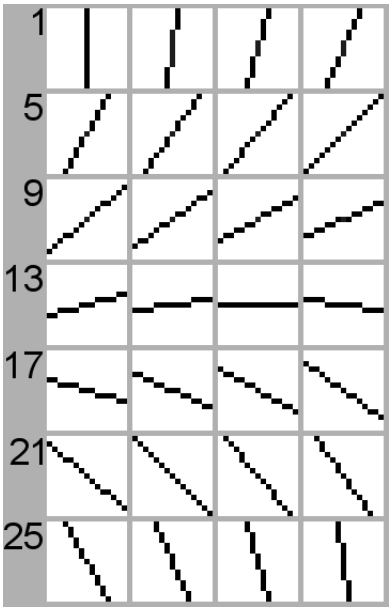


Figure 37: Structuring elements 1 to 28.

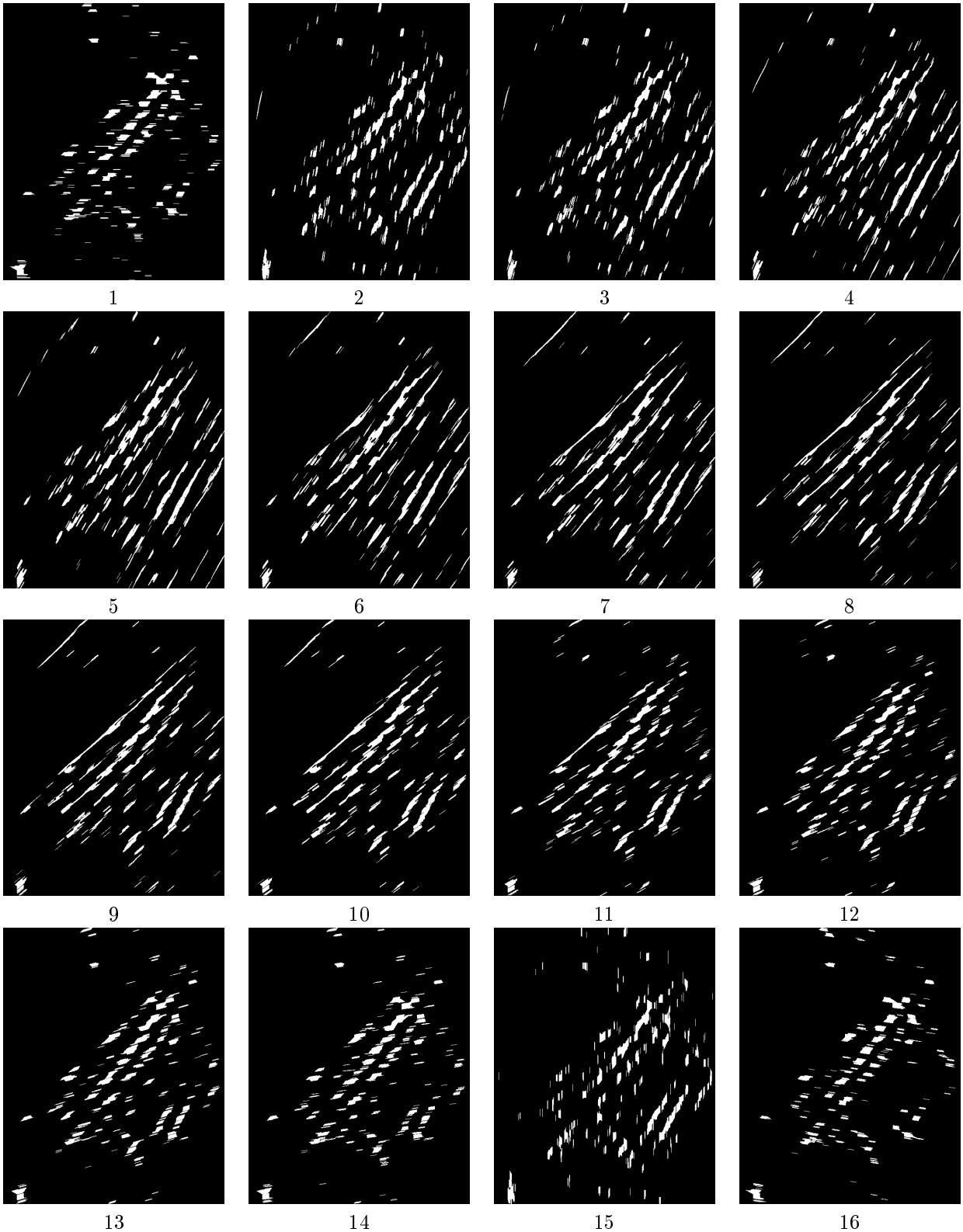


Figure 38: Double erosions by individual structuring elements 1 to 16.

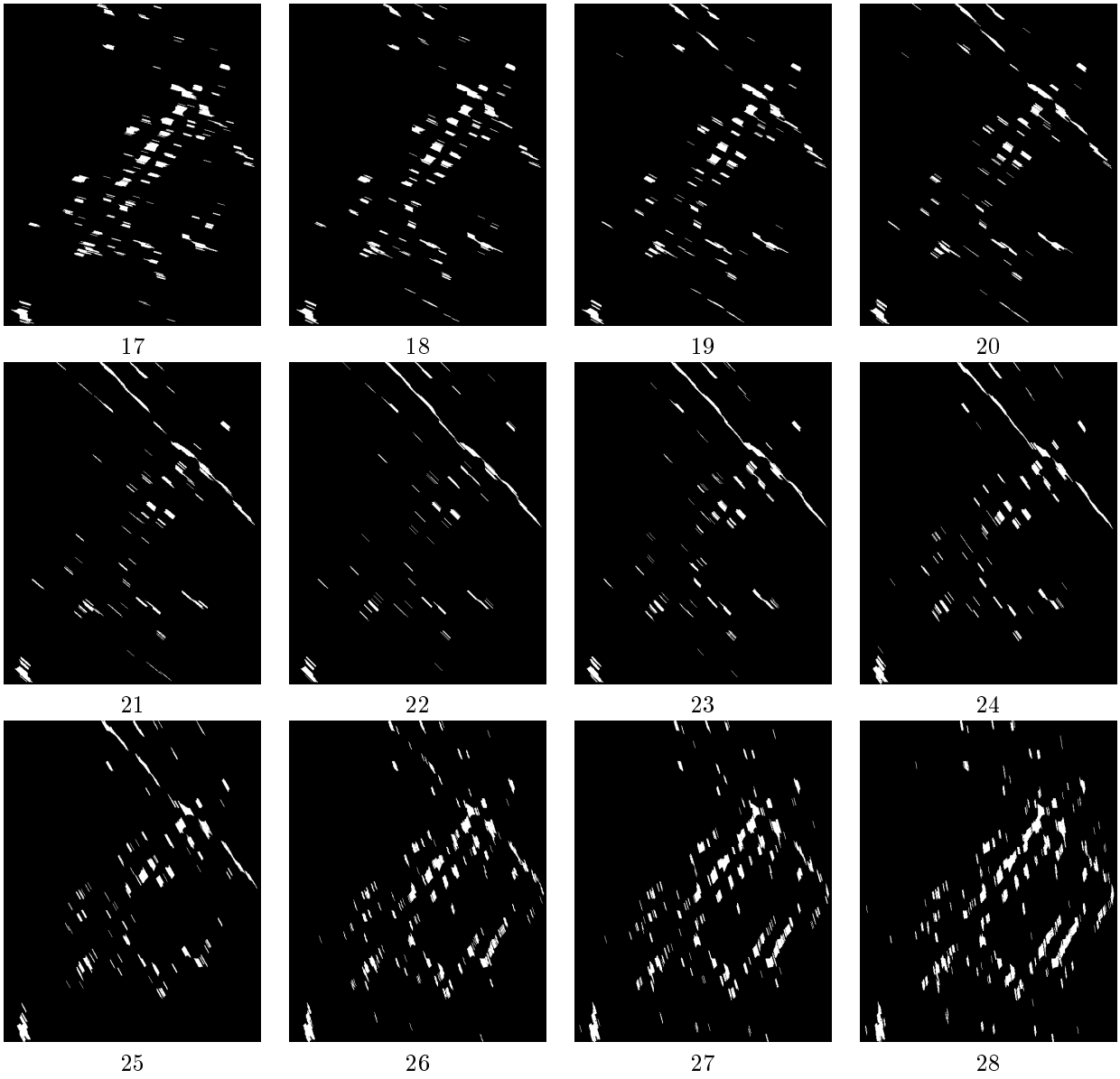


Figure 39: Double erosions by individual structuring elements 17 to 28.

Code for Drawing the Z-shaped Fuzzy Membership Function

This code is in SAS/IML.

```
1. proc iml;
2. a=0; b=100; n=120;
3. fx=j(n,2,1);
4. do x=1 to n;
5.      fx[x,1]=x;
6. end;
7. do x=1 to a;
```

```
8.   fx[x,2]=1;
9. end;
10. do x=a+1 to (a+b)/2;
11.     fx[x,2]=1-2*(((x-a)/(b-a))**2);
12. end;
13. do x=(a+b)/2 + 1 to b;
14.     fx[x,2]=2*(((x-b)/(b-a))**2);
15. end;
16. do x=b to n;
17.     fx[x,2]=0;
18. end;
19. print fx;
20. create data1 from fx[colname={'x' 'y'}];
21. append from fx;
22. quit;
23. symbol1 interpol=join width=5;
24. proc gplot data=data1;
25.     plot y*x;
26. run;
```

Code for Quality Assessment

This code is in Python.

```
1. import scipy
2. import numpy as np
3. import cv2
4. import matplotlib.pyplot as plt
5. from numpy import genfromtxt
6. imageTrue = cv2.imread('C:/Users/Renate/Desktop/Project/PleiadesCropped/
   New/NewSmall/roadsTrue3.png',0)
7. imageResult=cv2.imread('C:/Users/Renate/Desktop/Project/PleiadesCropped/
   New/NewSmall/Result.tif',0)
8. imarrayTrue=np.array(imageTrue)
9. imarrayResult=np.array(imageResult)
10. TP=0
```

```
11. TN=0
12. FP=0
13. FN=0
14. nrow=imarrayTrue.shape[0]
15. ncol=imarrayTrue.shape[1]
16. omission=[ ]
17. commission=[ ]
18. TPmatrix=[ ]
19. TrueRoad=0
20. rowcount=0
21. while rowcount<nrow:
22.     colcount=0
23.     omRow=[ ]
24.     comRow=[ ]
25.     TPRow=[ ]
26.     while colcount<ncol:
27.         if imarrayTrue[rowcount][colcount]==0:
28.             TrueRoad+=1
29.             if imarrayResult[rowcount][colcount]==0:
30.                 TP+=1
31.                 TPRow.append(0)
32.                 omRow.append(255)
33.                 comRow.append(255)
34.             if imarrayResult[rowcount][colcount]==255:
35.                FN+=1
36.                 comRow.append(255)
37.                 omRow.append(0)
38.                 TPRow.append(255)
39.        if imarrayTrue[rowcount][colcount]==255:
40.             if imarrayResult[rowcount][colcount]==255:
41.                 TN+=1
42.                 comRow.append(255)
43.                 omRow.append(255)
44.                 TPRow.append(255)
```

```
45.             if imarrayResult[rowcount][colcount]==0:
46.                 FP+=1
47.                 omRow.append(255)
48.                 comRow.append(0)
49.                 TPRow.append(255)
50.         colcount+=1
51.     rowcount+=1
52.     omission.append(omRow)
53.     commission.append(comRow)
54.     TPmatrix.append(TPRow)
55. completeness=TP/(TP+FN)
56. correctness=TP/(TP+TN)
57. quality=TP/(TP+FP+FN)
58. commissionR=(FP/TrueRoad)*100
59. omissionR=(FN/TrueRoad)*100
60. rowcount=0
61. omCount=0
62. comCount=0
63. while rowcount<nrow:
64.     colcount=0
65.     while colcount<ncol:
66.         if omission[rowcount][colcount]==0:
67.                 omCount+=1
68.         if commission[rowcount][colcount]==0:
69.                 comCount+=1
70.         colcount+=1
71.     rowcount+=1
72. scipy.misc.toimage(commission).save('C:/Users/Renate/Desktop/Project/
    PleiadesCropped/New/NewSmall/comm.tif')
73. scipy.misc.toimage(omission).save('C:/Users/Renate/Desktop/Project/
    PleiadesCropped/New/NewSmall/om.tif')
74. scipy.misc.toimage(TPmatrix).save('C:/Users/Renate/Desktop/Project/
    PleiadesCropped/New/NewSmall/TP.tif')
```

Trees to networks: an evaluation of neural random forests

Motlamedi Thupae 14400172

STK795 Research Report

Submitted in partial fulfilment of the degree BCom(Hons) Statistics

Supervisor: M.T. Loots

Department of Statistics, University of Pretoria

30 October 2017

Abstract

The essay seeks to investigate whether it is possible to restructure a collection of random forests as a collection of multilayered neural networks subject, to particular connection weights using the R statistical software [3]. To this end, this analysis seeks a random forest is reformulated into a neural network, leading to new hybrid procedures, namely neural random forests. Prior knowledge of the underlying design of regression trees is used as they have less parameters than standard networks that need adjusting, as well as exhibiting less restrictions on the decision boundaries. . Neural random forests consider the implications and advantages of both models and seeks to combine them in order to achieve a better performing model overall. The neural random forest uses the output of a random forest as the input for the neural network to essentially simulate a random forest (and its associated advantages) within a neural network. Neural random forests are reviewed by evaluating consistency results, numerical evidence, as well as assessments based on real data sets to gauge the method's performance against various prediction problems. It is then shown that, using RStudio and its associated packages, neural random forests cannot be formulated.

Declaration

I, *Motlamedi Thupae,* declare that this essay, submitted in partial fulfillment of the degree *BCom(Hons) Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

Motlamedi Thupae

Dr Theodor Loots

Date

Acknowledgements

The author would like to thank the MasterCard Foundation Scholarship Program for their financial support in the form of a postgraduate scholarship.

Contents

List of Figures

List of Tables

1 Introduction

Decision tree learning itself has been a key popular data-modelling technique whose use has spanned over fifty years in the fields of statistics and computer science. The approach has seen innumerable applications and by extension has influenced many modern predictive algorithms to date. This is owing to the simplicity and transparency of trees [3]. Regression trees can be traced back to earlier works on Categorical and Regression Trees introduced in 1984 [4]. Today, the work has been extended to random forests [6], applications of which contribute extensively to the most successful machine learning algorithms [3]. The performance of random forests has received keen interest since their inception. Numerous studies have been performed as a result. Most recently a study compared the performances of 179 different classifiers from 17 families on the complete set of the UCI collection of data sets. The study found that random forests indeed outperformed all other classifiers in a statistically significant way [8]. An integral part of this method the bagging principle: a sample of the data is taken. The sample is split into a training and testing set. (Generally, 80% towards the training and 20% the testing). Biau et al. (2016) extends the splitting into training. validation and testing subject to 50%, 25% and 25% splits respectively. A predictor is fitted on the training set, and the results thereof are pasted together. Thereafter the predictor is applied to the testing set to see if the results are comparable to the training set's. Random forests have been shown to work fast, exhibit substantial improvement over single tree learners and yield generalised error rates that often rank among the best [3]. "Forests have the flavour of deep network architectures owing to their ability to discriminate between a large number of regions" [2].

Neural networks are fashioned after their namesake: the brain's neurons. The tool works in a manner analogous to that of the human brain. The brain is made up of approximately 85 billion neurons. The branches of a neuron receive input signals from some stimulation or up-stream neurons. The signal is then processed in the cell body and transmits along a projection of the neuron to the output node. This output may be received by down-stream neurons or function organs such as muscles that deliver a reaction.

Neural networks have become the common name for multi-layer perceptrons. A perceptron is a linear binary classifier that partitions a space into parts using a linear function. This effectively separates classes using a straight line. For example two-features will have a decision boundary that is a straight line, input variables, the predictors and signal information. This information is then weighted according to its respective importance (the work done by the branches in neurons). The weighted signals are summed and processed by the activation function to deliver the output. The neuron in one layer is connected to all the neurons in the next layer, and since the information flows in one direction only this is also called a feed-forward network [14]. The power and advantage of neural networks lies in their ability to learn existing relationships directly from the data being modelled. Once the network has acquired this knowledge

through training, it can be applied to unknown data for the purpose of classification, prediction, time series analysis, etc. As a result neural networks do not require implementing appropriate algorithms in order to identify existing relationships. Alternatively, statistical analysis is reliant on assumptions regarding a model form, e.g, linearity in parameters or describing relationships between variables. Neural networks discover the the structure of the data through a training process. Once this is done, neural networks no longer need to learn from algorithms; they learn by example [14].

Neural networks have many parameters that render them easy to fit while remaining an excellent resource for complex modelling problems [3]. However, the aforementioned expressiveness bears the disadvantage of increased over-fitting risk, particularly on small data sets. Random forests on the other hand have fewer parameters but often perform inadequately for most data [3]. Extensive studies into the casting of forests into random forests have been performed in order to take advantage of both unique approaches while simultaneously bypassing their respective shortcomings [3]. As such, the neural random forest method is proposed.

This review seeks to investigate the neural random forests method as a tool that can aid in regression analysis. Section 2 explores the underlying theory of regression trees, random forests, neural networks and neural random forests that is split into subsections 2.1,2.2,2.3 and 2.4 respectively. Section 3 includes the results from the experiment and analysis of those results, with a final comparison in section 3.4 followed by the conclusion. Random forests will be restructured into a collection of neural networks that bear soft, non-linear and differentiable activation functions. They are thus trainable with a gradient-based optimisation algorithm and are expected to show better general performance [3].

The connection between forests and neural networks remains largely unexplored. The research report seeks to serve as an input into that exploration.

2 Background theory

2.1 Regression trees

Tree-based methods of classification were developed as part the CART program by Breiman et al (1984) [7]. A decision tree renders a form of classification for the purpose of prediction and/or regression. Regression trees differ from classification trees. In decision tree analysis X and Y are each observable- as they are discrete- and a tree is drawn up with $(x_1, y_1) ... (x_n, y_n)$as the working domain/ range for some $i = 1, ..., n$. In a node m, representing region R_m with N_m observations, let

$$\widehat{p}_{mk} = \frac{1}{N_m} \sum_{x, \in R_m} I(y_i = k),$$

be the proportion of class k observations in node m. The observation contained in node m is classified to the class in which it is the majority in node m owing to the condition $k(m) = argmax_k \widehat{p}_{mk}$. In the end the algorithm relies on a majority vote for classification. A certain leaf, that has been classified as having, for simplicity, a 0 or 1 will bear certain characteristics and subject to certain parameters (is x_{i1}bigger than 1 for example) will decide on what the data point should be classified as, given that it falls within that particular set of conditions. In this manner a binary tree is constructed to minimise the error in each training set.

A regression tree is a generalisation of the classification tree. However x_i and y_i are elements of real numbers. The data is continuous and not easily observed. The data consists of p inputs and N observations, and a response for each observation. Thus (x_i, y_i) for $i = 1, 2, ..., N$, with $x_i = (x_{i1}, x_{i2}, ..., x_{ip})$. The algorithm needs to automatically decide on the splitting variables and split points[10], as well as the shape of the tree. For this reason it is important to know where to split the data- as done on a simple two-dimensional plane- to begin the tree. From the initial split yes or no leaves are created and the data falls on either leaf. The aim is to minimise error such that: $\hat{y} = \sum(y - y_i)^2$ for some region R for the purpose of regression. $\hat{y}$is the average of the y_i's. The first split asks "is $x > 2$" and it creates two "yes or no" leaves: If no, then $x > 1$, which creates two more "yes or no" leaves. If no, then it is $x < 1$, and this region can be referred to as region $R1$. The average of the data points in $R1$ is used as a reference for all points that may fall within the region. This average is displayed underneath each leaf. For $x > 1$, referred to as region $R2$, the average is calculated as well. This process is followed until all "yes or no" questions are satisfied and all possible regions are classified. The "no" leaves are usually terminating and the "yes" leaves extend to other branches that split into different regions. For a partition into M regions $R_1, R_2, .., R_M$, model the response as constant c_m in each region:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m).$$

Maintaining the objective of minimising the sum of squares $\sum(y_i - f(x_i))^2$, the best response is then the average of y_i in region R_m, as seen by:

$$\widehat{c}_m = ave(y_i | x_i \in R_m).$$

To find the best binary partition in terms of minimum sum of squares is considered computationally uattainable [10], and as such necessitates a greedy algorithm: considering the set of all the data, set some splitting variable j and an associated split point s, and define

$$R_1(j, s) = \{X | X_j \leq s\},$$

and

$$R_2(j,s) = \{X|X > s\}\,.$$

After this definition, seek values for j and s that render a solution to

$$\min_{j,s}\left[\min_{c1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_1(j,s)}(y_i-c_2)^2\right].$$

And for any choice j and s, the inner minimisation is solved by $\widehat{c}_1 = ave(y_i|x_i \in R_1(j,s))$ and $\widehat{c}_2 = ave(y_i|x_i \in R_2(j,s))$. After finding the best split, distribute the data into the two regions. This process is then repeated on all the resulting regions, establishing the classifier [10]. The above is to grow a single tree. This is not enough owing to the caveat that trees themselves have the disadvantage of being unstable: they are characterised as having high variances due to the hierarchical nature of the process. An error in the top split essentially 'trickles down' to the resultant splits below it [10]. The problem can be alleviated by bagging: averaging the collection trees to reduce the variance.

The abbreviation 'bagging' refers to the method of bootstrap aggregation. Bootstrapping is a method that is widely used to improve the statistical accuracy of a model. It has been identified as being good for estimating extra-sample prediction error [10]. The rationale is as follows: consider a model that is fitted to a set of training data $Z = (z_1, z_2, ..., z_N)$ where $z_i = (x_i, y_i)$. At random, draw sample data sets from Z while ensuring that each sample size is the same as the original training sample's size. Sampling is done with replacement. Perform the process B times in order to obtain B bootstrap samples. Following this, refit the model to each bootstrap data set and evaluate the fit over the B replications [10]. Figure 1 depicts a prediction of $S(Z)$ resulting from bootstrapping on Z. The above described model is fitted and the resulting values $S(Z^1), ..., S(Z^B)$ are then used to assess the statistical accuracy of Z. Bootstrap sampling makes it possible to estimate any parameter of the distribution of the quantity, i.e. the variance,

$$\hat{var}[S(Z)] = \frac{1}{B-1}\sum_{b=1}^{B}\left(S(Z^b)-\overline{S}\right)^2$$

where $\overline{S} = \sum_b \bar{S}(Z^b)/B$, is the sample mean. To estimate the prediction error, fit the model on a set of bootstrap samples and monitor how well it predicts the original training set [10].

If $\hat{f^b}(x_i)$ is the predicted value at x_i, from the model fitted to the both bootstrap data set, the estimate is,

$$\hat{Err}_{boot} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N}L(y_i,f^b(x_i)).$$

It should be noted that Hastie et al. highlight a drawback of bootstrapping: the method does not provide a good estimate in general. This is owing to the fact that the bootstrap data sets are acting as

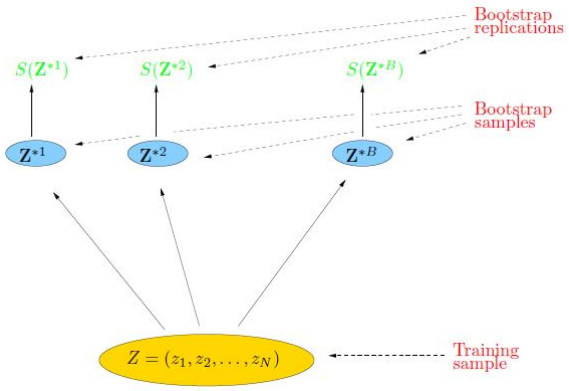


Figure 1: A schematic representation of the underlying intuition of bootstrapping[10].

the training samples whereas the original training set is acting as the test sample. This delivers a key problem in that the two observations have samples in common. This overlap can result in over-fitting the predictions, rendering them exceptionally good, yet ultimately unrealistic [10]. In lieu of this problem, Leo Breiman created bootstrap aggregation- otherwise known as bagging. The advantages of bagging are that it reduces variance and improves model accuracy.

Bagging greatly aids in improving the bootstraps prediction. It exploits the connection between bootstrapping and the Bayesian approach to inference in that the bootstrap mean is approximately a posterior average [10] [5]. Consider a regression problem: the model described under bootstrapping above is fitted to estimate $\hat{f}(x)$ at input x. The application of bagging averages this prediction across the bootstrap samples B. This in turn reduces its variance [10]. For each $b \in B$ and $b = 1, 2, ..., B$, fit the model rendering prediction $\hat{f}^b(x)$. The bagging estimate is defined by

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^b(x).$$

The empirical distribution putting equal probability $\frac{1}{N}$ on each of the data points (x_i, y_i) is denoted by $\widehat{\rho}$. The true bagging estimate is thus defined by $E_{\widehat{\rho}}\widehat{f}^*(x)$ where $Z^* = \{(x_1^*, y_1^*), ..., (x_N^*, y_N^*)\}$ and each $(x_i^*, y_i^*) \sim \widehat{\rho}$. $\hat{f}_{bag}(x)$ is a Monte Carlo estimate which is approached as the true bagging estimate, approaching it as $B \to \infty$[10]. Through this, bagging therefore allows for the averaging of many 'noisy but unbiased' models which subsequently reduces the variance [10].

Applied to regression trees, $\hat{f}(x)$ denotes the tree's prediction at input vector x with each bootstrap tree bearing differing features to the original, and may exhibit a varying number of terminal nodes [10]. The bagged estimate is the average prediction at x from B trees.

2.2 Random forests

Section 2.1 highlights the process for growing a regression tree. The process can be replicated to create a regression forest. The random forest method is a modification of bagging that builds a large connection of decorrelated trees, and then averages them [11]. Trees are generally noisy [11], and averaging greatly enhances the model. Each tree attained through bagging is identically distributed and as such the expected value of B trees is the same as the expected value of any other tree. An average of B $i.i.d$ random variables, each with variance σ^2, has variance $\frac{1}{B}\sigma^2$ [11]. During the tree growing process, and before each split , on the bootstrapped data set, randomly select $m \leq p$ of the input variables before splitting and then commence to grow the trees. Hastie et al puts the typical value for m at $\sqrt{p}$ 'or even as low as 1' [11]. Upon growing the B trees, the random forest predictor is defined as:

$$\widehat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x;\Theta_b).$$

Θ_b characterises the b^{th} random forest tree in terms of split variables [11], cut-points at each node as well as the terminal-node values. Through the reduction of m, the correlation between any pair of trees is reduced and by extension, the variance, and the variance of the average.

2.3 Neural networks

Neural networks have become the common name for multi-layer perceptrons. A perceptron is a linear binary classifier that partitions space into parts using a linear function. This effectively separates classes using a straight line. For example two-features will have a decision boundary that is a straight line, input variables, the predictors and signal information. This information is then weighted according to its respective importance (the work done by the branches in neurons). The weighted signals are summed and processed by the activation function to deliver the output. The neuron in one layer is connected to all the neurons in the next layer, and since the information flows in one direction only this is also called a feed-forward network [14].

The procedure can be expressed mathematically as:

$$y(x) = \phi(\Sigma_{i=1}^{m} w_i * x_i)$$

, where y is the output signal, $\Phi()$ is the activation function, x_i refers to the $x_1, ..., x_m$ input variables and w_i is weight assigned to each input variable. Information is received from the raw data contained in the feature variables and the activation function. This is combined with information from the input nodes, rendering the output. This model can be compared to the regression model. Each input variable

is in analogue to the predictors of a regression model with weight being the coefficient of each predictor. All input nodes constitute a single layer: a network containing only input and output nodes is a termed a single-layer, the simplest form of a neural network.

For example, simple linear regression problem: given the training data $X = \{x\}$, and the corresponding output $Y = \{y\}$, the aim is to set Y in a linear fashion such that $\mu_k \approx w_0 + w_1 * x_1$. Subscript k is the index of the data point in the training set (the neuron in question) and the subscript i refers to coordinate of the data point (the input neuron to which the weight refers). The activation function serves as a limiter of the permissible amplitude range of the output signal to some finite value. It is typically the range of the output of a neuron is written in either of the following forms [0,1] or [-1,1]. The intercept/ bias term b_k is included, whose effect is to increase or lower the net input of the activation function depending on whether it is negative or positive.. This formula can be extended to include multiple attributes: x can then be defined as $x = [x_0, ..., x_m]$ followed by $w = [w_{k1}, ..., w_{km}]$ resulting in a linear combination function of the form $\mu_k = w_{01} * x_0 + ... + w_{km} * x_m = \sum_{i=0}^{m} w_{ik} * x_i$. The Figure 2 depicts the intuitive process of neuron k:

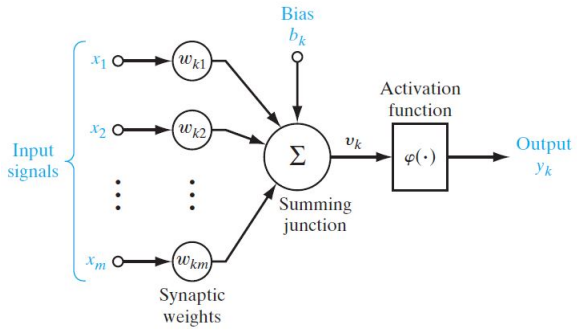


Figure 2: Intuitive process of neuron k [14]

From $\mu_k = \sum_{i=0}^{m} w_{ik} * x_i$, another condition that accommodates for the activation function and bias needs to be added that completes the form for neuron k: $y_k = \varphi(\mu_k + b_k)$. $\varphi(\bullet)$ refers to the activation function. Weolowski et al. highlights certain steps to take when configure a suitable neural network: the number of layers (input, hidden and output) must be established. Secondly, decide on how many nodes must be in the hidden layer. Additionally it is essential to choose the correct activation function, error criterion and learning algorithm. The activation function in the hidden and output layers is symmetric sigmoid. A neuron with an activation function that is permanently set to 1 is added (for an MLP). This neuron is called a bias: it connects to the neurons through a weight (referred to as the threshold). The threshold's purpose is to determine whether specific conditions are fulfilled in order to correctly in order to correctly assess the results obtained [14].

The correct set of weights is not known beforehand and as such a random variable is assigned to them. To properly update these weights the Back-propagation algorithm (BP) is added. This process is called training or teaching the model. The BP can be termed a generalisation of the least squares algorithm that adjusts these weights to minimise the mean squared error (MSE) between the target value and actual output. MSE is defined as:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{n}(out_{ik} - y_k)^2,$$

where m is the total number of training cases, n is the number of network outputs, out_{ik} is the output for the i^{th} neuron and k^{th} network output. This is a supervised process as the output value is compared to the target. It follows from above that to obtain the MSE it is essentially an aim to minimise the Sum of the Squared Errors (SSE) as is mostly done in regression models and by extension regression forests.

2.4 Neural random forests

This section follows the development of the neural random forest model as posited by Biau et al. The underlying structure is that of non-parametric regression estimation. The random input vector $X\epsilon\,[0,1]^d$is observed. The aim: predict the dependent variable $Y\ \epsilon\mathbb{R}$ through estimation by the regression function $r(x) = E[Y|X = x]$ [3]. Given this, assume a training sample $D_n = ((X_1, Y_1), ..., (X_n, Y_n))$, $n \geq 2$ of independent random variables distributed identically to (X, Y). The data in D_n is used to construct an estimate of the function r. Additionally, set the condition that the regression function estimate r_n is mean square error consistent if $E[r_n(X) - r(X)]^2 \to 0$ as $n \to \infty$. Upon establishing the above, it is possible to continue from a tree to a neural network.

Biau et al. (2016) define a regression function as an estimate that uses a 'hierarchical segmentation of the input space' [3]. The observations within the space are ranked and separated in some order of importance and applied to the estimate. Each tree node corresponds to one of the segmentation subsets in $[0,1]^d$. As such, the type of trees developed are considered ordinary binary regression trees: A node has exactly either zero or two leaves. The former renders the node terminal. If a node u represents the set $A \subseteq [0,1]^d$ and its leaves, designated $u_L$and u_R for 'left' and 'right' respectively, represent $A_L \subseteq [0,1]^d$ and $A_R \subseteq [0,1]^d$ then it is required that $A = A_L \cup A_R$ and that $A_L \cap A_R = \emptyset$. Thus $A_L$and A_R are mutually exclusive. The root is a representation of the entire space $[0,1]^d$and the leaves , taken together, form a division of $[0,1[^d$. Within an ordinary tree, to pass from A to either A_L or A_R occurs by answering: on $x = (x^{(1)}, ..., x^{(d)})$ "is $x^{(j)} \geq \alpha$" , for some dimension $j\epsilon\{1, ..., d\}$ and some $\alpha\epsilon[0,1]$.

In prediction the input is first passed into the tree root node and is then iteratively transmitted to the 'child' node that belongs to the region in which the input is located. Repeat the process until a leaf node is realised. If a leaf represents the region A, the natural regression function estimate takes the form

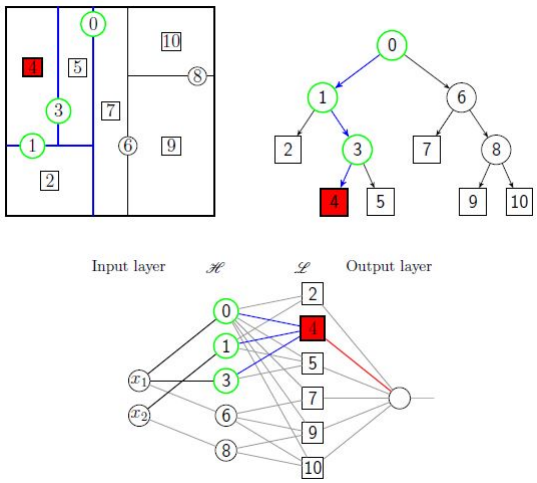


Figure 3: Process from regression tree to neural network: the regression is fitted on the training data. From there, the split directions and positions are extracted and these are used to fit the neural network. Applied to the random forest, 'M CART-type trees are converted into M tree-type neural networks. [3]

$t_n = \frac{\left(\sum_{i=1}^{n} Y_i \| X_i \epsilon A\right)}{N_n(A)}$, $x \epsilon A$, where $N_n(A)$ is the number of observations in cell A. The prediction of point x in leaf node A is the average of the outputs from all of the sample points belonging to that particular region, region A. Figure 3 depicts a two-dimensional example.

Assume at hand a regression tree $t_n$whose construction depends on data D_n and takes on values on each of the $K \geq 2$ terminal nodes: interpret this estimate as a three-layer neural network estimate comprised of two hidden layers as well as a single output layer. Let $\mathcal{H} = \{H_1, ..., H_{k-1}\}$ be the collection of all hyper-planes participating in the construction of t_n. It is important to note that $H_k \epsilon \mathcal{H}$ is of the form $H_k = \left\{x \epsilon [0,1]^d : h_k(x) = 0\right\}$, where $h_k(x) = x^{(jk)} - \alpha_{(jk)}$ for some, eventually data dependent, jk $\epsilon (1, ..., d)$ and $\alpha_{jk} \epsilon [0,1]$. To each leaf of the query point x, for each query hyper-plane H_k, it is possible to find the side on which x falls ($+1$ codes for right; -1 codes for left). With this notation, the tree estimate t_n is then considered identical to the neural network described below [3]. This formulates a neural network analogous to a random forest.

First hidden layer: the first hidden layer of neurons corresponds to $k-1$ perceptrons- one for each inner tree node, whose activation is defined as

$$\tau\left(h_k(x)\right) = \tau\left(x^{(jk)} - \alpha_{jk}\right),$$

where $\tau(u) = 2\|_{u \geq 0} - 1$, is a threshold activation function and the weight vector is single one-hot vector of feature jk with $-\alpha_{jk}$ as the bias value. Associated with each split in the tree is a neuron in the first layer that assigns the relative position of x in reference to the split. Ultimately, this layer renders the ± 1 vector $(\tau(h_1(x)), ..., \tau(h_{k-1}(x)))$, which describes all decisions of the inner tree nodes [3].

This includes those nodes not on the path of x. Intuitively, $\tau(h_1(x))$ is $+1$ if x is on one side of H_k; -1 if on the other side. By convention: $+1$ if $x \epsilon H_k$ [3]. It is imperative to note that each neuron k is connected to one-and-only-one input $x^{(jk)}$. This connection has a weight 1 and offset $-\alpha_{jk}$. Pursuant to the activations in the first layer, the precise terminal cell of x is then reconstructed in the second layer.

Second hidden layer: the first layer outputs a $k-1$-dimensional vector of $\tau\left(h_k(x)\right)$, the ± 1-bits. This vector gives the location of x in the leaves of the tree. It is now possible to extract the leaf node identity of x through a weighted combination of $\tau\left(h_k(x)\right)$, in conjunction with an appropriate thresholding [3]. Let $L = \{L_1, ..., L_k\}$ be the collection of all tree leaves, and let $L(x)$ be the leaf containing x. This second hidden layer possesses K neurons, one for each leaf, and assigns a terminal cell to x. To achieve this, connect a unit k from the first layer to a unit k' from the second layer if and only if H_k in the sequence of splits forming a path from the root to the leaf $L_{k'}$. This connection has a weight $+1$ if, in that path, the split by H_k stems from a node to a right child, otherwise -1. It follows then that if, $(u_1(x), ..., u_{k-1}(x))$ is the vector of ± 1 bits from the first layer's output, the output $v_{k'}(x) \epsilon \{-1, 1\}$ of neuron $k^{'}$ is $\tau\left(\sum_{k \to k'} b_{k,k'} u_k(x) + b^0_{k'}\right)$, where $k \to k'$ means that k is connected to k' and $b_{k,k'} = \pm 1$ is the corresponding weight. The offset $b^0_{k'}$ is set to

$$b^0_{k'} = -l(k') + \frac{1}{2} \tag{1}$$

where $l(k')$ is the length of the path from the root to $L_{k'}$ [3]. Biau et al. elaborates that to understand 2.4 , observe that there are exactly $l(k')$ connections starting from from the first layer and pointing to k', and that

$$\left\{ \begin{array}{ll} \sum_{k \to k'} b_{k,k'} u_k(x) - l(k') + \frac{1}{2} = \frac{1}{2} & , x \in L_{k'} \\ \displaystyle\sum_{k \to k'} b_{k,k'} u_k(x) - l(k') + \frac{1}{2} \leq -\frac{1}{2} & , otherwise \end{array} \right|$$

thus with the choice (1), the argument of the threshold function is $\frac{1}{2}$ if $x \epsilon L_{k'}$ and smaller than $-\frac{1}{2}$ otherwise. Hence $v_{k'}(x) = 1$ if and only if it is the terminal cell of $L_{k'}$.

Output layer: Let $((v_1(x), ..., v_k(x))$ be the output of the second hidden layer. If $v_{k'}(x) = 1$, then the output layer computes the average $\bar{Y_{k'}}$ of the Y_i corresponding to X_i falling in $L_{k'}$. Equivalently,

$$l(n) = \sum_{k'=1}^{K} w_{k'} v_{k'(x)} + b_{out}$$

where $w_{k'} = \frac{1}{2}\bar{Y_{k'}}$ for all $k' \epsilon \{1, ..., K\}$ and $b_{out} = \frac{1}{2}\sum_{k'=1}^{K} \bar{Y_{k'}}$.

3 Application

The analysis follows the same procedures, and uses some of the data sets, used by Biau et al.(2016). This is in order to accomplish two goals: to find similar, if not replicate, their results as well as gauge the practical attainability of neural random forests proposed in [3]. The data sets considered for analysis are from the UCI Machine Learning Repository [1]. Both the initial predictors, namely the random forest and neural network, are trained on the data sets. The data sets themselves are generally considered small scale data. This is in line with the objective to evaluate the models' performances on such types of data sets. Trees, and by extension random forests, are considered simple and transparent, with extensive applications to machine learning in explaining complex data sets. Additional studies have found that compared to other classifiers, random forests boast the best performance and have been identified as one machine learning algorithm that is capable of handling large-scale high-dimensional data sets [3]. Neural networks are characterised by many parameters that make them an adaptable and rich instrument for complex data modelling. This is however disadvantaged by an increased over fitting risk [3]. This risk is contrasted by the random forest's robustness to over fitting. Biau et al. (2016) posit a combination of these two methods in order to develop hybrid methods that exploit their advantages while removing their respective disadvantages. Ultimately prior knowledge from regression trees is used to initialise neural random forests. To fit the neural random forest, the output from the random forest - namely, the split directions and split positions, are extracted and used as the input to fit a neural network. It should be noted that this approaching at combining random forests and neural networks has been critiqued: the treatment of a random forest as an input is seen as hindering the performance of the neural model in addition to being hard to learn[13]. However, it should be noted that those findings are from a study that focused on classification instead of regression. The investigation into the merits of the claim fall outside the scope of this analysis.

The evaluation criteria for performance is based on the model's root mean squared error (RMSE) as done in [3]. The statistical software used is R [1], particularly the RStudio console. This software has the advantage of access to several machine learning packages that can be used to model predictors as well as summarise results in both graphical and tabular form from within. The packages chosen are randomForest [12] and neuralnet [9]. These were chosen for both their simplicity and ability to be tuned in the required manner. Biau et al.(2016) uses the sickit-learn implementation to learn the random forest and the neural network is trained using the tensorflow framework. The above mentioned packages work similarlyr.

The data sets chosen are summarised in table 1. These are considered small-scale data sets and have been chosen to investigate the performance of random forests and neural networks on such data.

[1] https://cran.r-project.org/doc/FAQ/R-FAQ.html

Additionally, their nature aids in the practical implications of attempting to initialise the neural random forest methods. Minor adjustments were made to the data sets: non-numerical input features and variables with missing values were removed from the samples. Random within sampling was performed, with replacement, on the data set subject to a 50:25:25 ratio split into training, validation and testing sets respectively.

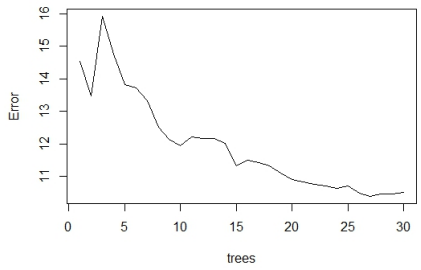
Name	Number of samples	Number of features
Auto MPG	398	7
Housing	506	13
Forest fires	517	10
Concrete	1030	8

Table 1: Summary of data sets and their attributes after adjustments [1]

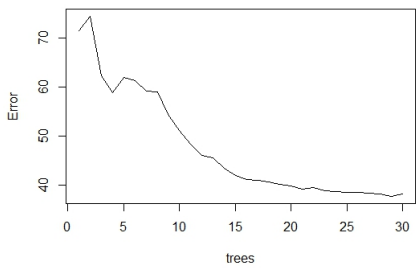
3.1 Random forest

The random forest is trained with 30 trees subject to a maximum depth perception of 6 [3]. This translates to a maximum of 6 tree nodes set into the randomForest function. The number of variables randomly sampled as candidates at each split is also set to 6. Upon training the forest, a prediction is performed using those results on the validation and testing samples. The mean squared error, hereafter referred to as error, values associated with the number of trees from these predictions were plotted. Overall the error values decrease as more trees are grown across all data sets. The results are graphically presented in figure 4. Figures 4 a, b and d exhibit consistently decreasing error rates as the number of trees in the forest grow to 30. Whereas figure 4 d experiences a slight increase when there are between 5 and 10 trees, it is neither sharp nor uncontrolled. Figure 4 a shows a sharp increase to its highest error at 4 trees and then continues to have less sharp and better controlled increases while exhibiting a gradual decrease as the forest grows. Figure 4 c shows the least controlled behaviour: there is a significantly sharp decline (the lowest witnessed) when there are between 3 and 4 trees, with a subsequent and significantly sharp increase when there are 5 trees. From then it proceeds to exhibit similar behaviour to the other trained models. Figure 4 d exhibits the best trained error rate performance, as the plot shows low error values throughout as the forest is grown.

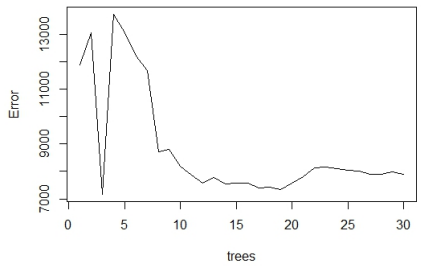
In order to determine the overall performances of each model and its applicable prediction, the errors from all trees are averaged. The square root of that value is calculated to obtain the RMSE. The RMSE values of each prediction are summarised in table 2. The RMSE values from training, validation and testing for all four data sets are shown. Across all data sets there is an increase in RMSE values from training, validation to testing. This differs from the behaviour of the values found in [3]. Only the forest fires RMSE decreases from validation to sampling. The differences however, are not particularly large. This does not mean the claim of robustness to over fitting should be discarded. The model results are



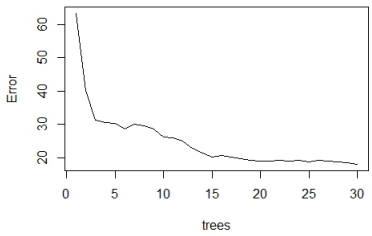
a) Auto MPG



b) Concrete compressive strength



c) Area burned by forest fires



d) Boston housing median value

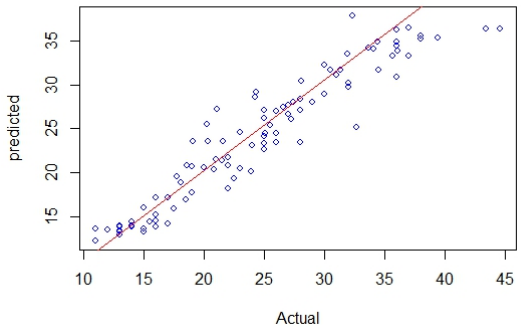
Figure 4: Error rate performances of trained random forest as the number of trees increased to 30.

consistent with the claim of robustness when considering these slight increases in RMSE values. In terms of the RMSE values, the prediction of area burned by forest fires is the best performing random forest.

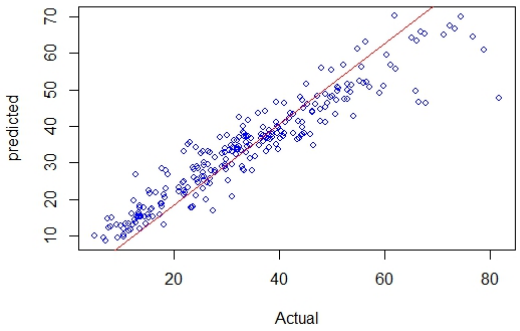
Data set	Training	Validation	Testing
Auto MPG	1.4435	2.3336	2.5493
Concrete	2.8552	5.6986	5.7001
Forest fires	45.5609	49.9067	29.2824
Housing median value	1.8682	2.831	3.2081

Table 2: Summary of RMSE and standard deviations for the random forest models

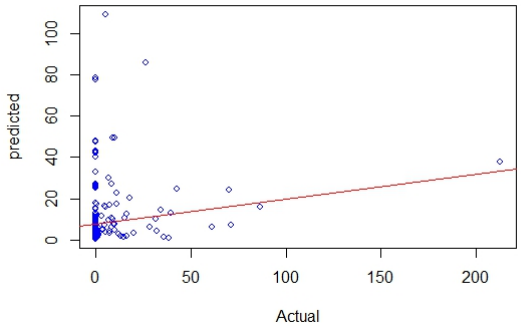
Figure 5 contains the actual- versus predicted values plots for all data sets. Each prediction was fitted to a linear regression in order to gauge the fit between actual sample values and predicted sample values. All show a positive linear relationship between actual observations and the predicted values. Figures 5 a, b and d show that the models delivered a good prediction: the values are clustered around the fitted linear regression line. Figure 5 a shows a wider spread however, whereas b and d show consistent clustering with only a few outliers. The clustering about the fitted line establishes confidence in the prediction being a good fit: the actual- and predicted values are close to each other. Figure 5 shows interesting behaviour: The values are congregated at very low values with more outliers than the other models. However, within the original data set, the observed values were mostly small- many equal 0 with few being larger than 20. There are, overall, large differences among the values. Inferences regarding the goodness of fit is hard: the scattered nature of the values leads to the conclusion that this is not a good prediction.



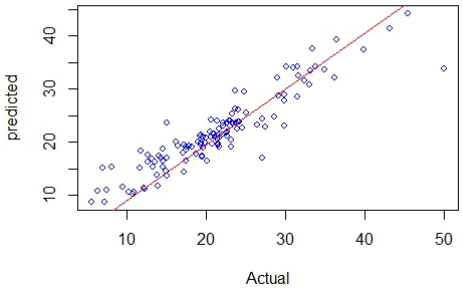
a) Auto MPG



b) Concrete compressive strength



c) Area burned by forest fires



d) Boston housing median value

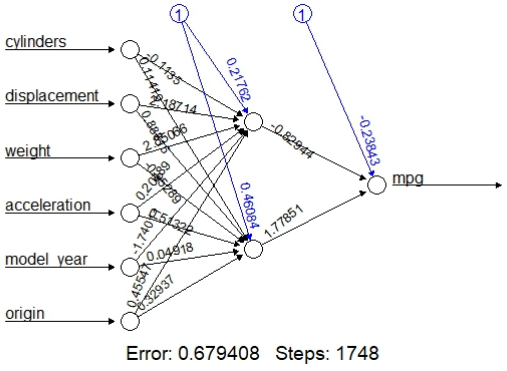
Figure 5: Fit plots for testing prediction accuracy of the random forest models.

3.2 Neural network

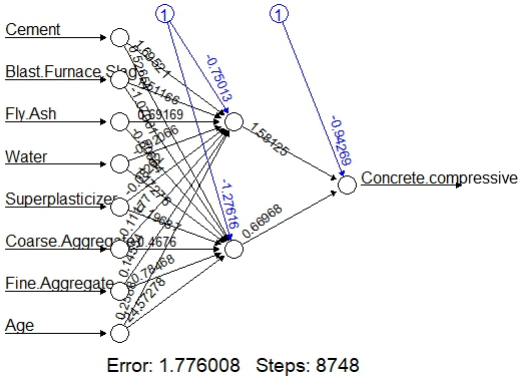
The neural networks were trained in accordance with the 50:25:25 ratio and configured to have three layers: two hidden and one output. The experiment was repeated 10 times. Figure 6 shows the trained neural networks for all data sets. The neurons in the first hidden layer correspond with the number of input variables in each data set, the 6 neurons in figure 6 a corresspond with 6 input variables in the Auto MPG data set. This allows the user to identify what each input contributes towards determining the outcome. The structure follows the instructions in [3]. The performances of these neural networks are analysed with the aid of table 3 and the fitted plots in figure 7. It should be noted that in order to fit the neural network, the data is scaled- or rather normalised. This is an important step in order for the package to work. However, no meaningful interpretation is possible if the the results are not de-scaled. As such the values in figure 6 are normalised. This is how neuralnet[9] outputs results. The prediction results are descaled using a parameter. For example, for Auto MPG data, the values in the dependent variable (mpg) were used to calculate the value of the descaling paramter:

$$D_n = (max(mpg\$mpg) - min(mpg\$mpg)) + min(mpg\$mpg) \tag{2}$$

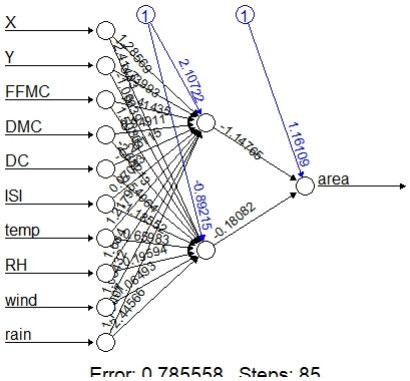
The value for equation 2 is multiplied with the predicted values to enable a meaningful interpretation of the results. The complete descaling procedure is shown in the R code included in the Appendix.



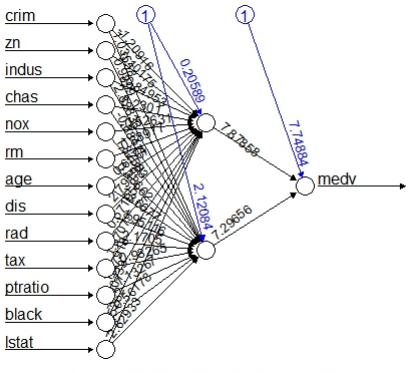
a) Auto MPG



b) Concrete compressive strength



c) Area burned by forest fires



d) Boston housing median value

Figure 6: Trained neural networks.

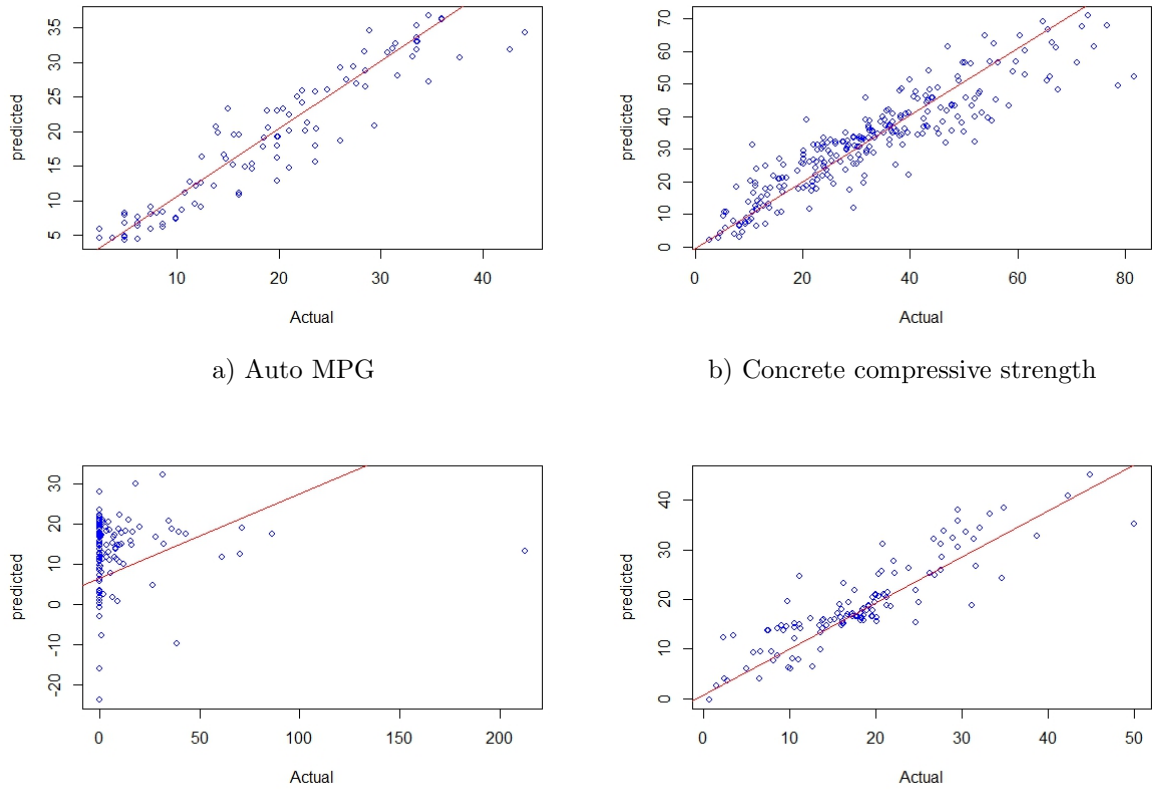
Table 3 contains the RMSE values from the neural network predictions. Performances of the models are based on the differences in the RMSE values between the training, validation and testing predictions. The Auto MPG and forest fires predictions experience overall decreasing RMSE values between training and testing. The Auto MPG RMSE values increase from validarion to testing and a decrease for forest fires. The majority of the models experience an increase in RMSE values from validation to testing, calling into question the model's robustness to over-fitting. Boston housing median value RMSE values increase throughout. The area burned by forest fires prediction maintains the highest RMSE values. However, there is a substantial decrease in these values from training to validation to testing: this is indicative of a well performing model and leads to the inference that the model is learning as it is trained. This is not enough to gauge predictive performance: goodness fit established through an 'actual- versus predicted values' scatter plot as seen in figure 7.

The plots in figure 7 indicate that there is a difference between the actual- and predicted values. Figure 7 a and d show more scattering in the points and more outliers as most of the points stray from the fitted regression line. Figure 7 b shows a better fit as the points seem to congregate about the

Data set	Training	Validation	Testing
Auto MPG	3.7485	3.1696	3.5237
Concrete	6.8007	7.2664	7.1977
Forest fires	82.1534	35.2315	25.2297
Housing median value	3.7105	3.7441	4.2952

Table 3: Summary of RMSE and standard deviations for the neural network results.

fitted line. But, there are many outlying points as well. The points in figure 7 7 are mainly clustered around themselves and not around the fitted line. Many of the values remain small but there is over- and underestimation in the prediction in addition to significant outliers. This is indicative of a very poor performance. Overall the models show adequate performance, that can be improved.



a) Auto MPG

b) Concrete compressive strength

c) Area burned by forest fires

d) Boston housing median value

Figure 7: Fit plot of results from neural network.

3.3 Neural random forest

In order to formulate the neural random forest, Biau et al. (2016) sets out a procedure to follow. First a random forest is grown on the training data. From there isolate and extract the split directions and split positions. The splits will serve as the input for the neural network in order to convert 'M CART-type trees into M tree-type neural networks. However, the splits proved to be unattainable: the randomForest

, or any other, package did not render an output that enables one to isolate and extract the splits. As such, the neural random forest method could not be evaluated at this time. Neural random forests using R statistical software and its available packages remains unattainable.

3.4 Comparison of results

In lieu of the results of section 3.3 the comparison is limited to random forests and neural networks. Their respective performances are measured against each other.

The random forest predictions have lower RMSE values than the neural network.For all data sets except for forest fires, in terms of RMSE, the random forest outperforms the neural network. Additionally the actual- versus prediction plots follow the same behaviour: the random forest points are better fitted with less outliers across the board. The neural network plots exhibit some adequate performance, however it requires improvement and as such the neural network is not as good a predictor as the random forest.

4 Conclusion

The aim of this report was to evaluate the feasibility and results of the neural random forest method of Biau et al. (2016) [3]. Inherent in the evaluation is the assessment of two models: random forests and neural networks. Random forests have been identified as one of the best performing ensemble methods that delivers results that are consistently robust to over-fitting. Additionally, they have been identified as an important machine learning algorithm that is capable of handling large-scale high-dimensional data sets [3]. Neural networks have many parameters that render them an adaptable and rich tool for complex data modelling , but bear the burden of increased over-fitting risk. Neural random forests seek to exploit each model's unique advantages to establish a unique hybrid method. In the course of the analysis, random forests prove that they do indeed outperform neural networks on small scale data. Random forests maintain consistently lower RMSE values in addition to delivering better fitting predictions. Where random forests lacked in performance the neural network did not offer much improvement. It is recommended that the analysis of the forest fires data set be kept for classification: the area burned should be split into categories such as 'none, small, medium and large' for example. This will help in understanding the factors that influence the relative sizes of the areas burned by forest fires.

From the application, it is clear that these configurations and in lieu of the empirical results, neural random forests are not easily attainable in R [**?**]. An alternative approach is available that addresses the shortcomings of neural random forests in that the treatment of a random forest as an input is seen as hindering the performance of the neural model in addition to being hard to learn[13]. Wang et al.(2017) offer an alternative that warrants investigation. The method proposed in [13] outperforms the other classifiers in its exploitation of its random forest tree like structure as well as the neural network's ability

to linearly combine features followed by a non-linear function. In light of this, it has been confirmed that random forests outperform neural networks and a proper combination of the two models could result in better performance overall.

References

[1] A Asuncion and DJ Newman. Uci machine learning repository. irvine, ca: University of california, school of information and computer science. *URL [http://www. ics. uci. edu/˜ mlearn/MLRepository. html]*, 2007.

[2] Yoshua Bengio. Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[3] Gérard Biau, Erwan Scornet, and Johannes Welbl. Neural random forests. *arXiv preprint arXiv:1604.07143*, 2016.

[4] Leo Breiman. Neural networks: A review from statistical perspective: Comment. *Statistical Science*, 9(1):38–42, 02 1994.

[5] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, Aug 1996.

[6] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[7] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[8] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems. *Journal of Machine Learning Research*, 15(1):3133–3181, 2014.

[9] Stefan Fritsch and Frauke Guenther. Package âneuralnetâ. 2016.

[10] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Model Assessment and Selection*, pages 219–259. Springer New York, New York, NY, 2009.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Random Forests*, pages 587–604. Springer New York, New York, NY, 2009.

[12] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[13] Suhang Wang, Charu Aggarwal, and Huan Liu. Using a random forest to inspire a neural network and improving on it. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 1–9. SIAM, 2017.

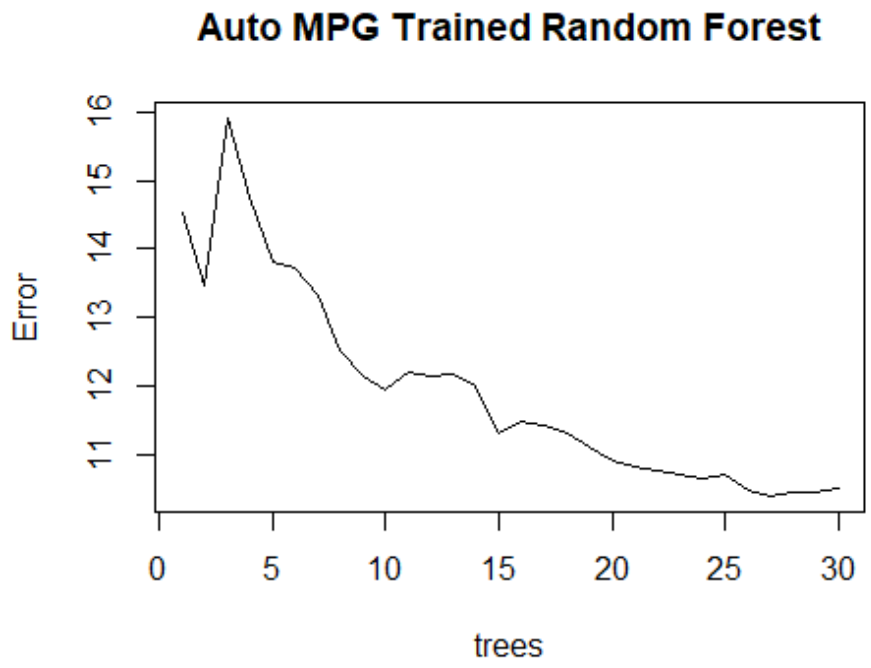
[14] Marek Wesolowski and Bogdan Suchacz. Artiicial neural networks: Theoretical background and pharmaceutical applications: A review. *Journal of AOAC International*, 95(3):652 – 668, 2012.

5 Appendix

Random forest: code and output

```
#import datatset mpg into r
mpg=read.csv('C:/Users/USER/Google Drive/2017/School/STK
795/Research/Application research/mpg.csv')
mpg= subset(mpg,select = -c(horsepower, name)) #remove columns with missing
values and characters that do not add to reg
#str(mpg) #horsepower and name columns removed


set.seed(123) #apply random sampliing to split samples into 3: train, val and
test via Biau instruction
samples = sample(seq(1, 3), size = nrow(mpg), replace = TRUE, prob = c(.5,
.25, .25))
train = mpg[samples == 1,] #training sample
test = mpg[samples == 2,] #testing sample
val = mpg[samples == 3,] #validation sample

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.3.3

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

#create a formula that lists the variables of interest. This code applies in
neural networks as well
#helps to create own formulas when they are not inherent in the function
vars=colnames(mpg)
predictvars=vars[!vars%in%"mpg"]
predictvars=paste(predictvars,collapse = "+")
form=as.formula(paste("mpg~",predictvars,collapse = "+"))
predictvars

## [1] "cylinders+displacement+weight+acceleration+model_year+origin"

form

## mpg ~ cylinders + displacement + weight + acceleration + model_year +
##     origin

RandomForest=randomForest(form,ntree=30, data=train,  nodesize=6, mtry=6,
replace=TRUE, mse=TRUE , importance=TRUE)
RandomForest

##
## Call:
##  randomForest(formula = form, data = train, ntree = 30, nodesize = 6,
```

```
mtry = 6, replace = TRUE, mse = TRUE, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 30
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 10.49601
##                     % Var explained: 83.07

plot(RandomForest, main = 'Auto MPG Trained Random Forest')
```

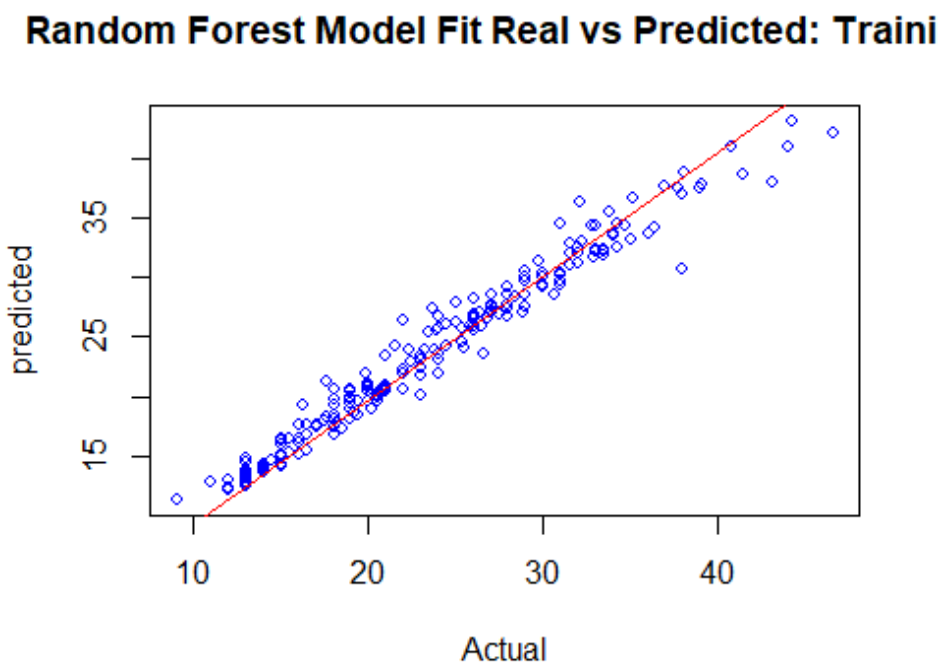



```
#Evaluation of results via validation and testing. MSEs are calculated below
RFTrainPredict=predict(RandomForest,train)
RFValPredict=predict(RandomForest,val)
RFTestPredict=predict(RandomForest,test)
RFTrainPredict

##        1        3        6       10       12       15       17       18
## 17.47333 16.82611 14.35528 14.11889 14.27667 23.04444 19.72256 20.77700
##       19       29       30       35       36       38       39       40
## 27.10500 11.38333 27.10500 17.60883 17.40917 18.41356 14.11056 13.83917
##       41       42       43       44       45       46       47       48
## 13.94500 13.79333 12.29778 12.92111 12.57222 18.20000 21.95667 18.18667
##       49       51       52       54       56       57       60       62
## 18.21444 26.69000 29.94889 30.20472 27.63500 26.71806 23.12156 23.43778
##       63       64       66       70       74       75       76       77
## 13.76056 14.03472 13.84778 13.05222 13.25556 13.38333 13.91278 19.86611
##       79       80       81       83       85       86       90       91
```

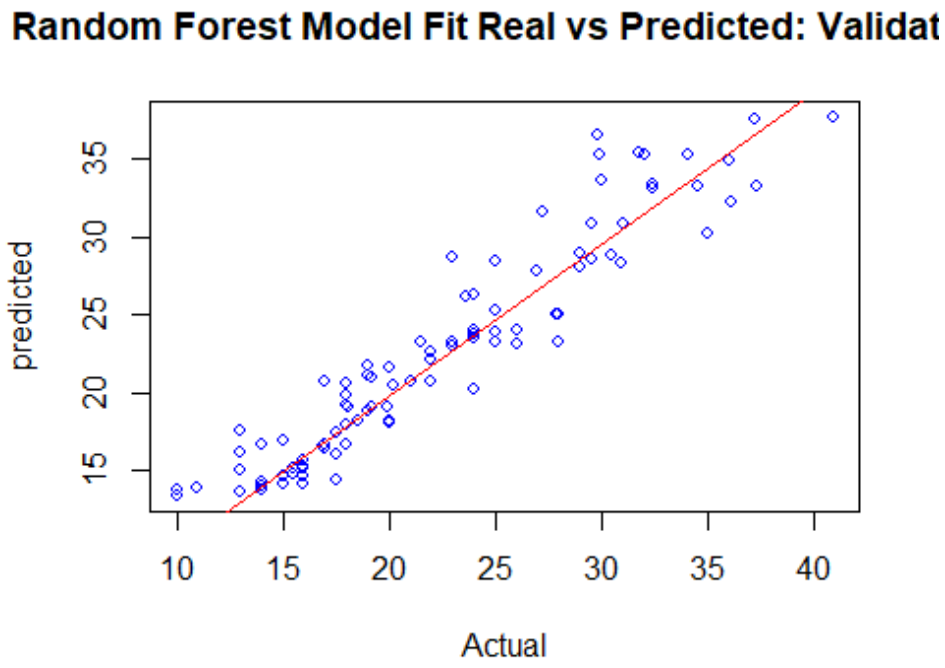
```
## 20.55394 26.51389 22.28111 22.50111 27.19722 13.47000 15.12167 12.26556
##       93       95       96       98       99      102      103      105
## 13.71889 12.64111 12.28222 19.24494 16.40500 21.84622 26.80972 12.26889
##      109      110      112      113      116      120      122      123
## 21.11333 20.66722 20.54028 20.37500 14.17056 20.98444 15.98500 21.86361
##      124      125      127      128      129      140      141      142
## 20.83833 12.86111 20.95906 19.98861 16.35761 13.62222 13.96222 27.55667
##      143      144      146      147      148      149      152      153
## 28.22389 25.54572 31.23944 27.46056 26.79667 25.90294 30.32222 19.42883
##      154      155      156      158      159      160      162      164
## 17.46861 15.92556 16.41444 14.93972 15.14222 14.31583 16.10083 17.41450
##      165      168      169      170      172      177      182      184
## 20.46467 28.58528 23.34489 19.62194 23.51589 18.96244 32.12544 26.20583
##      186      187      191      192      196      197      199      201
## 25.97322 27.57306 14.74750 20.68911 29.95139 26.12628 32.35378 17.49944
##      205      207      208      209      210      211      212      213
## 31.93033 25.89350 20.52556 14.42417 20.67600 19.72144 17.67183 15.51361
##      215      217      218      221      225      226      227      232
## 14.87972 32.03383 29.13706 32.23006 15.08556 17.96483 19.65783 15.27278
##      234      235      236      237      239      243      245      247
## 29.62183 24.16544 25.94956 24.13494 31.83839 24.29067 37.88200 34.31850
##      251      252      254      257      258      259      267      268
## 18.42650 19.02622 20.00572 20.04972 19.59461 20.27856 29.44117 26.89878
##      269      272      273      274      278      279      282      283
## 27.33461 23.88694 23.86683 25.56300 19.34228 32.87622 21.96156 23.91828
##      284      285      286      287      288      289      291      292
## 20.30517 20.41494 17.65139 18.28072 16.83528 17.88639 16.42378 18.70533
##      293      298      299      302      306      307      308      309
## 17.29806 24.57806 20.10239 32.44778 28.31367 27.01072 26.49706 32.28694
##      310      311      312      314      315      318      319      322
## 38.63439 38.74650 36.28339 27.75422 26.98789 34.53044 31.37783 32.99567
##      323      325      326      328      329      332      335      336
## 42.02994 40.97528 43.07811 34.10722 30.32817 35.54828 27.41528 33.15189
##      341      342      344      345      346      348      349      351
## 25.95267 25.35894 37.75528 37.37233 36.67300 37.63933 37.47050 34.30761
##      354      358      359      361      362      364      365      367
## 34.29689 31.59239 31.08867 28.60861 25.72061 22.89828 23.64933 21.34122
##      368      369      370      371      372      374      378      379
## 29.22222 28.46189 33.45278 29.70244 30.54856 25.69156 34.40439 36.99861
##      383      387      388      389      390      392      395      396
## 33.69528 27.82117 30.75622 26.68589 26.35467 33.57556 40.90644 32.49333
##      397      398
## 28.45744 29.37889

RFValPredict

##        7        9       13       14       22       23       25       26
## 14.35528 14.09194 14.18556 16.65333 23.47444 23.28056 20.79411 13.42500
##       27       28       33       55       61       72       73       78
## 13.82778 13.94000 28.40278 30.20472 21.61000 21.75028 14.67056 22.18556
```

```
##        82        92        94       100       101       108       115       117
## 23.23000 13.70361 13.80889 19.24728 17.98328 19.83167 23.18833 14.17083
##       119       121       130       131       133       134       135       136
## 26.23722 21.07283 30.88667 24.02361 23.92306 15.71917 15.29472 16.67872
##       150       157       161       163       166       167       171       174
## 24.00750 14.70667 16.69167 16.89394 18.21722 17.60167 23.03689 23.82250
##       175       176       178       180       183       185       188       194
## 20.59222 28.08100 23.29172 20.78444 25.08694 25.25028 14.39278 20.26244
##       198       200       203       204       214       216       223       224
## 28.99378 18.12867 17.51611 28.57322 15.10139 16.19783 16.42917 14.81333
##       228       229       231       233       241       242       244       246
## 18.78956 18.22800 15.11917 15.23250 28.86317 22.64967 23.29089 32.28550
##       250       253       255       263       265       266       270       276
## 19.15456 20.99517 20.49089 19.06667 19.15200 16.08372 28.29594 20.74400
##       280       290       304       305       313       324       331       333
## 30.86978 16.61878 35.37211 33.25428 37.52161 25.03967 37.67628 36.53583
##       337       338       339       343       350       353       355       357
## 26.19522 33.14822 31.60611 33.58444 35.29578 35.29422 33.25939 33.42822
##       375       381       385       393
## 28.73717 34.90700 35.31589 27.76400

RFTestPredict

##        2        4        5        8       11       16       20       21
## 13.22778 17.16500 17.13167 14.35528 15.96389 20.82217 27.02028 22.67611
##       24       31       32       34       37       50       53       58
## 23.46417 23.46111 24.13500 20.69694 17.75028 24.58722 28.89389 23.08056
##       59       65       67       68       69       71       84       87
## 27.12000 13.54556 14.14444 13.54306 13.18556 13.76028 27.07167 13.84556
##       88       89       97      104      106      107      111      114
## 13.90056 13.90056 13.94389 12.25111 13.38500 13.52222 21.74389 21.45422
##      118      126      132      137      138      139      145      151
## 27.96556 20.56006 29.77667 13.86222 12.86556 13.78889 31.09000 24.41250
##      173      179      181      189      190      193      195      202
## 26.21867 20.50506 23.31894 14.45167 14.40417 18.18333 19.31050 16.95850
##      206      219      220      222      230      238      240      248
## 28.30028 30.84656 25.41506 15.91639 15.22500 31.68672 32.27617 35.34072
##      249      256      260      261      262      264      271      275
## 33.81778 24.49789 20.33994 20.77894 18.90367 19.59133 27.21478 23.58656
##      277      281      294      295      296      297      300      301
## 23.50289 21.42356 33.47389 34.04978 33.31628 28.02244 26.09472 20.07333
##      303      316      317      320      321      327      330      334
## 31.71556 29.15578 23.54272 31.64028 33.28767 36.30922 36.41461 25.12872
##      340      347      352      356      360      363      366      373
## 27.39111 37.83450 34.89861 34.15367 30.38550 28.63522 25.52228 27.65911
##      376      377      380      382      384      386      391      394
## 34.89939 36.42294 36.22156 34.39667 35.53089 35.23689 30.13778 26.67050

RFTrainMSE=mean((train$mpg- RFTrainPredict)^2)
RFTrainSTDev=sd(train$mpg - RFTrainPredict)
```

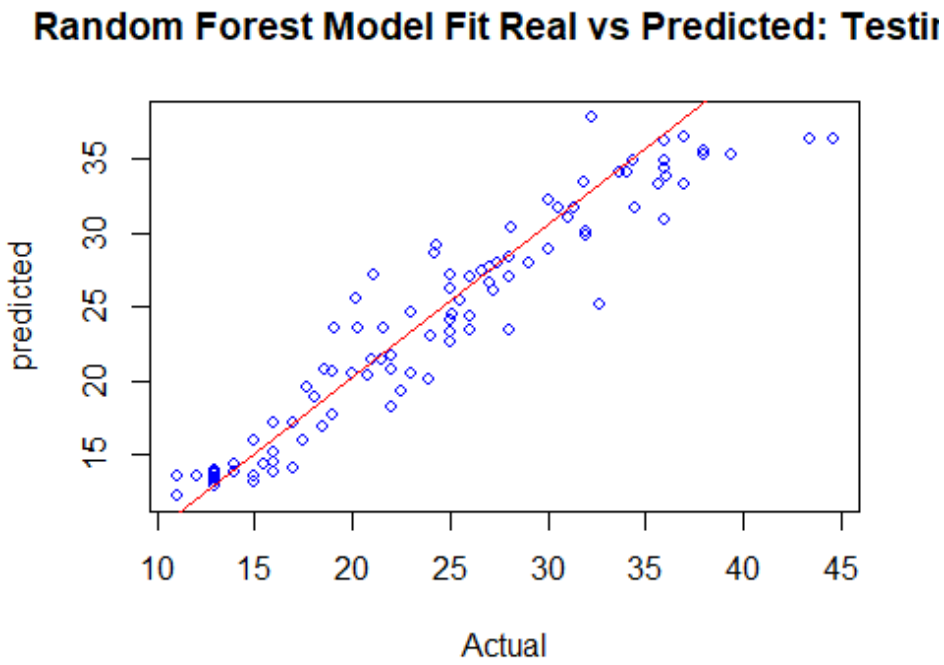
```
RFTrainRMSE=(RFTrainMSE)^0.5
RFValMSE=mean((val$mpg-RFValPredict)^2)
RFVALSTDev=sd(val$mpg - RFValPredict)
RFValRMSE=(RFValMSE)^0.5
RFTestSTDev=sd(test$mpg - RFTestPredict)
RFTestMSE=mean((test$mpg-RFTestPredict)^2)
RFtestRMSE=(RFTestMSE)^0.5


#Fit plots to check prediction accuracy from all samples
TrainregRF=lm(mpg~RFTrainPredict, data=train)
plot(train$mpg,RFTrainPredict,col='blue',main = 'Random Forest Model Fit Real
vs Predicted: Training',pch=1,cex=0.9,type = "p",xlab = "Actual",ylab =
"predicted")
abline(TrainregRF,col="red")
```




```
ValregRF=lm(mpg~RFValPredict,data = val)
plot(val$mpg,RFValPredict,col='blue',main = 'Random Forest Model Fit Real vs
Predicted: Validation',pch=1,cex=0.9,type = "p",xlab = "Actual",ylab =
"predicted")
abline(ValregRF,col="red")
```




```
TestregRF=lm(mpg~RFTestPredict, data=test)
plot(test$mpg,RFTestPredict,col='blue',main = 'Random Forest Model Fit Real
vs Predicted: Testing',pch=1,cex=0.9,type = "p",xlab = "Actual",ylab =
"predicted")
abline(TestregRF,col="red")
```


Random Forest Model Fit Real vs Predicted: Testi
predicted
15
20
25
30
35
10
15
20
25
30
35
40
45
Actual

```
##############################################################################
###########
```

Neural network: code and output

```
#import datatset mpg into r
mpg=read.csv('C:/Users/USER/Google Drive/2017/School/STK
795/Research/Application research/mpg.csv')
mpg= subset(mpg,select = -c(horsepower, name)) #remove columns with missing
values and characters that do not add to reg
#str(mpg) #horsepower and name columns removed
##########################################

#library(dplyr) #load grammar library

#library(readr)
#scale data in order normalise to fit neural net

dn=(max(mpg$mpg)-min(mpg$mpg))+min(mpg$mpg) #descaling parameter
dn

## [1] 46.6

apply(mpg, 2,range)

##       mpg cylinders displacement weight acceleration model_year origin
## [1,]  9.0         3           68   1613          8.0         70      1
## [2,] 46.6         8          455   5140         24.8         82      3

maxV=apply(mpg, 2, max)
minV=apply(mpg, 2, min)
scmpg=as.data.frame(scale(mpg,center = minV, scale = maxV-minV))

#create samples subject to the 50,25,25 percent split as per Biau2016

set.seed(123)

samples = sample(seq(1, 3), size = nrow(mpg), replace = TRUE, prob = c(.5,
.25, .25))
train = scmpg[samples == 1,]
test = scmpg[samples == 2,]
val = scmpg[samples == 3,]

#library(MASS)
library(neuralnet) #load neural network package

## Warning: package 'neuralnet' was built under R version 3.3.3

#create a default formula 'form' to enter into arguments
vars=colnames(mpg)
predictvars=vars[!vars%in%"mpg"]
```

```
predictvars=paste(predictvars,collapse = "+")
form=as.formula(paste("mpg~",predictvars,collapse = "+"))
predictvars #explanatory variables

## [1] "cylinders+displacement+weight+acceleration+model_year+origin"

form #full formula

## mpg ~ cylinders + displacement + weight + acceleration + model_year +
##     origin

#############################################################################
#######
#fit a neural network with 2 hidden layers, 10 repetitions == TRAINING

nn=neuralnet(formula=form,data = train,linear.output = TRUE, hidden =
2,err.fct = "sse", rep = 10)
str(nn) #netresult list containts the preditions based on the testing set

## List of 13
##  $ call                : language neuralnet(formula = form, data = train,
hidden = 2, rep = 10, err.fct = "sse",      linear.output = TRUE)
##  $ response            : num [1:210, 1] 0.239 0.239 0.16 0.16 0.133 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. ..$ : chr "mpg"
##  $ covariate           : num [1:210, 1:6] 1 1 1 1 1 0.2 0.6 0.6 0.2 1 ...
##  $ model.list          :List of 2
##   ..$ response : chr "mpg"
##   ..$ variables: chr [1:6] "cylinders" "displacement" "weight"
"acceleration" ...
##  $ err.fct             :function (x, y)
##   ..- attr(*, "type")= chr "sse"
##  $ act.fct             :function (x)
##   ..- attr(*, "type")= chr "logistic"
##  $ linear.output       : logi TRUE
##  $ data                :'data.frame': 210 obs. of  7 variables:
##   ..$ mpg         : num [1:210] 0.239 0.239 0.16 0.16 0.133 ...
##   ..$ cylinders   : num [1:210] 1 1 1 1 1 0.2 0.6 0.6 0.2 1 ...
##   ..$ displacement: num [1:210] 0.618 0.646 0.933 0.832 0.703 ...
##   ..$ weight      : num [1:210] 0.536 0.517 0.773 0.634 0.566 ...
##   ..$ acceleration: num [1:210] 0.2381 0.1786 0.119 0.0298 0 ...
##   ..$ model_year  : num [1:210] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ origin      : num [1:210] 0 0 0 0 0 1 0 0 1 0 ...
##  $ net.result          :List of 10
##   ..$ : num [1:210, 1] 0.172 0.178 0.142 0.15 0.135 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1] 0.159 0.161 0.109 0.127 0.14 ...
##   .. ..- attr(*, "dimnames")=List of 2
```

```
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1] 0.167 0.171 0.121 0.139 0.15 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1] 0.162 0.167 0.111 0.132 0.147 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1] 0.171 0.179 0.116 0.142 0.157 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1] 0.154 0.159 0.106 0.13 0.144 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1] 0.198 0.206 0.11 0.148 0.209 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1] 0.166 0.174 0.122 0.153 0.167 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1] 0.184 0.205 0.143 0.205 0.233 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1] 0.157 0.162 0.112 0.129 0.136 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##  $ weights            :List of 10
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] 0.218 -0.113 2.187 2.851 0.205 ...
##   .. ..$ : num [1:3, 1] -0.238 -0.829 1.779
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] 0.6129 -0.0926 0.6418 3.4712 -0.301 ...
##   .. ..$ : num [1:3, 1] 1.26 -0.64 -0.54
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -1.02 0.11 -0.11 -3.29 1.03 ...
##   .. ..$ : num [1:3, 1] -0.712 1.234 0.801
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -1.369 0.096 0.504 -3.678 1.017 ...
##   .. ..$ : num [1:3, 1] 1.257 0.814 -1.182
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -1.576 -0.527 -0.878 -2.245 1.298 ...
##   .. ..$ : num [1:3, 1] 0.606 0.576 -0.522
```

```
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] 0.8087 -0.1892 -0.0118 -0.3589 -0.0432 ...
##   .. ..$ : num [1:3, 1] 0.0582 1.2461 -0.6822
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -0.655 1.49 -4.448 -2.083 -3.015 ...
##   .. ..$ : num [1:3, 1] 1.039 3.432 -0.965
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] 1.76046 -0.15682 -0.00457 -0.55919 -0.26018 ...
##   .. ..$ : num [1:3, 1] -0.901 1.342 0.724
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -1.763 0.315 0.869 -3.509 0.919 ...
##   .. ..$ : num [1:3, 1] 0.172 1.507 -0.171
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] 0.7783 0.0984 -0.7365 3.8271 0.2291 ...
##   .. ..$ : num [1:3, 1] 0.487 -1.396 0.994
##  $ startweights       :List of 10
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -1.185 2.199 1.312 -0.265 0.543 ...
##   .. ..$ : num [1:3, 1] -0.516 -0.993 1.676
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -0.441 -0.723 -1.236 -1.285 -0.574 ...
##   .. ..$ : num [1:3, 1] 1.9553 -0.0903 0.2145
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -0.739 -0.574 -1.317 -0.183 0.419 ...
##   .. ..$ : num [1:3, 1] -1.36 -0.665 0.485
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -0.3756 -0.5619 -0.3439 0.0905 1.5985 ...
##   .. ..$ : num [1:3, 1] 2.293 1.548 -0.133
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -1.7565 -0.3888 0.0892 0.845 0.9625 ...
##   .. ..$ : num [1:3, 1] 0.736 0.386 -0.266
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] 0.118 0.134 0.221 1.641 -0.219 ...
##   .. ..$ : num [1:3, 1] 0.021 1.25 -0.715
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -0.753 -0.939 -1.053 -0.437 0.331 ...
##   .. ..$ : num [1:3, 1] 0.704 -0.106 -1.259
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] 1.684 0.911 0.237 1.218 -1.339 ...
##   .. ..$ : num [1:3, 1] -0.722 1.519 0.377
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -2.052 -1.364 -0.201 0.866 -0.102 ...
##   .. ..$ : num [1:3, 1] -0.18 1.01 -1.99
##   ..$ :List of 2
##   .. ..$ : num [1:7, 1:2] -0.427 0.117 -0.893 0.334 0.411 ...
##   .. ..$ : num [1:3, 1] 0.378 -0.945 0.857
##  $ generalized.weights:List of 10
##   ..$ : num [1:210, 1:6] 0.341 0.332 0.376 0.368 0.418 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
```

```
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1:6] 0.182 0.186 0.111 0.144 0.157 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1:6] 0.0594 0.0606 0.0336 0.0451 0.0512 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1:6] 0.1857 0.1951 0.0946 0.1456 0.1815 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1:6] 0.462 0.499 0.249 0.405 0.461 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1:6] -0.247 -0.228 -0.542 -0.395 -0.322 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1:6] 0.781 0.823 0.3 0.59 0.944 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1:6] -0.149 -0.133 -0.32 -0.22 -0.181 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1:6] -1.005 -0.921 -1.359 -0.72 -0.336 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##   ..$ : num [1:210, 1:6] -0.02038 -0.01717 0.00894 0.01179 0.01431 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:210] "1" "3" "6" "10" ...
##   .. .. ..$ : NULL
##  $ result.matrix      : num [1:20, 1:10] 0.67941 0.00931 1748 0.21762 -
0.1135 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:20] "error" "reached.threshold" "steps"
"Intercept.to.1layhid1" ...
##   .. ..$ : chr [1:10] "1" "2" "3" "4" ...
##  - attr(*, "class")= chr "nn"

plot(nn) #visualise nn

TrainfitNN=compute(nn,train[2:7])

TrainfitNN$net.result=TrainfitNN$net.result*dn
```

```
train=train*dn


TrainregNN=lm(formula = mpg~TrainfitNN$net.result , data = train)
plot(train$mpg,TrainfitNN$net.result,col='blue',main = 'Neural Model Fit Real
vs Predicted: Training',pch=1,cex=0.9,type = "p",xlab = "Actual",ylab =
"predicted")
abline(TrainregNN,col="red")

TrainingError=train$mpg- TrainfitNN$net.result
MSENNTrain=mean((TrainingError)^2)

TrainingRMSE=(MSENNTrain)^0.5
TrainingSTDev=sd(TrainingError)
TrainingRMSE
```

```
## [1] 3.748490718
```

```
TrainingSTDev
```

```
## [1] 3.757447456
```

```
#SEE HOW WELL THE MODEL FITS VIA VALIDATION AND TESTING (predictions)
ValFitNN=compute(nn,val[2:7])
TestFitNN=compute(nn,test[,2:7])
ValFitNN$net.result=ValFitNN$net.result*dn
TestFitNN$net.result=TestFitNN$net.result*dn
val=val*dn
test=test*dn
#ValFitNN
summary(ValFitNN)
```

```
##            Length Class  Mode
## neurons     2     -none- list
## net.result 92     -none- numeric
```

```
str(ValFitNN)
```

```
## List of 2
##  $ neurons   :List of 2
##   ..$ : num [1:92, 1:7] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:92] "7" "9" "13" "14" ...
##   .. .. ..$ : chr [1:7] "1" "cylinders" "displacement" "weight" ...
##   ..$ : num [1:92, 1:3] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:92] "7" "9" "13" "14" ...
##   .. .. ..$ : NULL
##  $ net.result: num [1:92, 1] 6.96 7.18 8.37 13.95 13.66 ...
##   ..- attr(*, "dimnames")=List of 2
```

```
##   .. ..$ : chr [1:92] "7" "9" "13" "14" ...
##   .. ..$ : NULL

#TestFitNN
summary(TestFitNN)

##            Length Class  Mode
## neurons     2     -none- list
## net.result 96     -none- numeric

str(TestFitNN)

## List of 2
##  $ neurons   :List of 2
##   ..$ : num [1:96, 1:7] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:96] "2" "4" "5" "8" ...
##   .. .. ..$ : chr [1:7] "1" "cylinders" "displacement" "weight" ...
##   ..$ : num [1:96, 1:3] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:96] "2" "4" "5" "8" ...
##   .. .. ..$ : NULL
##  $ net.result: num [1:96, 1] 8.06 8.38 7.41 6.36 9.14 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:96] "2" "4" "5" "8" ...
##   .. ..$ : NULL

#######EVALUATE########
ValidationError=val$mpg- ValFitNN$net.result
plot(ValidationError)
TestingError=test$mpg - TestFitNN$net.result
MSENNVal=mean((ValidationError)^2)
MSENNTest=mean((TestingError)^2)

ValidationRMSE=(MSENNVal)^0.5
TestingRMSE=(MSENNTest)^0.5

ValidationSTDev=sd(ValidationError)
TestingSTDev=sd(TestingError)

ValidationRMSE

## [1] 3.169557582

ValidationSTDev

## [1] 3.146972395

TestingRMSE

## [1] 3.531401505
```

```
TestingSTDev

## [1] 3.523669235

ValregNN=lm(formula = mpg~ValFitNN$net.result , data = val)
plot(val$mpg,ValFitNN$net.result,col='blue',main = 'Neural Model Fit Real vs
Predicted:  Validation',pch=1,cex=0.9,type = "p",xlab = "Actual",ylab =
"predicted")
abline(ValregNN,col="red")

TestregNN=lm(mpg~TestFitNN$net.result, data = test)
plot(test$mpg,TestFitNN$net.result,col='blue',main = '',pch=1,cex=0.9,type =
"p",xlab = "Actual",ylab = "predicted")
abline(TestregNN,col="red")
################################################################################
###########
```

On the generalization of scalar diffusion processes with applications to financial time series

Jacobus Marthinus van der Berg 13035836

WST795 Research Report

Submitted in partial fulfillment of the degree BSc(Hons) Mathematical Statistics

Supervisor: Dr. Etienne A.D. Pienaar

Department of Statistics, University of Pretoria

31 October 2017

Abstract

Diffusion processes, according to [12], allow for the investigation and quantification of the dynamics of various real world financial models. The dynamics of diffusion processes are governed by stochastic differential equations (SDEs), which dictate how such processes evolve over time. Although diffusion processes are assumed to evolve continuously over time, the real-world processes can only be observed at discrete time epochs. Therefore to analyze such processes, the time-dimension over which the diffusion processes are defined, needs to be discretized, e.g. monthly or daily financial data.

The transition density function of a diffusion process provides valuable insights on the process's dynamics. It relates the probability of observing the trajectory of a diffusion process over discrete epochs. Unfortunately closed-form solutions for the true transition density function cannot be obtained for all diffusion processes. The premise of the present paper is to develop an efficient closed-form approximation for a diffusion process. A scalar diffusion process will firstly be considered to develop the relevant approximations and inferential methods. The obtained methodology will then be applied on a mixed-effects (random and deterministic parameters) diffusion process, which is a generalization of the scalar (all parameters deterministic) diffusion process.

The present paper will analyze and develop the applicable techniques on the scalar Cox, Ingersoll and Ross (CIR) process. Although there exists a closed-form true transition density function for the scalar CIR process, approximations for the transition density function will be derived and then compared to the true transition density function. A Hermite-series transition density function approximation, as developed by [1] will be derived. In an attempt to improve the approximation's accuracy provided by the Hermite-series transition density function approximation, moment truncation, as developed by [12], will be utilized to develop a more efficient closed-form approximation for the transition density function.

Once a closed-form transition density function approximation is obtained a likelihood function will be derived and utilized in maximum likelihood estimation, based on the observed Standard and Poore 500 volatility index (S&P500VIX). By simulating future trajectories of the diffusion process, based on the maximum likelihood estimates, predictions can be made.

Finally, a generalization of the scalar CIR process will then be formulated on which the developed approximation and inferential techniques will be applied. This generalization will be known as the mixed-effects CIR process, since this process will consist of both deterministic and random parameters.

Declaration

I, *Jacobus Marthinus van der Berg,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Jacobus Marthinus van der Berg

_ _

Etienne A.D. Pienaar, PhD

_ _

31 October 2017

Acknowledgments

Herewith would I like to thank and express immense gratitude towards Dr Pienaar for all his assistance, advice and patience through the process of completing this research report. Secondly, I give thanks to the South African Reserve Bank for the financial support.

Contents

List of Figures

List of Algorithms

1 Introduction

According to [12], diffusion processes can be viewed as the stochastic counterparts to systems of ordinary differential equations (ODEs). Furthermore, these diffusion processes evolve continuously over time, inherit the Markov property and are driven by Brownian motion. The dynamics of a general diffusion process are governed by the SDE, [1]:

$$dX_t = \mu(X_t, t; \boldsymbol{\theta})dt + \sigma(X_t, t; \boldsymbol{\theta})dW_t, \tag{1}$$

$$\text{s.t } t \in [s, T],$$

where X_t is the state variable of interest, $\mu(X_t, t, \boldsymbol{\theta})$ and $\sigma(X_t, t, \boldsymbol{\theta})$ are time- and state dependent functions, and $\boldsymbol{\theta}$ the parameter vector. If all parameters of $\boldsymbol{\theta}$ are scalars, the model is referred to as a scalar diffusion process. If one or more of the parameters of $\boldsymbol{\theta}$ becomes random in nature then this generalized version of the scalar diffusion process is referred to as a mixed-effects model. It is worth noting that $[s, T]$ is assumed to be the continuous time-dimension of the diffusion process given in Equation 1, and X_s the initial value of the process.

Various results in the field of stochastic calculus, such as *Itô*'s Lemma, can be applied to Equation 1 to derive the dynamics of functions of the SDEs, and to approximate the stochastic integrals in closed-form. In Appendix A some fundamental and relevant stochastic calculus and stochastic processes results are discussed.

Scalar diffusion processes, in the form of Equation 1 will be explored, particularly in the application of these processes in the financial environment. However, in financial markets, we observe our data in a discrete time-dimension, therefore a variety of simulations and approximations will be used to discretize such valuable continuous-time models for efficient application in the world of finance. The applications will be focused on short-rate modeling, but can be applied to various fields in financial modeling, e.g. derivative pricing. A short-rate model, according to[3], is a time-homogeneous model with an instantaneous- or short rate of interest as the single factor or state variable. In Equation 1, X_t represents the value of a given short-rate at time t. The dynamics of the development of such a short-rate, over the given time-dimension, is of particular interest, and will be investigated. This random movement of the short-rate can be seen in the trajectory of a given diffusion process. According to [2], short-rate models are one of the most explored fields in financial modeling. Examples of familiar short-rate diffusion processes, which can be analyzed and generalized, include those by Vasicek (Ornstein-Uhlenbeck), Cox, Ingersoll and Ross (CIR), Merton, etc. Comparison of such short-rate models can be made on the basis of their dynamics and ability to capture the random development of the short-term interest rate.

The CIR short-rate model will be considered, and through the analysis and inference of the dynamics

of this diffusion process, closed-form transition density function approximations will be derived, which can be used to predict future expected short-rate values. The techniques and approximations developed on a given scalar diffusion process will be applied to a generalized version (mixed-effects model) of the given process.

Diffusion processes form an integral part of financial modeling, and although these processes exhibit attractive theoretical properties, from a modeling perspective, there exist various difficulties in the actual application of such processes in financial markets. The present paper's application will focus on financial short-rate modeling. Analysis of the term-structure (i.e how short-term interest rate processes evolve over time) is of particular interest. In [3] short-rate models are defined as time-homogeneous single-factor models, with the short rate of interest as the single state.

The scalar CIR process, a model which is often used in the analysis of the term structure of interest rates, will be analyzed. This scalar CIR process will also be generalized to a mixed-effects model. The dynamics of both the scalar and generalized model will be investigated and compared.

The evaluation will start by plotting the process's trajectory, as given in Equation 1, over a discrete time-dimension to observe the nature, movement and stationarity of the diffusion process. The Euler-Maruyama scheme will be utilized to simulate these continuous-time trajectories, over a discrete time-dimension, to obtain a sampled distribution. Since a closed-form true transition density function does not exist for all diffusion processes, the aim of this paper will be to derive an efficient closed-form approximation for such processes.

A variety of techniques and methodologies have been be developed in prior literature to derive satisfactory transition density function approximations that are used to perform inference on the dynamics and capabilities of such processes. The approximation methods developed and utilized in the present paper include: moment truncation methods, as done by [12], the Hermite-series transition density function approximation of transition densities, as in [1]. Once a closed-form approximation to the transition density function is found, a likelihood function will be derived and through maximum likelihood estimation, the unknown parameters can be estimated. Approximations of the transition densities and maximum likelihood estimation of the unknown parameter vector, as in [1] also provides valuable insights into the approximate transition probabilities and inferences about the parameters of such processes. Since we obtain maximum likelihood estimators, we have that these parameters inherit welcome asymptotic results, which will be useful in the analysis of the diffusion process's steady-state distribution. The underlying distributions will not always be Gaussian of nature, which we will see in our Euler-Maruyama simulations, and this should be kept in mind and adjusted for in the proceedings of this paper.

The focus will be on deriving a closed-form approximation for the true transition density function by moment truncation, as developed by [12], where a saddlepoint transition density function approximation,

depending only the theoretical cumulants of the diffusion process, will be developed. The aim is to improve on the approximation accuracy provided by the Hermite-series transition density function approximation.

The CIR process and generalized version thereof will be investigated to develop an accurate closed-form transition density function approximation, to model the random movement of a given short-rate over a given time-dimension. There exists a closed-form solution for the true transition density function of the CIR process, however this true closed-form solution will be utilized in comparisons to develop an efficient closed-form transition density function approximation, which can be applied when a true transition density function does not exist. The approximation techniques discussed and developed in the present paper will be applied to a mixed-effects CIR process. Moment truncation, as developed by [12], will be utilized to develop an efficient closed-form approximation for the transition densities, which in turn will be compared to the Hermite-series transition density function approximation, as developed by [1]. The true transition density function for the scalar CIR process will be compared to the derived approximations to determine the approximation's accuracy. Once a closed-form transition density function is obtained, a likelihood function will be developed and utilized in maximum likelihood estimation based on the S&P500 volatility index (S&P500VIX). The obtained maximum likelihood estimates will be used to simulate random trajectories and analyze the given process's dynamics. A prediction of future values can then be made and compared to known forecasting techniques.

2 Diffusion process analysis

2.1 Diffusion processes

Firstly, a formal definition, by [11], of diffusion processes is given:

Definition 1. X_t, a Markov Process with $p(\eta, t|x, s)$ as the transition probability, is called a diffusion process if the following three conditions are met:

Firstly, let $y - x = \lambda$, then for all x and all $\epsilon > 0$:

1. Continuity of the process: $\int_{|\lambda|>\epsilon} p(dy, t|x, s) = o(t - s)$ uniformly over $[s, t]$ for $s < t$.
2. Drift coefficient: there exists a real-valued function $\mu(x, t)$ s.t. $\int_{|\lambda|\leq\epsilon} p(dy, t|x, s)\lambda = \mu(x, t)(t - s) + o(t - s)$ uniformly over $[s, t]$ for $s < t$.
3. Diffusion coefficient: there exists a real-valued function $\sigma(x, t)$ s.t. $\int_{|\lambda|\leq\epsilon} p(dy, t|x, s)\lambda^2 = \sigma(x, t)(t - s) + o(t - s)$ uniformly over $[s, t]$ for $s < t$.

Diffusion processes, according to [12], can be considered as the class of models which are represented as solutions to stochastic differential equations which evolve continuously over time, possessing the Markov

property and are driven by change in time and Brownian motion. Therefore the process is driven by both deterministic and random or stochastic forces. Combining the definitions, for a general diffusion process, X_t , of [12] and [1], the dynamics of a diffusion process can be given by the stochastic differential equation:

$$dX_t = \mu(X_t, t; \boldsymbol{\theta})dt + \sigma(X_t, t; \boldsymbol{\theta})dW_t \tag{2}$$

where $\mu(X_t, t; \boldsymbol{\theta})$ and $\sigma^2(X_t, t; \boldsymbol{\theta})$ are the drift and diffusion of the diffusion process, respectively. Note that for the purpose of this research report, the state variable, X_t, is defined over a single dimension, but can be extended to $p > 1$ dimensions. W_t, $t \geq 0$, is a Wiener process or a Brownian motion, in one dimension, defined as follows:

Definition 2. W_t for $t \geq 0$, where $W_t \in \mathbb{R}$, is a one-dimensional Brownian motion or a Wiener Process if, according to [6], the following fundamental properties hold:

- If $t_0 < t_1 < ... < t_{n-1} < t_n$ then W_{t_0}, $W_{t_1} - W_{t_0}$, ..., $W_{t_n} - W_{t_{n-1}}$ are independent; hence Brownian motion has independent increments.
- If $s, t \geq 0$, and E a possible subset of a $\sigma - algebra$, say ξ, as in Appendix A1, then $W_{t+s} - W_s \sim N(0, t)$, i.e $\mathbb{P}(W_{t+s} - W_s \in E) = \frac{1}{\sqrt{2\pi t}} \int_E e^{-\frac{y^2}{2t}} dy$.
- $t \to W_t$ is continuous, with probability one.
- $W_0 = 0$.

The time-dimension under consideration is $[s, T]$ s.t $s \leq t \leq T$. By [12], we can model diffusion processes, as in Equation 2, by differential equations, where the change in time, dt, governs the deterministic part of the process and the dW_t-term (Brownian motion) drives the random or stochastic part of the process. It is said that the sample paths of the diffusion process, X_t, are determined by a stochastic differential equation as in Equation 2. An individual occurrence or sample path of the state space, Ω, is given by $X_t(\omega) \in \mathbb{R}$ s.t $\omega \in \Omega$. This paper assumes that the sample path followed, is known and therefore $X_t(\omega) = X_t$. According to [10] the drift and diffusion can be viewed as the instantaneous changes in the process under determination:

$$\mu(X_t, t; \boldsymbol{\theta}) = \lim_{\Delta \to 0} \mathbb{E}\left[\frac{X_{t+\Delta} - X_t | X_t}{\Delta}\right] \text{ and}$$

$$\sigma^2(X_t, t; \boldsymbol{\theta}) = \lim_{\Delta \to 0} \mathbb{E}\left[\frac{(X_{t+\Delta} - X_t)^2 | X_t}{\Delta}\right].$$

It is worth noting that the diffusion is also referred to as the instantaneous variance of the process. The movement in the process, due to time changes, are described by the drift, whereas the magnitude of

random movements in the process are described by the diffusion. According to [12], to analyze a diffusion process, a solution to Equation 2, should be derived over an appropriate transition horizon. Given that the process started in the known state of X_s at time $s < t$, the trajectory of the stochastic process X_t, from Definition 22, is given by:

$$X_t = X_s + \int_s^t \mu(X_v, v; \boldsymbol{\theta})dv + \int_s^t \sigma(X_v, v; \boldsymbol{\theta})dW_v. \tag{3}$$

The drift and diffusion coefficients determine the existence of a solution to Equation 22 in Appendix A2. X_t as given in Equation 2 is said to have a unit diffusion if the diffusion-coefficient is equal to one, i.e $\sigma(X_t, t; \boldsymbol{\theta}) = 1$.

2.2 Scalar diffusion processes and mixed-effects diffusion processes

Consider the general diffusion process given in Equation 2, with parameter vector $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_m\}$ for some $m \in \mathbb{N}$. For a scalar diffusion process, θ_i is fixed or deterministic of nature for all values of $i = 1, 2, ..., m$. The works of [5] states that when a random effect is incorporated in the diffusion process, in the sense that at least one θ_i for some $i = 1, 2, ..., m$ become random of nature , with a known or unknown probability distribution, then this generalized version of the scalar diffusion model is referred to as a mixed-effects diffusion process. In some cases, as with the CIR process, an analytical expression for the true transition density function and hence a likelihood function for inferential purposes can be obtained. However when a mixed-effects model is considered, generally an analytical expression for the true transition density function and hence a likelihood function cannot be obtained. This problem leads to the premise of the present paper, which is to develop an efficient approximation for the transition density function of a scalar or mixed-effects diffusion process, which can be applied in the absence of a true transition density for the given diffusion process. The theory will be developed on a scalar diffusion process, which will be extended to a mixed-effects diffusion process.

2.3 Analysis of the true transition density function of a scalar diffusion process

The ideal in modeling any diffusion process, is the existence of a true transition density function, in closed-form. Unfortunately this theoretical density, in closed-form rarely exist for diffusion processes. As in Definition 9 let $(S, \mathcal{F}, \mathbb{P})$, where $S \subseteq \mathbb{R}$, be a probability space, over which the true transition density function, $p(y, t|x, s)$, of the diffusion process in Equation 22 in Appendix A2, is defined. It is worth noting that the continuous-time-dimension is discretized. According [11] this true transition density function, if it exists, can be obtained through solving the following equations:

Kolmogorov Forward Equation:

$$\frac{\partial}{\partial t}p(y,t|x,s;\boldsymbol{\theta}) = -\frac{\partial}{\partial y}[\mu(y,t)p(y,t|x,s;\boldsymbol{\theta})] + \frac{1}{2}\frac{\partial^2}{\partial y^2}[\sigma^2(y,t)p(y,t|x,s;\boldsymbol{\theta})], \tag{4}$$

Kolmogorov Backward Equation:

$$-\frac{\partial}{\partial s}p(y,t|x,s;\boldsymbol{\theta}) = -\mu(x,s)\frac{\partial}{\partial x}p(y,t|x,s;\boldsymbol{\theta}) + \frac{1}{2}\sigma^2(x,s)\frac{\partial^2}{\partial x^2}p(y,t|x,s;\boldsymbol{\theta}). \tag{5}$$

As stated earlier, the true transition density function of the trajectory of the process given in Equation 2 can rarely be obtained in closed-form, either because the theoretical density does not exist or it is immensely difficult and time-consuming to obtain the true closed-form transition density function. This leads to the main purpose of this paper, that is to obtain an accurate closed-form approximation for this transition density function. Examples of diffusion processes where a true transition density function exists, according to [1], include those by Cox, Ingersoll and Ross (CIR), Black and Scholes and Vasicek. The CIR process will be thoroughly analyzed in the application-section of this paper.

2.4 transition density function approximations of scalar diffusion processes

Although diffusion processes, as given in Equation 2, gives a full description of the process's trajectory over an infinitesimally time period, in general it is extremely difficult or even impossible to get a theoretical closed-form expression for the transition density function of the process, [1]. Therefore the aim of this section is to derive the general methodology and results for closed-form approximations for the transition probability of the trajectory of the diffusion process given in Equation 2. There exist a variety of approximation techniques, [13] discusses the Markov chain Monte Carlo approach, which is applied to Bayesian statistics, as well a short discussion of the Euler Approximation. However this paper will focus on the Hermite-series transition density function approximation as done by [1] and moment truncation as done by [12].

2.4.1 Euler-Maruyama scheme of a scalar diffusion process

This section, based on the work done by [9], aims to find numerical solutions to stochastic differential equations and hence diffusion processes. Consider a diffusion process as given in Equation 2. This section utilizes the Euler-Maruyama scheme to simulate a numerical solution for the transition density function of the given diffusion process, to get a general idea about the transition density function's shape. Consider the scalar stochastic differential equation which governs the diffusion process, X_t, written in integral form in Equation 3 and in differential form in Equation 2. A numerical approach is now applied to Equation 2 over the time-dimension $[s,T]$ s.t $t \in [s,T]$. Firstly the process start by the dicretization of the time-

dimension: let $\Delta_t = \frac{T}{N}$ for $N \in \mathbb{N}$, and $\eta_i = i\Delta_t$. Denote X_i as the numerical approximation to $X(\eta_i)$. The Euler-Maruyama scheme is now given by the following recursive relationship:

$$X_i = X_{i-1} + \mu(X_{i-1}, i-1; \boldsymbol{\theta})\Delta_t + \sigma(X_{i-1}, i-1; \boldsymbol{\theta})(W_{\eta_i} - W_{\eta_{i-1}}) \tag{6}$$

$$\text{for all } i = s+1, s+2, ..., s+N-1, s+N \text{ and}$$

$$X_s \text{ as initial state.}$$

The integral Equation 3 is a result of:

$$X(\eta_i) = X(\eta_{i-1}) + \int_{\eta_{i-1}}^{\eta_i} \mu(X(u), u; \boldsymbol{\theta})du + \int_{\eta_{i-1}}^{\eta_i} \sigma(X(u), u; \boldsymbol{\theta})dW(u).$$

Remark 3. $W_{\eta_i} - W_{\eta_{i-1}} = \Delta W_t \overset{d}{=} W_{\eta_i - \eta_{i-1}} = W_{i\Delta_t - (i-1)\Delta_t} = W_{\Delta_t}$ where $\Delta W_t \sim N(0, \Delta_t^2)$
The Euler-Maruyama (EM) scheme is an efficient method to get a general idea about the true transition density function over a continuous time-dimension, by the simulation of a discretized process.

2.4.2 Hermite-series transition density function approximation of a scalar diffusion

The theory discussed and derived in this section is the works of [1], with the diffusion process given in Equation 2 as the general model. The premise of this section is to obtain a closed-form transition density function approximation to an unknown true transition density function, namely $p_X(x_t, t|x_s, s; \boldsymbol{\theta})$, i.e the conditional density of $X_t = x_t$, given $X_s = x_s$, s.t. $t \in [s, T]$. These approximations can be utilized to estimate the parameter vector $\boldsymbol{\theta}$ and apply various results to financial data for financial decision making, e.g derivative pricing. The method of Hermite-series transition density function approximation for the transition density function of the general diffusion process, given in Equation 2, will now be discussed.

According to [1], to obtain the general Hermite-series transition density function approximation, $g_X^{(K)}(x_t, t|x_s, s; \boldsymbol{\theta})$, where $K \geq 0$ is the order of approximation, for the general true transition density function, $p_X(x_t, t|x_s, s; \boldsymbol{\theta})$, such that $g_X^{(K)}(x_t, t|x_s, s; \boldsymbol{\theta}) \approx p_X(x_t, t|x_s, s; \boldsymbol{\theta})$, the function, X_t, first needs to be transformed into Y_t, which will have a unit diffusion. The transformation, $X_t \to Y_t$, to get a unit diffusion is referred to as a Lamperti transform, where the techniques of this transformation method is discussed in [7]. The Lamperti Transform to Y_t is given in [1] by the following equation:

$$Y_t = \int^{X_t} \frac{1}{\sigma(v, t; \boldsymbol{\theta})} dv \equiv \varphi(X_t, t; \boldsymbol{\theta}). \tag{7}$$

By an application of *Itô*-Lemma as given in Theorem 25, Y_t has the desired unit diffusion, as given by the following diffusion process:

$$dY_t = \mu_Y(Y_t, t; \boldsymbol{\theta})dt + 1dW_t. \tag{8}$$

The implication of the assumption that $\sigma(X_t, t; \boldsymbol{\theta}) > 0$ in D_{X_t}, is that $\varphi(X_t, t; \boldsymbol{\theta})$ in Equation 7 is increasing and hence invertible. Therefore the drift coefficient $\mu_Y(Y_t, t; \boldsymbol{\theta})$ in Equation 8 can be given as:

$$\mu_Y(Y_t, t; \boldsymbol{\theta}) = \frac{\mu(\varphi^{-1}(Y_t, t; \boldsymbol{\theta}), t, \boldsymbol{\theta})}{\sigma(\varphi^{-1}(Y_t, t; \boldsymbol{\theta}), t, \boldsymbol{\theta})} - \frac{1}{2}\frac{\partial}{\partial X_t}\sigma(\varphi^{-1}(Y_t, t; \boldsymbol{\theta}), t, \boldsymbol{\theta}).$$

The transformation $X_t \to Y_t$ is made in order to derive a closed-form expansion for $p_Y(y_t, t|y_s, s; \boldsymbol{\theta})$, the transition density function of Y_t. Therefore an analytical expansion of $p_Y(y_t, t|y_s, s; \boldsymbol{\theta})$ up to order of approximation $K \geq 0$, namely $g_Y^{(K)}(y_t, t|y_s, s; \boldsymbol{\theta})$, must be found, s.t. $g_Y^{(K)}(y_t, t|y_s, s; \boldsymbol{\theta}) \approx p_Y(y_t, t|y_s, s; \boldsymbol{\theta})$. But since the interest lies in finding an approximation for the transition density function of X_t, i.e $g_X^{(K)}(x_t, t|x_s, s; \boldsymbol{\theta}) \approx p_X(x_t, t|x_s, s; \boldsymbol{\theta})$, the Jacobian formula, as given in [1], can be used to obtain $p_X(x_t, t|x_s, s; \boldsymbol{\theta})$ from $p_Y(y_t, t|y_s, s; \boldsymbol{\theta})$ as follows:

$$\begin{aligned} p_X(x_t, t|x_s, s; \boldsymbol{\theta}) &= \frac{\partial}{\partial x_t}\mathbb{P}[X_t \leq x_t | X_s = x_s] \\ &= \frac{\partial}{\partial x_t}\mathbb{P}[Y_t \leq \varphi(x_t, t; \boldsymbol{\theta}) | X_s = \varphi(x_s, s; \boldsymbol{\theta})] \\ &= \frac{\partial}{\partial x_t}\int_{y_{lower}}^{\varphi(x_t, t; \boldsymbol{\theta})} p_Y(y_t, t|\varphi(x_s, s; \boldsymbol{\theta}), s; \boldsymbol{\theta}) dy_t \\ &= \frac{p_Y(\varphi(x_t, t; \boldsymbol{\theta}), t|\varphi(x_s, s; \boldsymbol{\theta}), s; \boldsymbol{\theta})}{\sigma(\varphi(x_t, t; \boldsymbol{\theta}); \boldsymbol{\theta})}. \end{aligned} \tag{9}$$

Hence, a closed-form Hermite-series transition density function approximation $p_X(x_t, t|x_s, s; \boldsymbol{\theta})$ from $p_Y(y_t, t|y_s, s; \boldsymbol{\theta})$ will be derived using the transformation $X_t \to Y_t$ and the Jacobian formula given in 9. The derivation of the approximation of the transition densities of X_t and Y_t will start with a Hermite-series expansion of the transition density function of Y_t around a Normal density. Up to order $K \geq 0$, the analytical part of the expansion of $p_Y(y_t, t|y_s, s; \boldsymbol{\theta})$ is given by the following expression:

$$g_Y^{(K)}(y_t, t|y_s, s; \boldsymbol{\theta}) = \frac{1}{\sqrt{\Delta}}\phi\left(\frac{y_t - y_s}{\sqrt{\Delta}}\right) e^{\int_{y_s}^{y_t} \mu_Y(\upsilon, t_\upsilon; \boldsymbol{\theta}) d\upsilon} \sum_{k=0}^{K} c_k(y_t, t|y_s, s; \boldsymbol{\theta}) \frac{\Delta^k}{k!} \tag{10}$$

where $\Delta = t - s$ for $t \in [s, T]$,

ϕ denotes the Standard Normal density, i.e $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ for $X \sim N(0, 1)$,

$$c_s(y_t, t|y_s, s; \boldsymbol{\theta}) = 1 \ ,$$

and for all $i > s$

$$c_i(y_t, t|y_s, s; \boldsymbol{\theta}) = i(y_t - y_s)^{-i} \int_{y_s}^{y_t} (\upsilon - y_s)^{i-1} \left[\lambda_{Y_t}(\upsilon; \boldsymbol{\theta}) c_{i-1}(\upsilon, t_\upsilon|y_s, s; \boldsymbol{\theta}) + \frac{1}{2}\frac{\partial^2}{\partial \upsilon^2} c_{i-1}(\upsilon, t_\upsilon|y_s, s; \boldsymbol{\theta})\right] d\upsilon \tag{11}$$

$$\text{s.t. } \lambda_{Y_t}(\upsilon; \boldsymbol{\theta}) = \frac{1}{2}\left[\mu_{Y_t}^2(y_t, t; \boldsymbol{\theta}) + \frac{\partial}{\partial y_t}\mu_{Y_t}(y_t, t; \boldsymbol{\theta})\right].$$

In Equation 10 the term $\frac{1}{\sqrt{\Delta}}\phi\left(\frac{y_t-y_s}{\sqrt{\Delta}}\right)$ is Gaussian and the term $e^{\int_{y_s}^{y_t}\mu_Y(\upsilon,t_\upsilon;\boldsymbol{\theta})d\upsilon}$ corrects for the drift in the diffusion process given in Equation 2 over the time-dimension $[s,T]$. The correction term $\lambda_{Y_t}(\upsilon;\boldsymbol{\theta})$ aids in improving the discretizatian bias caused by using the Gaussian term as well as corrects for the non-normality of $p_Y(y_t,t|y_s,s;\boldsymbol{\theta})$, however there will always remain some level of descretization bias.
By [1] the sequence of equations, for all $K\geq 0$, given in Equation 10, solves the Kolmogorov forward and Kolmogorov backward equations. Hence to obtain a closed-form conditional transition density function for $Y_t=y_t$ given $Y_s=y_s$, i.e $g_Y^{(K)}(y_t,t|y_s,s;\boldsymbol{\theta})$ for all orders of approximation $K\geq 0$, the following adapted Kolmogorov forward and Kolmogorov backward equations, from Equation 4and in5 , respectively, are solved:

$$\begin{aligned}
&\frac{\partial}{\partial\Delta}g_Y^{(K)}(y_t,t|y_s,s;\boldsymbol{\theta})+\frac{\partial}{\partial y_t}[\mu_{Y_t}(y_t,t;\boldsymbol{\theta})g_Y^{(K)}(y_t,t|y_s,s;\boldsymbol{\theta})]-\frac{1}{2}\frac{\partial^2}{\partial y_t^2}g_Y^{(K)}(y_t,t|y_s,s;\boldsymbol{\theta})=o(\Delta^K),\\
&\frac{\partial}{\partial\Delta}g_Y^{(K)}(y_t,t|y_s,s;\boldsymbol{\theta})-\mu_{T_s}(y_s,s;\boldsymbol{\theta})\frac{\partial}{\partial y_s}g_Y^{(K)}(y_t,t|y_s,s;\boldsymbol{\theta})-\frac{1}{2}\frac{\partial^2}{\partial y_s^2}g_Y^{(K)}(y_t,t|y_s,s;\boldsymbol{\theta})=o(\Delta^K).
\end{aligned}\tag{12}$$

Lemma 4. *A real-valued function g(x) is o(x) if* $\lim_{x\to 0}\frac{g(x)}{x}=0$

Since the interest lies with approximating the transition density function of $X_t=x_t$ given $X_s=x_s$. The Jacobian formula in Equation 9, will then be applied to get $g_X^{(k)}(X_t,t|X_s,s;\theta)$ from $g_Y^{(K)}(Y_t,t|Y_s,s;\theta)$ which is obtained as a solution to Equation12. The relation is as follows:

$$g_X^{(K)}(X_t,t|X_s,s;\boldsymbol{\theta})\equiv\frac{1}{\sigma(X_t,t;\theta)}g_Y^{(K)}(\varphi(X_t,t;\boldsymbol{\theta}),t|\varphi(X_s,s;\boldsymbol{\theta}),s;\boldsymbol{\theta}),\tag{13}$$

for all orders of $K\geq 0$, and for $\varphi(x_\tau,\tau;\boldsymbol{\theta})$ a one-to-one function of X_τ at time τ. Order of approximation $K=1$ or $K=2$ is seen as efficient in financial modeling, as in these cases the approximation is quite accurate. Order $K=1$ will usually be used since the amount of extra work and computational power needed to calculate the Hermite-series transition density function approximation for $K=2$ exceeds the added accuracy that the approximation for $K=2$ provides. However increasing the order of approximation by the addition of terms $c_i(y_t,t|y_s,s;\boldsymbol{\theta})$, for $i=1,2,3,4,\ldots$, as given in Equation 11, will still lead to increasing accuracy in the approximation of the transition density function, however the trade-off between the added accuracy and the extra computational time needed should be considered.

2.4.3 Moment truncated saddlepoint transition density function approximation of a scalar diffusion process

Moment truncation is based on the works of [12]. Again, consider the general diffusion process given in Equation 2 under the time-dimension $[s,T]$, where $t\in[s,T]$. The aim is to find a more accurate and

robust closed-form approximation for the transition density function of $X_t = x_t$ given $X_s = x_s$. For the given diffusion process, a sequence of ordinary differential equations (ODEs) for the theoretical moments of the diffusion process, X_t, can be calculated as done in the Thesis of [12]. For $f : \mathbb{R} \to \mathbb{R}$, these moments equations will be in the general form of:

$$m'_j(t) = f(m_{j-1}(t), m_j(t), \boldsymbol{\theta}_j), \tag{14}$$

for $j = 1, 2, 3, ...$

s.t $m_j(s) = X_s^j$ for $j = 1, 2, 3, ...$ where

$E[X_t^j | X_s] = m_j(t)$, for all $t \neq s$, is the $j - th$ theoretical moment of the given diffusion process,

and $\boldsymbol{\theta}_j$ a scalar parameter function.

Through the application of Laplace transforms, where:

$$\mathcal{L}\{m_j(t)\} = M_j(v) = \int_0^\infty e^{-vt} m_j(t) dt,$$

partial fractions and a variety of differential equations methodologies and linear algebra techniques, solutions for the theoretical moments of the process given in the sequence of ODEs in Equation 14 are obtained. The solutions of the ODEs, which are the theoretical moment trajectories of the scalar diffusion process, will be in the general form of:

$$m_j(t) = z(m_{j-1}(t), .., m_1(t), m_0(t), A_j(\alpha, \beta, \sigma)) \tag{15}$$

Where $A_j(\alpha, \beta, \sigma)$ is a scalar parameter function unique to each moment trajectory. From these moments, $m_j(t)$, the cumulants, $K_j(t)$, are calculated for all $j = 1, 2, 3, ...$ Note that $K_j(t)$ denotes the $j - th$ theoretical cumulant. Only the first four moments and cumulants will be utilized in obtaining a closed-form approximation for the conditional transition density function, as this still leads to a very accurate approximation of the conditional transition density function. Increasing the number of moments and cumulants used, does not increase the marginal benefit in terms of accuracy in the approximation. The first four cumulants is calculated from the first four moments through the following relationships:

$$\begin{aligned}
K_1(t) &= m_1(t), \\
K_2(t) &= m_2(t) - (m_1(t))^2, \\
K_3(t) &= 2(m_1(t))^3 - 3(m_1(t))(m_2(t)) + m_3(t), \\
K_3(t) &= -6(m_1(t))^4 + 12(m_1(t))^2(m_2(t)) - 3(m_2(t))^2 - 4(m_1(t))(m_3(t)) + m_4(t).
\end{aligned} \tag{16}$$

Using the cumulants as derived above, and applying the methodology, as provided in [8], the saddlepoint transition density function approximation for a given diffusion process will be derived. Firstly, consider the exact and approximate cumulant generating function respectively:

$$K_X(t) = \sum_{i=1}^{\infty} \frac{1}{i!} t^i K_i(t), \tag{17}$$

$$\widetilde{K}_X(t) = \sum_{i=1}^{N} \frac{1}{i!} t^i K_i(t), \tag{18}$$

for a chosen $N \in \mathbb{N}$.

This paper will choose $N = 4$, since for $N > 4$ the additional computational time needed outweighs the small increase in accuracy. A Taylor series is used to get the approximate cumulant generating function, $\widetilde{K}_{X_t}(t)$, for the given diffusion process. The exact cumulant generating function of the diffusion process is $K_{X_t}(t) = \ln(M_{X_t}(t))$, provided $M_{X_t}(t)$ exists and $M_{X_t}(t) > 0$ for all values of t, where $M_{X_t}(t)$ is the exact moment generating function of the diffusion process under consideration. Next, consider the first and second order partial derivatives , in terms of t:

$$\begin{aligned} \widetilde{K}'_X(t) &= \frac{\partial}{\partial t} \widetilde{K}_X(t), \\ \widetilde{K}''_X(t) &= \frac{\partial^2}{\partial t^2} \widetilde{K}_X(t). \end{aligned} \tag{19}$$

Setting $X_t = \widetilde{K}'_X(t) = \frac{\partial}{\partial t} \widetilde{K}_X(t)$, t is determined as a function of X_t, i.e:

$$t = \varsigma(X_t). \tag{20}$$

Using the result given in [8], and the results obtained in Equation 19 and 20, the closed-form moment-truncated saddlepoint transition density function approximation, $h_X(X_t, t|X_s, s; \boldsymbol{\theta})$, for the theoretical closed-form transition density function of the scalar diffusion process is obtained as:

$$h_X(x_t, t|x_s, s; \boldsymbol{\theta}) = \exp(\widetilde{K}_X(t) - t x_t) \sqrt{(2\pi \widetilde{K}''_X(t))^{-1}}. \tag{21}$$

2.4.4 Hermite-series transition density function approximation compared to the moment-truncated saddlepoint approximation

The work of [16] indicates that the Hermite-series transition density function approximation can only be applied to reducible (i.e $Y_t \to X_t$ is a one-to-one transformation) diffusion processes, although all univariate processes are reducible, not all multivariate diffusion processes are reducible. The Hermite-series transition density function approximation is difficult to implement and there is significant improvement

needed in the accuracy from that provided by the Hermite-series transition density function approximation.

Since a simpler, more general and accurate transition density function approximation is required, the saddlepoint approximation is ideal since it only requires the first few moment trajectories of the given diffusion process. The saddlepoint approximation also seems to be more robust to changes in the underlying parameters.

Although neither the Hermite-series transition density function approximation, nor the moment-truncated saddlepoint approximation integrate to 1, this can be corrected for by normalizing constants.

2.4.5 Maximum likelihood estimation of parameters for a scalar diffusion process

Assuming normality in the residual distribution, for an observed dataset with n observations the likelihood function is given by:

$$L(\boldsymbol{\theta}|\boldsymbol{X}) = \prod_{i=1}^{n}(p_X(X_i, i|X_s, s; \boldsymbol{\theta})), \tag{22}$$

$$\text{for } \boldsymbol{X} = (X_s, ..., X_T),$$

and the log-likelihood function is given by:

$$\ln(L(\boldsymbol{\theta}|\boldsymbol{X})) = \ln\Big\{\prod_{i=1}^{n}(p_X(X_i, i|X_s, s; \boldsymbol{\theta}))\Big\} = \sum_{i=1}^{n}\ln\Big\{(p_X(X_i, i|X_s, s; \boldsymbol{\theta}))\Big\}. \tag{23}$$

To find the maximum likelihood estimators, Equation 23 needs to be maximized, that is $\boldsymbol{\theta}_{max}$ is to be found which maximizes the log-likelihood function. Since diffusion processes have the Markov property the likelihood function can also be defined as,[16]:

$$L(\boldsymbol{\theta}|\boldsymbol{X}) = \prod_{i=1}^{n}(p_X(X_i, i|X_s, s; \boldsymbol{\theta})) = p_X(X_s, s; \boldsymbol{\theta}) \prod_{i=1}^{n}(p_X(X_i, i|X_{i-1}, i-1; \boldsymbol{\theta})$$

3 Generalization of a scalar diffusion process to a mixed-effects diffusion process

3.1 Mixed-effect diffusion process

The dynamics of a general mixed-effects diffusion process are given by the following SDE:

$$dX_t = \mu(X_t, t; \mathring{\boldsymbol{\theta}})dt + \sigma(X_t, t; \mathring{\boldsymbol{\theta}})dW_t, \tag{24}$$

$$\text{s.t } t \in [s, T].$$

X_t is the state variable of interest, $\mu(X_t, t, \mathring{\boldsymbol{\theta}})$ and $\sigma(X_t, t, \mathring{\boldsymbol{\theta}})$ are time- and state dependent drift and diffusion coefficients respectively, where $\mathring{\boldsymbol{\theta}} \equiv (\alpha, \beta, \mathring{\sigma})$ is the parameter vector consisting of α and β as the scalar parameters and $\mathring{\sigma}$ as the random effect. In the present paper it is assumed that $\mathring{\sigma} \sim N(\nu, \varrho^2)$. Therefore the parameter vector can also be written as:

$$\boldsymbol{\delta} = (\alpha, \beta, \nu, \varrho) \equiv \mathring{\boldsymbol{\theta}} = (\alpha, \beta, \mathring{\sigma}).$$

Note that any parameter in the parameter vector can be made a random effect following any probability distribution; $\mathring{\sigma} \sim N(\nu, \varrho^2)$ was chosen for simplicity and to illustrate the affect of a random diffusion in a diffusion process.

3.2 Transition density function approximations of a mixed-effects diffusion processes

Only a generalized version of the moment-truncated saddlepoint approximation will be derived and applied for inferential purposes.

3.2.1 Simulated trajectory for a mixed-effects diffusion process

The trajectory of the mixed-effects diffusion process given in Equation 6 is simulated $M \in \mathbb{N}$ times. Each simulation takes on a new simulated value of $\mathring{\sigma}$ from a $N(\nu, \varrho^2)$ distribution. The average of the M simulated trajectories are then investigated. As M increase the average trajectory smooths out on the mean-reverted value of β.

3.2.2 Euler-Maruyama scheme for a mixed-effects diffusion process

The Euler-Maruyama scheme as applied to the scalar diffusion process as given in Equation 6, is repeated $M \in \mathbb{N}$ times. Each iteration takes on a new simulated value of $\mathring{\sigma}$ from a $N(\nu, \varrho^2)$ distribution. Observing the average over all the Euler-Maruyama simulations leads to the conclusion that as M increase, the density of observing $X_t = \beta$ (the mean value to which the process reverts) also increases.

3.2.3 Moment truncated saddlepoint transition density function approximation of a scalar diffusion process

Consider the general mixed-effects diffusion process given in Equation 24 under the time-dimension $[s, T]$, where $t \in [s, T]$. The methodology is similar to the saddlepoint transition density function approximation

derived for the scalar diffusion process, with the clear exception that σ, which was fixed, now becomes$\mathring{\sigma}$ which is a random effect. The moments of $\mathring{\sigma}$ will now become of importance.

$$m_j'(t) = f(m_{j-1}(t), m_j(t), \mathring{\boldsymbol{\theta}}_j) \tag{25}$$

$$\text{for } j = 1, 2, 3, ...$$

$$\text{s.t } m_j(s) = X_s^j \text{ for } j = 1, 2, 3, ... \text{ and}$$

$$\text{s.t. } E[X_t^j | X_s] = m_j(t), \text{ for all } t \neq s \text{ , is the } j - th \text{ moment of the diffusion process,}$$

$$\text{and } \mathring{\boldsymbol{\theta}}_j \text{ a mixed-effects parameter function for moment ODE.}$$

Through the application of Laplace transforms, where $\mathcal{L}\{m_j(t)\} = M_j(v) = \int_0^\infty e^{-vt} m_j(t) dt$, partial fractions and a variety of differential equations methodologies and linear algebra techniques, solutions for the theoretical moments of the process given in the sequence of ODEs in Equation 25 are obtained. The solutions of the ODEs, which are the moment trajectories of the mixed-effects diffusion process, will be in a general form of:

$$m_j(t) = z(m_{j-1}(t), .., m_1(t), m_0(t), B_{jj}(\alpha, \beta, \mathring{\sigma}), ..., B_{11}(\alpha, \beta, \mathring{\sigma})) \tag{26}$$

The coefficients $B_{jj}(\alpha, \beta, \mathring{\sigma})$ are a unique functions for each moment trajectory, where:

$$B_{jj} = \mathbb{E}_{\mathring{\sigma}}[A_j(\alpha, \beta, \mathring{\sigma})] = \zeta(\alpha, \beta, \nu, \varrho^2), \tag{27}$$

$$\text{where } \mathring{\sigma} \sim N(\nu, \varrho^2) \text{ in the present paper.}$$

From these moments, $m_j(t)$, the cumulants, $K_j(t)$, are calculated for all $j = 1, 2, 3, ...$ Note that $K_j(t)$ denotes the $j - th$ theoretical cumulant. Only the first four moments and cumulants will be utilized in this document in obtaining a closed-form approximation for the conditional transition density function:

$$\begin{aligned}
K_1(t) &= m_1(t), \\
K_2(t) &= m_2(t) - (m_1(t))^2, \\
K_3(t) &= 2(m_1(t))^3 - 3(m_1(t))(m_2(t)) + m_3(t), \\
K_3(t) &= -6(m_1(t))^4 + 12(m_1(t))^2(m_2(t)) - 3(m_2(t))^2 - 4(m_1(t))(m_3(t)) + m_4(t).
\end{aligned} \tag{28}$$

Using the cumulants as derived above and applying the methodology, as provided in [8], the saddlepoint transition density function approximation for a given diffusion process will be derived. Firstly, consider

the exact and approximate cumulant generating function respectively:

$$K_X(t) = \sum_{i=1}^{\infty} \frac{1}{i!} t^i K_i(t), \tag{29}$$

$$\widetilde{K}_X(t) = \sum_{i=1}^{N} \frac{1}{i!} t^i K_i(t), \tag{30}$$

for a chosen $N \in \mathbb{N}$.

The exact cumulant generating function of the diffusion process is $K_{X_t}(t) = \ln(M_{X_t}(t))$, provided $M_{X_t}(t)$ exists and $M_{X_t}(t) > 0$ for all values of t, where $M_{X_t}(t)$ is the exact moment generating function of the diffusion process. Next consider the first and second order partial derivatives , in terms of t:

$$\begin{aligned} \widetilde{K}'_X(t) &= \frac{\partial}{\partial t} \widetilde{K}_X(t), \\ \widetilde{K}''_X(t) &= \frac{\partial^2}{\partial t^2} \widetilde{K}_X(t). \end{aligned} \tag{31}$$

Setting $X_t = \widetilde{K}'_X(t) = \frac{\partial}{\partial t} \widetilde{K}_X(t)$, t is determined as a function of X_t, i.e:

$$t = \varpi(X_t). \tag{32}$$

Using the result given in [8], and the results obtained in Equation 31 and 32, the closed-form saddlepoint transition density function approximation, $h_X(X_t, t|X_s, s; \mathring{\boldsymbol{\theta}})$, for the theoretical closed-form transition density function of the mixed-effects diffusion process is obtained as:

$$h_X(x_t, t|x_s, s; \mathring{\boldsymbol{\theta}}) = \exp(\widetilde{K}_X(t) - t x_t) \sqrt{(2\pi \widetilde{K}''_X(t))^{-1}}. \tag{33}$$

$$\text{s.t } \mathring{\boldsymbol{\theta}} = (\alpha, \beta, \mathring{\sigma}), \text{ where } \mathring{\sigma} \sim N(\nu, \varrho^2).$$

3.2.4 Maximum likelihood estimation of parameters for a mixed-effects diffusion process

Assuming normality in the residual distribution, let

$$\boldsymbol{\delta} = (\alpha, \beta, \nu, \varrho),$$

for an observed dataset with n observations the likelihood function is given by:

$$L(\boldsymbol{\delta}|\boldsymbol{X}) = \prod_{i=1}^{n} (p_X(X_i, i|X_s, s; \boldsymbol{\delta})) \tag{34}$$

$$\text{for } \boldsymbol{X} = (X_s, ..., X_T),$$

$$\boldsymbol{\delta} = (\alpha, \beta, \nu, \varrho),$$

where ν and ϱ are the mean and standard deviation of $\mathring{\sigma}$ respectively.

The log-likelihood function is given by:

$$\ln(L(\boldsymbol{\delta}|\boldsymbol{X})) = \ln\Big\{\prod_{i=1}^{n}(p_X(X_i, i|X_s, s; \boldsymbol{\delta}))\Big\} = \sum_{i=1}^{n}\ln\Big\{(p_X(X_i, i|X_s, s; \boldsymbol{\delta}))\Big\} \tag{35}$$

To find the maximum likelihood estimators, Equation 35 needs to be maximized, that is $\boldsymbol{\delta}_{max} = (\alpha_{max}, \beta_{max}, \nu_{max}, \rho_{max})$ is to be found which maximizes the log-likelihood function. Since diffusion processes have the Markov property the likelihood function can also be defined as, [16]:

$$L(\boldsymbol{\delta}|\boldsymbol{X}) = \prod_{i=1}^{n}(p_X(X_i, i|X_s, s; \boldsymbol{\delta})) = p_X(X_s, s; \boldsymbol{\delta})\prod_{i=1}^{n}(p_X(X_i, i|X_{i-1}, i-1; \boldsymbol{\delta})$$

4 Application of the derived methods on the CIR process

A scalar CIR process will firstly be considered, where all parameters of the parameter vector are fixed effects or scalars. The scalar process has an analytical solution to the true transition density function; this true transition density function will be used as reference point for the derived approximations' accuracy. This scalar CIR process will then be generalized to a mixed-effects CIR process, where the parameter vector consists of fixed effects and one random effect, hence the name "mixed-effects" process. Since an analytical solution to the true transition density function of the mixed-effects process is unattainable, the derived approximations are of fundamental importance.

4.1 Application to the scalar CIR process

As in [4], consider Cox, Ingersoll and Ross (CIR) scalar diffusion process, which is a widely used diffusion process in short-rate modeling, with dynamics given by the following stochastic differential equation :

$$dX_t = \alpha(\beta - X_t)dt + \sigma\sqrt{X_t}dW_t \tag{36}$$

s.t. $[s, T]$ is the time-dimension where $t \in [s, T]$ and $s \geq 0$, and $[X_s, X_T]$ the state-space, with X_s as the initial state, and parameter vector $\boldsymbol{\theta} = (\alpha, \beta, \sigma)$, where α, β, σ are scalars or deterministic parameters. X_t is a stochastic process, representing the value of a short-rate at time t. $\alpha(\beta - X_t)$ is the drift-term and $\sigma\sqrt{X_t}$ the diffusion, with α often being referred to as the speed of the diffusion process. [1] indicates

that the CIR model is an ideal short-rate model when X_t has domain $D_{X_t} = (0, \infty)$. According to [4], for $\alpha, \beta > 0$, the diffusion process in Equation 36 is equivalent to a continuous-time AR(1) model. Over longer duration it is given that $\lim_{t \to \infty} X_t = \beta$. Hence, if X_t represents the short-rate value at time t and $\alpha, \beta > 0$, then over the long term, this randomly moving short-rate will revert to β. By [4], X_t can eventually become zero if $\sigma^2 > 2\alpha\beta$. Putting the constraint $\sigma^2 \leq 2\alpha\beta$ would make it impossible for X_t to reach zero for all values of $t \in [s, T]$ s.t $s \geq 0$. If $X_s > 0$ then $X_t > 0$ for all $t \geq 0$. Further [4] stated that the model given Equation 36 satisfies the following properties:

1. $X_t \geq 0$ for all $t \geq 0$ especially for $t \in [s, T]$ and $s \geq 0$,
2. if $X_t = 0$ for some $t \in [s, T]$ then X_t can eventually become positive,
3. $VAR(X_t)$ and the value of X_t are directly related, and
4. a steady-state distribution exists for X_t

4.1.1 Parameter adjustment effects on the simulated trajectory and transition density function of the CIR process

Before the relevant parameter vector is chosen for this paper, the effects of various parameter values will briefly be indicated. The code for the output in Figure 1 is provided in Algorithm 2.
The code for the output in Figure 2 is provided in Algorithm 3.
In the works of [16] it is stated that the CIR process has the mean-reversion property, and the value of the process will be positive if $\sigma^2 < 2\alpha\beta$. These properties is suiting for short-rate modeling. From both Figure 1 and Figure 2 it is clear that α is the speed of the reversion of the short-rate towards the mean-reverted value of β. Furthermore σ is related to the volatility in the range of values X_t takes on over time; the density becomes more condense towards the mean reverted value of β as σ decrease. For $\alpha < 0$ the short-rate diverges at a fast rate.

4.1.2 Parameter values to be used in the present paper for the scalar CIR process

Throughout the following sections the parameter values and domains used to analyze the **scalar** CIR process in Equation 36, are given by:

- the time-dimension $[s, T] = [0, 5]$,
- state-space $[X_s, X_T] = [0, 5]$,
- with $X_s = 2.75$ as the initial state, and
- parameter vector $\boldsymbol{\theta} = (\alpha, \beta, \sigma) = (0.8, 3, 0.25)$.

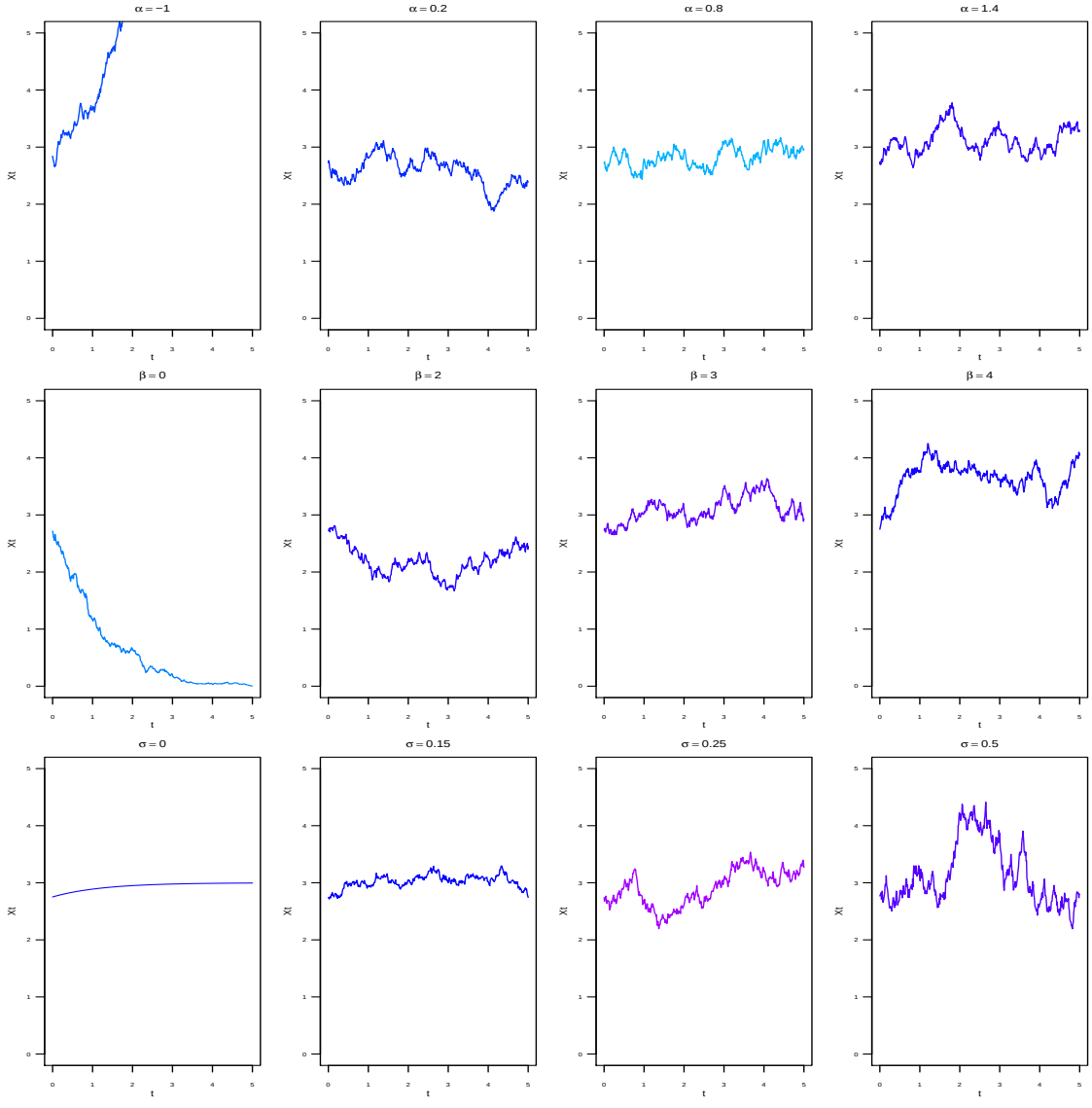


Figure 1: Simulated trajectories for the scalar CIR process for various parameter values.

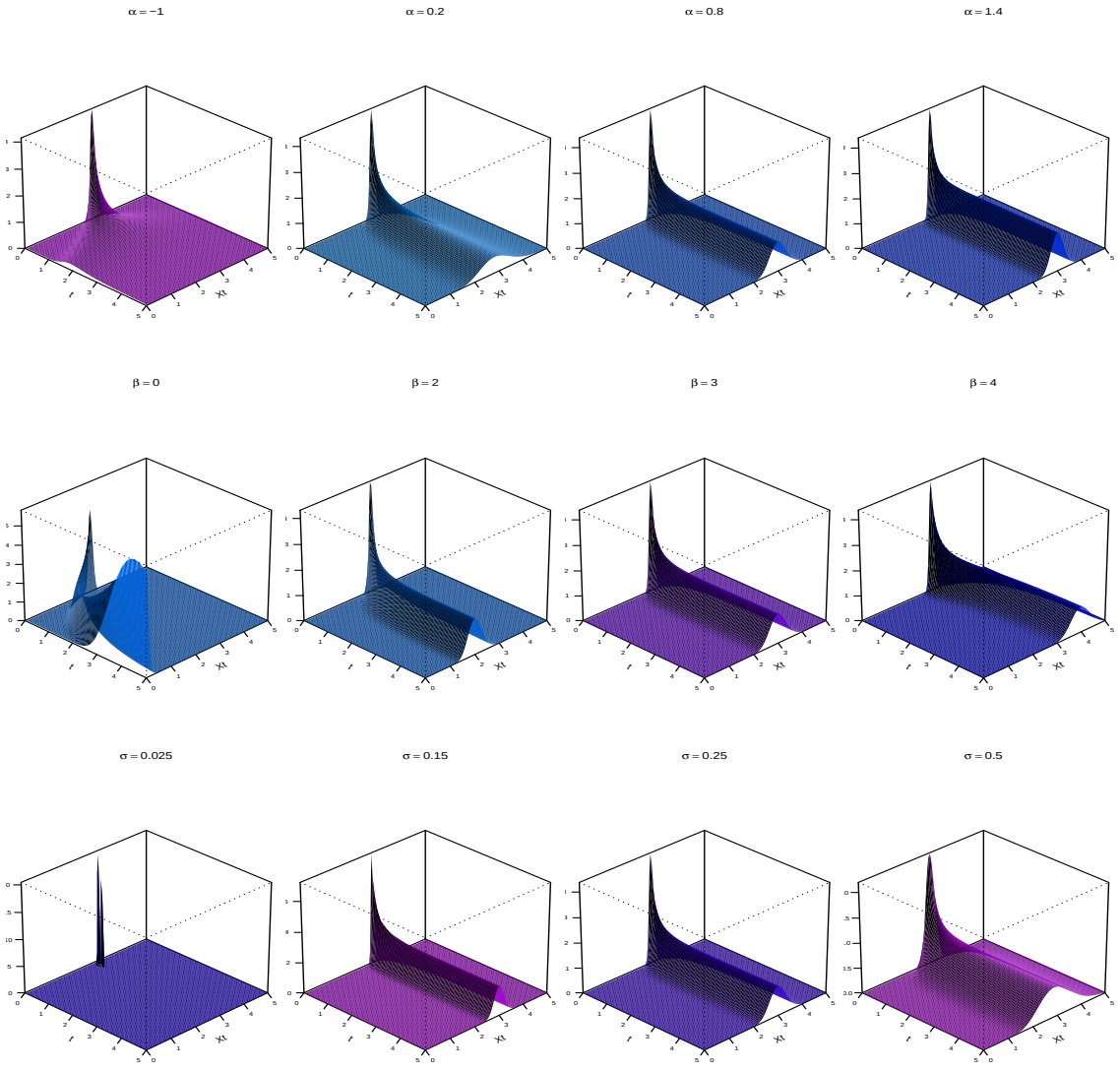


Figure 2: Perspective plots for the scalar CIR process for various parameter values.

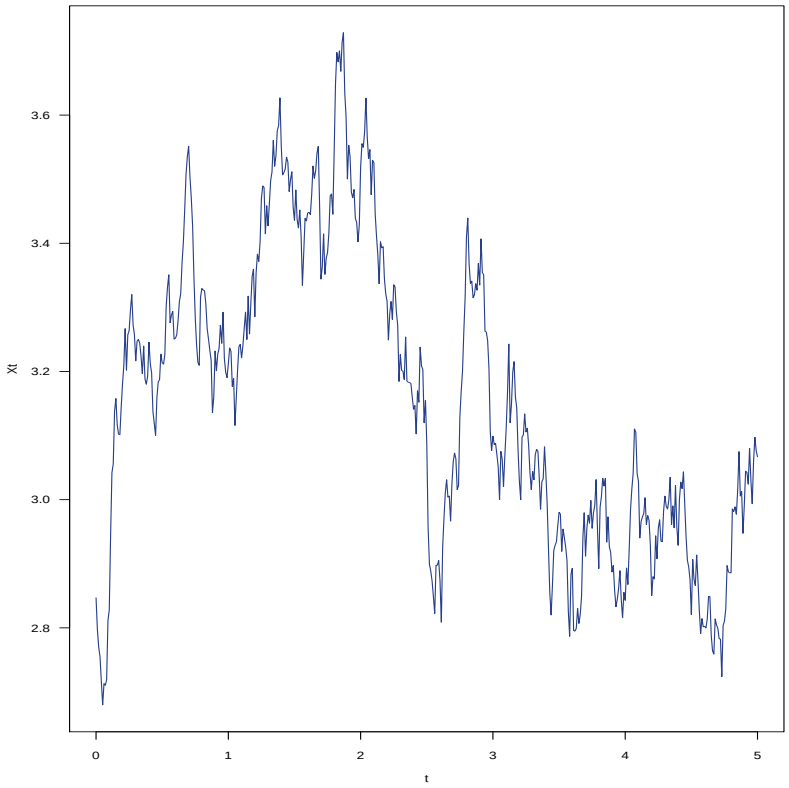


Figure 3: Simulated trajectory of the scalar CIR diffusion process.

The parameter vector for the **mixed-effects** CIR process is chosen to be $\mathring{\boldsymbol{\theta}} = (\alpha, \beta, \mathring{\sigma}) = (0.8, 3, \mathring{\sigma})$, s.t $\mathring{\sigma} \sim N(0.25, 0.15^2)$, i.e $\boldsymbol{\gamma} = (0.8, 3, 0.25, 0.15^2)$.

4.1.3 Simulated trajectory of the scalar CIR process

Algorithm 1 in Appendix B provides the code for plotting the simulated trajectory of the CIR process, as given in Equation 36. Since $\Delta_{W_t} \sim N(0, \Delta_t^2)$, the function **rnorm()** in R is used to simulate these random Brownian motion or Wiener Process values. These values contribute to the stochastic or random part of the trajectory. Let $dt \approx \Delta_t = 0.01$ be the step length in the time-dimension $[0, 7]$. The continuous-time CIR process as given in Equation 36 are discretized by the implementation of the following recursive algorithm:

$$X_{s+\Delta_t} = X_s + \alpha(\beta - X_s)\Delta_t + \sigma\sqrt{X_s}\Delta_{W_t}, \tag{37}$$

and for all $t > s + \Delta_t$:

$$X_{t+1} = X_t + \alpha(\beta - X_t)\Delta_t + \sigma\sqrt{X_t}\Delta_{W_t}. \tag{38}$$

Figure 3 shows the trajectory of the scalar CIR process, i.e as X_t develops over the time-dimension.

Equation 37 and 38 are used to plot this trajectory.

4.1.4 Euler-Maruyama (EM) scheme of the scalar CIR process

To get an idea of the general shape of the transition density function of the scalar CIR process, the Euler-Maruyama scheme, as given in Equation 6, is utilized. Consider the time-dimension $[0, 5]$ s.t $t \in [0, 5]$, the dicretization of the time-dimension is given by: $\Delta_t = \frac{T}{N} = \frac{5}{500} = 0.01$ for $N = 500 \in \mathbb{N}$. The Euler-Maruyama scheme is now given by the following recursive relationship:

$$X_i = X_{i-1} + \alpha(\beta - X_{i-1})\Delta_t + \sigma\sqrt{X_{i-1}}\Delta_{W_i}, \tag{39}$$

for all $i = 1, 2, ..., 499, 500$, and with $X_0 = 2.75$ as initial state, and $\Delta_{W_t} \sim N(0, \Delta_t^2 = 0.01^2)$. The code for the EM scheme is given in Algorithm 1. In Figure 4 the EM scheme, simulated with 100000 trajectories and step-length of 0.01, indicates that the transition density function seems to be very close to a Gaussian/Normal distribution. This gives an indication of the general conditional transition density function of the CIR process's shape. In the next section it will be shown that the actual true transition density function of the scalar CIR process is in fact a non-central Chi-square transition density function.

4.1.5 True transition density function of the scalar CIR process

The true transition density function, $p_X(X_t, t|X_s, s; \boldsymbol{\theta})$, of the scalar CIR process can be obtained by solving the the following forward Kolmogorov equation, as given in Equation 4, [16]:

$$\frac{\partial}{\partial t}p_X(X_t, t|X_s, s; \boldsymbol{\theta}) = -\frac{\partial}{\partial X_t}[\alpha(\beta - X_t)p_X(X_t, t|X_s, s; \boldsymbol{\theta})] + \frac{1}{2}\frac{\partial^2}{\partial X_t^2}[\sigma^2 X_t p_X(X_t, t|X_s, s; \boldsymbol{\theta})]. \tag{40}$$

By solving for $p_X(X_t, t|X_s, s; \boldsymbol{\theta})$, in Equation 40, an analytical solution to the true transition density function for the scalar CIR process, as derived in [4], is provided by:

$$p_X(X_t, t|X_s, s; \boldsymbol{\theta}) = c\exp(-(u + v))\left(\frac{v}{u}\right)^{\frac{q}{2}} I_q(2(uv)^{\frac{1}{2}}), \tag{41}$$

$$\text{where } s < t,$$

$$c = \frac{2\alpha}{\sigma^2(1-\exp(-\alpha\Delta))},$$

$$u = cX_s\exp(-\alpha\Delta),$$

$$\Delta = t - s,$$

$$v = cX_t,$$

$$q = \frac{2\alpha\beta}{\sigma^2} - 1,$$

where $I_q(2(uv)^{\frac{1}{2}})$ is a modified Bessel function of the 1^{st} kind and of the $q-th$ order, with dynamics given by the following ODE:

$$t^2 X_t'' + tX_t' - (t^2 + q^2)X_t = 0,$$

where the solution for this differential equation, derived in [17], is given as:

$$I_q(2(uv)^{\frac{1}{2}}) = (uv)^{\frac{q}{2}} \sum_{k=0}^{\infty} \frac{1}{\Gamma(k+1)\Gamma(q+k+1)}(uv)^k,$$

where q is given as above and $\Gamma(.)$ a gamma function, such that $\Gamma(n) = (n-1)!$. Analysis of the scalar CIR diffusion process in Equation 36, as done by [4], yields that the following function of the time-dependent stochastic process, X_t (short-rate), follows a non-central Chi-squared distribution. That is:

$$2cX_t \sim \chi^2(2q + 2, 2u),$$

with $(2q + 2)$ degrees of freedom, and $2u$ as the parameter of non-centrality. The first- and second order moments for the conditional distribution of X_t, given X_s, are as follows:

$$\mathbb{E}[X_t|X_s] = X_s e^{-\alpha(t-s)} + \beta(1 - e^{-\alpha(t-s)})$$

$$VAR(X_t|X_s) = X_s\left(\frac{\sigma^2}{\alpha}\right)(e^{-\alpha(t-s)} - e^{-2\alpha(t-s)}) + \beta\left(\frac{\sigma}{2\alpha}^2\right)(1 - e^{-\alpha(t-s)})^2[4]$$

Letting t tend to infinity, the steady-state transition density function for the scalar CIR diffusion process is obtained:

$$\lim_{t\to\infty} p_X(x_t, t|x_s, s; \boldsymbol{\theta}) = \frac{\omega^{\upsilon}}{\Gamma(\upsilon)} x^{\upsilon-1} e^{-\omega x},$$

$$\text{where } \omega = \frac{2\alpha}{\sigma^2},$$

$$\upsilon = \frac{2\alpha\beta}{\sigma^2} \ ,$$

$$\lim_{t\to\infty} \mathbb{E}[X_t|X_s] = \beta \text{ and } \lim_{t\to\infty} VAR(X_t|X_s) = \frac{2\alpha\beta}{\sigma^2}.$$

In Figure 4 the true transition density function is shown, with the corresponding code given in Appendix B in Algorithm 1. Letting $s = 0$ and $T = 5$, where $\Delta = T - s = 5$. Although the transition density function closely resembles a Normal/Gaussian transition density function, X_t actually follows a non-central Chi-square or Gamma distribution, as given above. The perspective plot given in Figure 5 indicates with which probability different states (values of X_t e.g. short-rate values) are attained, as time, t, progresses through the time-dimension $[s, T] = [0, 5]$. It is clear that a steady-state distribution is quickly attained and resembles a Gaussian/Normal distribution.

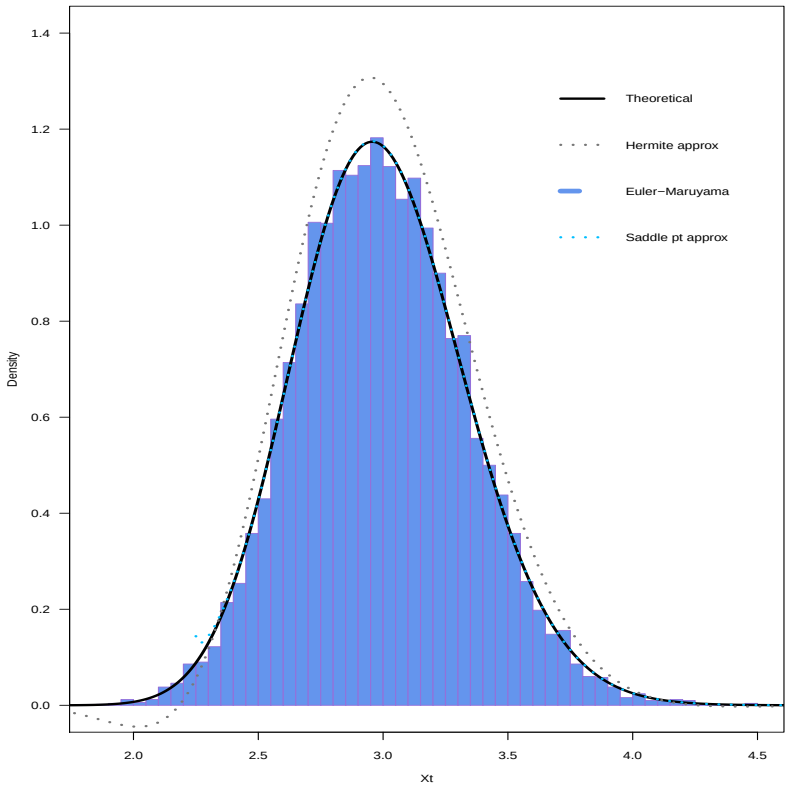


Figure 4: True transition density function, Euler-Maruyama scheme, Hermite-series transition density function approximation for $K = 1$ and the moment-truncated (saddlepoint) approximation of the scalar CIR diffusion process.

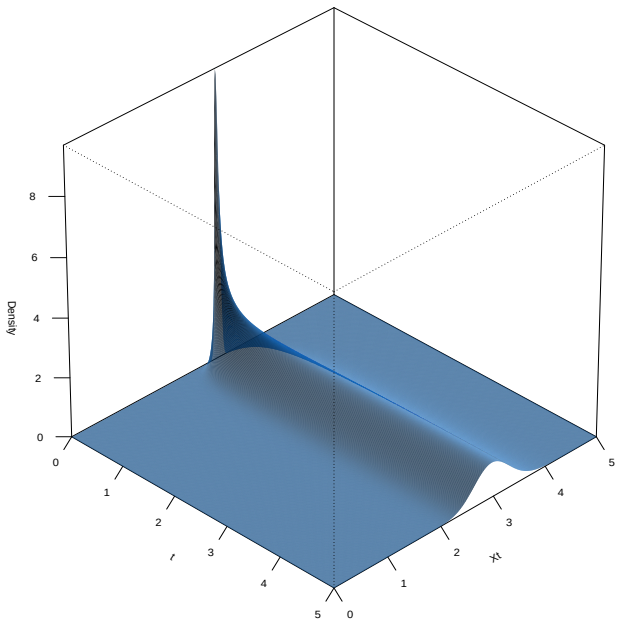


Figure 5: Perspective plot of the transition density function of the scalar CIR diffusion process.

Up to this point the closed-form true transition density function of the scalar CIR diffusion process has been analyzed, but in the case of most diffusion processes a closed-form true transition density function is unattainable. The scalar CIR process was chosen for analysis purposes to give a reference value for the theoretical density function to display the effectiveness and accuracy of the closed-form transition density function approximations that will be discussed in the next sections. When the mixed-effects CIR process, with an unattainable true transition density function, is introduced, the importance of an efficient transition density function approximation will be become clear.

4.1.6 Hermite-series transition density function approximation of the scalar CIR process

Consider the univariate scalar CIR diffusion process given in Equation 37, with the corresponding dynamics and parameters. The Hermite-series transition density function approximation of order $K = 0, 1$ and 2, for the transition density function of the CIR diffusion process, will now be derived, as done by [1]. Firstly the transformation $X_t \to Y_t$ needs to be made such that such that Y_t has a unit diffusion. Therefore consider the transformation:

$$Y_t = \gamma(X_t, t; \boldsymbol{\theta}) = \frac{2\sqrt{X_t}}{\sigma}. \tag{42}$$

It will now be showed that Y_t has unit diffusion. Applying $Itô$'s Lemma as in Equation 94 yields:

$$dY_t = \frac{\partial}{\partial t}\gamma(X_t, t; \boldsymbol{\theta})dt + \frac{\partial}{\partial X_t}\gamma(X_t, t; \boldsymbol{\theta})dX_t + \frac{1}{2}\frac{\partial^2}{\partial X_t^2}\gamma(X_t, t; \boldsymbol{\theta})(dX_t)^2. \tag{43}$$

Taking the derivatives according to $Itô$'s Lemma:

$$\begin{aligned} \frac{\partial}{\partial t}\gamma(X_t, t; \boldsymbol{\theta}) &= 0, \\ \frac{\partial}{\partial X_t}\gamma(X_t, t; \boldsymbol{\theta}) &= \frac{1}{\sigma\sqrt{X_t}}, \\ \frac{\partial^2}{\partial X_t^2}\gamma(X_t, t; \boldsymbol{\theta}) &= -\frac{1}{2\sigma}X_t^{-\frac{3}{2}}. \end{aligned} \tag{44}$$

Since $dX_t = \alpha(\beta - X_t)dt + \sigma\sqrt{X_t}dW_t$, using the result in Remark 26 implies:

$$\begin{aligned} (dX_t)^2 =& (\alpha(\beta - X_t)dt + \sigma\sqrt{X_t}dW_t)^2 \\ =& \sigma^2 X_t (dW_t)^2 \\ =& \sigma^2 X_t dt. \end{aligned} \tag{45}$$

Substituting X_t as given in Equation 36, the partial derivatives in Equation 44 and $(dX_t)^2$ from Equation 45 into Equation 43 the result follows:

$$dY_t = \Big[\frac{\alpha(\beta - X_t)}{\sigma\sqrt{X_t}} - \frac{\sigma}{2\sqrt{X_t}}\Big]dt + dWt. \tag{46}$$

Since

$$\sigma(Y_t, t; \boldsymbol{\theta}) = 1,$$

in Equation 46, Y_t clearly has the unit diffusion, as required.

Let:

- $\Delta = t - s$, where $t \in [s, T]$ for $s \geq 0$,
- specifically set $s = 0$ and $t = 7$ s.t $\Delta = t - s = 7$ and
- consider the transformation $\gamma(x_\tau, \tau; \boldsymbol{\theta}) = y_\tau = \frac{2\sqrt{x_\tau}}{\sigma}$, since the domain of X_τ , for all $\tau \in [s, T]$ is $D_{X_\tau}[0, \infty)$, the transformation is one-to-one. Therefore can the Hermite-series expansion now be applied

The Hermite-series transition density function approximation of order $K \geq 0$, can now be derived using Equation 10. The Hermite-series transition density function approximation for the transition density function of order $K = 0$ will now be derived. The Hermite-series transition density function approximation

of order $K = 0$ for Y_t is given by:

$$g_Y^{(0)}(y_t, t|y_s, s; \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\Delta}} \exp(-\frac{(y_t - y_s)^2}{2\Delta} - \frac{y_t^2\alpha}{4} + \frac{\alpha y_s^2}{4}) \times y_t^{-\frac{1}{2}+\frac{2\alpha\beta}{\sigma^2}} \times y_s^{\frac{1}{2}-\frac{2\alpha\beta}{\sigma^2}}. \tag{47}$$

Using the relation in Equation 13 and the transformation given in Equation 42 the Hermite-series transition density function approximation of order $K = 0$ for X_t is given by:

$$g_X^{(0)}(x_t, t|x_s, s; \boldsymbol{\theta}) \equiv \tfrac{1}{\sigma(x_t,t;\boldsymbol{\theta})} g_Y^{(0)}(\gamma(x_t, t; \boldsymbol{\theta}), t|\gamma(x_s, s; \boldsymbol{\theta}), s; \boldsymbol{\theta}),$$

$$g_X^{(0)}(x_t, t|x_s, s; \boldsymbol{\theta}) = \frac{1}{\sigma\sqrt{x_t}} \frac{1}{\sqrt{2\pi\Delta}} \lambda(x_t, t|x_s, s; \boldsymbol{\theta}) \Psi(x_t, t|x_s, s; \boldsymbol{\theta}), \tag{48}$$

where

$$\lambda(x_t, t|x_s, s; \boldsymbol{\theta}) = \exp(-\frac{(\gamma(x_t, t; \boldsymbol{\theta}) - \gamma(x_s, s; \boldsymbol{\theta}))^2}{2\Delta} - \frac{\gamma^2(x_t, t; \boldsymbol{\theta})\alpha}{4} + \frac{\alpha\gamma^2(x_s, s; \boldsymbol{\theta})}{4})$$

$$\text{and } \Psi(x_t, t|x_s, s; \boldsymbol{\theta}) = \gamma(x_t, t; \boldsymbol{\theta})^{-\frac{1}{2}+\frac{2\alpha\beta}{\sigma^2}} \times \gamma(x_s, s; \boldsymbol{\theta})^{\frac{1}{2}-\frac{2\alpha\beta}{\sigma^2}}.$$

The Hermite-series transition density function approximation for the transition density function of order $K = 1$ will now be derived. The Hermite-series transition density function approximation of order $K = 1$ for Y_t is given by:

$$g_Y^{(1)}(y_t, t|y_s, s; \boldsymbol{\theta}) = g_Y^{(0)}(y_t, t|y_s, s; \boldsymbol{\theta})\{1 + \Delta c_1(y_t, t|y_s, s; \boldsymbol{\theta})\}, \tag{49}$$

where

$$c_1(y_t, t|y_s, s; \boldsymbol{\theta}) = -\tfrac{(48\beta^2\alpha^2 - 48\beta\alpha\sigma^2 + 9\sigma^4 + y_t\alpha^2\sigma^2(-24\beta + y_t^2\sigma^2)y_s + y_t^2\alpha^2\sigma^4 y_s^2 + y_t\alpha^2\sigma^4 y_s^3)}{24 y_t y_s \sigma^4}.$$

Using the relation in Equation 13 and the transformation given in Equation 42 the Hermite-series transition density function approximation of order $K = 1$, for X_t is given by:

$$g_X^{(1)}(x_t, t|x_s, s; \boldsymbol{\theta}) \equiv \tfrac{1}{\sigma(x_t,t;\boldsymbol{\theta})} g_Y^{(1)}(\gamma(x_t, t; \boldsymbol{\theta}), t|\gamma(x_s, s; \boldsymbol{\theta}), s; \boldsymbol{\theta}),$$

$$g_X^{(1)}(x_t, t|x_s, s; \boldsymbol{\theta}) \equiv \tfrac{1}{\sigma\sqrt{x_t}} g_Y^{(0)}(\gamma(x_t, t; \boldsymbol{\theta}), t|\gamma(x_s, s; \boldsymbol{\theta}), s; \boldsymbol{\theta})\{1 + \Delta c_1(\gamma(x_t, t; \boldsymbol{\theta}), t|\gamma(x_s, s; \boldsymbol{\theta}), s; \boldsymbol{\theta})\}.$$

With the Hermite-series transition density function approximation, of order $K = 1$, for X_t, given by:

$$g_X^{(1)}(x_t, t|x_s, s; \theta) = \psi_X(x_t, t|x_s, s; \theta)\{1 + \Delta c_1(\gamma(x_t, t; \boldsymbol{\theta}), t|\gamma(x_s, s; \boldsymbol{\theta}), s; \boldsymbol{\theta})\}, \tag{50}$$

where

$$\psi_X(x_t, t|x_s, s; \theta) = \tfrac{1}{\sigma\sqrt{x_t}} \tfrac{1}{\sqrt{2\pi\Delta}} \exp(-\tfrac{(\gamma(x_t,t;\boldsymbol{\theta}) - \gamma(x_s,s;\boldsymbol{\theta}))^2}{2\Delta} - \tfrac{\gamma(x_t,t;\boldsymbol{\theta})^2\alpha}{4} + \tfrac{\alpha\gamma(x_s,s;\boldsymbol{\theta})^2}{4}) \Psi(x_t, t|x_s, s; \boldsymbol{\theta}),$$

where

$$\Psi(x_t,t|x_s,s;\boldsymbol{\theta}) = \gamma(x_t,t;\boldsymbol{\theta})^{-\frac{1}{2}+\frac{2\alpha\beta}{\sigma^2}} \times \gamma(x_s,s;\boldsymbol{\theta})^{\frac{1}{2}-\frac{2\alpha\beta}{\sigma^2}},$$

and

$$\begin{aligned} c_1(\gamma(x_t,t;\boldsymbol{\theta}),t|\gamma(x_s,s;\boldsymbol{\theta}),s;\boldsymbol{\theta}) = -\frac{1}{24\gamma(x_t,t;\boldsymbol{\theta})\gamma(x_s,s;\boldsymbol{\theta})\sigma^4}\Big[&48\beta^2\alpha^2 - 48\beta\alpha\sigma^2 + 9\sigma^4 \\ &+\gamma(x_t,t;\boldsymbol{\theta})\alpha^2\sigma^2(-24\beta+\gamma(x_t,t;\boldsymbol{\theta})^2\sigma^2)\gamma(x_s,s;\boldsymbol{\theta}) \\ &+\gamma(x_t,t;\boldsymbol{\theta})^2\alpha^2\sigma^4\gamma(x_s,s;\boldsymbol{\theta})^2 + \gamma(x_t,t;\boldsymbol{\theta})\alpha^2\sigma^4\gamma(x_s,s;\boldsymbol{\theta})^3)\Big]. \end{aligned}$$

The Hermite-series transition density function approximation for the transition density function of order $K=2$ will now be derived. The Hermite-series transition density function approximation of order $K=2$, for Y_t is given by:

$$g_Y^{(2)}(y_t,t|y_s,s;\boldsymbol{\theta}) = g_Y^{(0)}(y_t,t|y_s,s;\boldsymbol{\theta})\{1+\Delta c_1(y_t,t|y_s,s;\boldsymbol{\theta}) + \tfrac{\Delta^2}{2}c_2(y_t,t|y_s,s;\boldsymbol{\theta})\}$$

with $c_1(y_t,t|y_s,s;\boldsymbol{\theta})$ as in $K=1$ and with $c_2(y_t,t|y_s,s;\boldsymbol{\theta})$ given as:

$$c_2(y_t,t|y_s,s;\boldsymbol{\theta}) = \delta(y_t,t|y_s,s;\boldsymbol{\theta})\Upsilon(y_t,t|y_s,s;\boldsymbol{\theta})$$

s.t

$$\delta(y_t,t|y_s,s;\boldsymbol{\theta}) = \tfrac{1}{576y_t^2y_s^2\sigma^8}$$

and

$$\begin{aligned} \Upsilon(y_t,t|y_s,s;\boldsymbol{\theta}) =& 9(256(\alpha\beta)^4 - 512(\alpha\beta)^3\sigma^2 + 224(\alpha\beta)\sigma^4 + 32(\alpha\beta)\sigma^6 - 15\sigma^8) \\ &+6y_t\alpha^2\sigma^2(-24\beta+y_t^2\sigma^2)(16\beta^2\alpha^2 - 16\beta\alpha\sigma^2 + 3\sigma^4)y_s \\ &+y_t^2\alpha^2\sigma^4(672\beta^2\alpha^2 - 48\beta\alpha(2+y_t^2\alpha)\sigma^2 + (-6+y_t^4\alpha^2)\sigma^4)y_s^2 \\ &+2y_t\alpha^2\sigma^4(48\beta^2\alpha^2 - 24\beta\alpha(2+y_t^2\alpha)\sigma^2 + (9+y_t^4\alpha^2)\sigma^4)y_s^3 \\ &+3y_t^2\alpha^4\sigma^6(-16\beta+y_t^2\sigma^2)y_s^4 + 2y_t^3\alpha^4\sigma^8y_s^5 + y_t^2\alpha^4\sigma^8y_s^6. \end{aligned}$$

Using the relation in Equation 13 and the transformation given in Equation 42, the Hermite-series transition density function approximation of order $K=2$, for X_t is given by:

$$g_X^{(2)}(x_t,t|x_s,s;\boldsymbol{\theta}) \equiv \tfrac{1}{\sigma(x_t,t;\boldsymbol{\theta})}g_Y^{(2)}(\gamma(x_t,t;\boldsymbol{\theta}),t|\gamma(x_s,s;\boldsymbol{\theta}),s;\boldsymbol{\theta}),$$

$$g_X^{(2)}(x_t,t|x_s,s;\boldsymbol{\theta}) \equiv \frac{1}{\sigma(x_t,t;\boldsymbol{\theta})}g_Y^{(0)}(\gamma(x_t,t;\boldsymbol{\theta}),t|\gamma(x_s,s;\boldsymbol{\theta}),s;\boldsymbol{\theta})\xi(x_t,t|x_s,s;\boldsymbol{\theta}), \tag{51}$$

s.t

$$\xi(x_t,t|x_s,s;\boldsymbol{\theta}) = \{1+\Delta c_1(\gamma(x_t,t;\boldsymbol{\theta}),t|\gamma(x_s,s;\boldsymbol{\theta}),s;\boldsymbol{\theta}) + \tfrac{\Delta^2}{2}c_2(\gamma(x_t,t;\boldsymbol{\theta}),t|\gamma(x_s,s;\boldsymbol{\theta}),s;\boldsymbol{\theta})\},$$

with

$$c_2(y_t, t|y_s, s; \boldsymbol{\theta}) = \eta(y_t, t|y_s, s; \boldsymbol{\theta})\rho(y_t, t|y_s, s; \boldsymbol{\theta}),$$

s.t

$$\eta(y_t, t|y_s, s; \boldsymbol{\theta}) = \frac{1}{576\gamma^2(x_t,t;\boldsymbol{\theta})\gamma^2(x_s,s;\boldsymbol{\theta})\sigma^8},$$

and

$$\begin{aligned}
\rho(y_t, t|y_s, s; \boldsymbol{\theta}) =& 9(256(\alpha\beta)^4 - 512(\alpha\beta)^3\sigma^2 + 224(\alpha\beta)\sigma^4 + 32(\alpha\beta)\sigma^6 - 15\sigma^8) \\
&+6\gamma(x_t, t; \boldsymbol{\theta})\alpha^2\sigma^2(-24\beta + \gamma^2(x_t, t; \boldsymbol{\theta})\sigma^2)(16\beta^2\alpha^2 - 16\beta\alpha\sigma^2 + 3\sigma^4)\gamma(x_s, s; \boldsymbol{\theta}) \\
&+\gamma^2(x_t, t; \boldsymbol{\theta})\alpha^2\sigma^4(672\beta^2\alpha^2 - 48\beta\alpha(2 + \gamma^2(x_t, t; \boldsymbol{\theta})\alpha)\sigma^2 + (-6 + \gamma^4(x_t, t; \boldsymbol{\theta})\alpha^2)\sigma^4)\gamma^2(x_s, s; \boldsymbol{\theta}) \\
&+2\gamma(x_t, t; \boldsymbol{\theta})\alpha^2\sigma^4(48\beta^2\alpha^2 - 24\beta\alpha(2 + \gamma^2(x_t, t; \boldsymbol{\theta})\alpha)\sigma^2 + (9 + \gamma^4(x_t, t; \boldsymbol{\theta})\alpha^2)\sigma^4)\gamma^3(x_s, s; \boldsymbol{\theta}) \\
&+3\gamma^2(x_t, t; \boldsymbol{\theta})\alpha^4\sigma^6(-16\beta + \gamma^2(x_t, t; \boldsymbol{\theta})\sigma^2)\gamma^4(x_s, s; \boldsymbol{\theta}) \\
&+2\gamma^3(x_t, t; \boldsymbol{\theta})\alpha^4\sigma^8\gamma^5(x_s, s; \boldsymbol{\theta}) + \gamma^2(x_t, t; \boldsymbol{\theta})\alpha^4\sigma^8\gamma^6(x_s, s; \boldsymbol{\theta}).
\end{aligned}$$

In Figure 6, the Hermite-series transition density function approximation of order $K = 0, 1, 2$, for the CIR process is plotted against the true transition density function, with the corresponding code given in Algorithm 4. The Hermite-series transition density function approximation for the CIR process for $K = 0, 1, 2$ closely resembles a Gaussian density. For K small, according to [1], the asymptotic result $\lim_{K\to\infty} |g_X - g_X^{(K)}| = 0$ follows. It is clear that the Hermite-series transition density function approximation, of order $K = 0$, is far from accurate. For $K = 1$ and $K = 2$ the Hermite-series transition density function approximations looks almost identical, this is due to the fact that $c_2(\gamma(x_t, t; \boldsymbol{\theta}), t|\gamma(x_s, s; \boldsymbol{\theta}), s; \boldsymbol{\theta}) \approx 0$, this is illustrated in Figure 6. Therefore for $K = 2$ the added accuracy is outweighed by the extra the computational power, and therefore the Hermite-series transition density function approximation for the CIR process of order $K = 1$ is sufficient and is also plotted for comparison in Figure 4. It should be noted from Figure 6 that the Hermite-series transition density function approximations for $K \geq 1$ are still far from the true transition density function of the CIR process, this leads to the need to obtain a more efficient and accurate approximation for the CIR process's transition density function. This improved approximation is the moment-truncated saddlepoint transition density function approximation, which will now be discussed in the next section.

4.1.7 Moment truncated saddlepoint transition density function approximation of the scalar CIR process

Since a more accurate and efficient closed-form approximation for the transition density function of the scalar CIR process is required, the moment-truncated saddlepoint transition density function approximation of the scalar CIR process, based on the techniques of [12], will now be derived. Again consider the

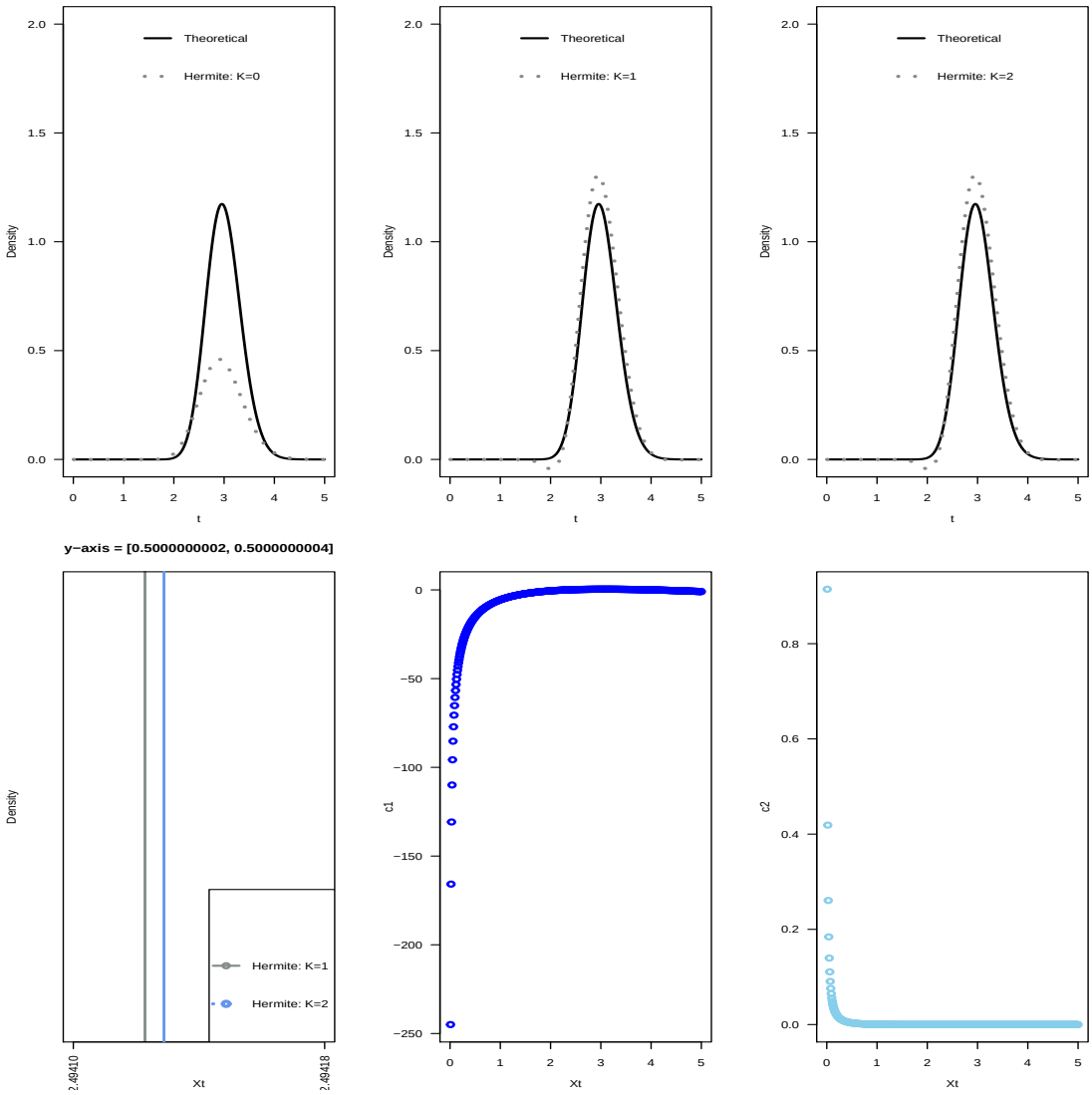


Figure 6: True transition density function with an overlayed Hermite-series transition density function approximation, of order $K = 0, 1, 2$, for the scalar CIR process. It is also shown how insignificant the improvement in approximation-accuracy is from $K = 1$ to $K = 2$ in the Hermite-series approximation, due to $c2$ being significantly close to 0.

scalar CIR process, with dynamics and parameters as given in Equation 36, with time-dimension $[s,T]$ s.t $t \in [s,T]$. Letting $s=0$ and $T=5$ s.t $\Delta = T-s = 5$. The sequence of moment equations in the form of ordinary differential equations, as in Equation 14, is given by:

$$\begin{aligned} m_1^{'}(t) &= 1(\alpha\beta - \alpha m_1(t)), \\ m_2^{'}(t) &= 2(\alpha\beta m_1(t) - \alpha m_2(t)) + \sigma^2 m_1(t), \\ m_3^{'}(t) &= 3(\alpha\beta m_2(t) - \alpha m_3(t)) + 3\sigma^2 m_2(t), \\ m_4^{'}(t) &= 4(\alpha\beta m_3(t) - \alpha m_4(t)) + 6\sigma^2 m_3(t). \end{aligned} \tag{52}$$

where $E[X_t^j|X_s] = m_j(t)$, for all $t \neq s$, is the $j-th$ moment of the given diffusion process, and $m_j(s) = X_s^j$, for $j = 1,2,3...$ Through the application of Laplace transforms, where $\mathcal{L}\{m_j(t)\} = M_j(v) = \int_0^\infty e^{-vt} m_j(t)dt$, and $\mathcal{L}^{-1}\{M_j(v)\} = m_j(t)$, partial fractions and various differential and linear algebra techniques, solutions to the ODEs in Equation 52 will now be derived. Consider:

$$m_1^{'}(t) = 1(\alpha\beta - m_1(t)). \tag{53}$$

Applying the Laplace transform throughout Equation 53:

$$\mathcal{L}\{m_1^{'}(t)\} = \mathcal{L}\{\alpha\beta - m_1(t)\},$$

$$vM_1(v) - m_1(s) = \alpha\beta - \alpha M_1(\nu),$$

$$\text{s.t } m_1(s) = X_s.$$

Using partial fractions and simplifying:

$$M_1(v) = \tfrac{1}{v}\beta + X_s\tfrac{1}{\alpha+v} - \beta\tfrac{1}{\alpha+v}.$$

Applying the inverse Laplace transform:

$$\mathcal{L}^{-1}\{M_1(v)\} = \mathcal{L}^{-1}\{\tfrac{1}{v}\beta + X_s\tfrac{1}{\alpha+v} - \beta\tfrac{1}{\alpha+v}\}.$$

The solution to the ODE in Equation 53 immediately follows:

$$E[X_t|X_s] = m_1(t) = X_s e^{-\alpha t} + \beta(1 - e^{-\alpha t}).$$

Consider:

$$m_2^{'}(t) = 2(\alpha\beta m_1(t) - \alpha m_2(t)) + \sigma^2 m_1(t). \tag{54}$$

Applying the Laplace transform throughout Equation 54:

$$\mathcal{L}\{m_2^{'}(t)\} = \mathcal{L}\{2(\alpha\beta m_1(t) - \alpha m_2(t)) + \sigma^2 m_1(t)\},$$

$$vM_2(v) - m_2(s) = (2\alpha\beta + \sigma^2)M_1(v) - 2\alpha M_2(v),$$

$$\text{s.t } m_2(s) = X_s^2.$$

Using partial fractions and simplifying:

$$M_2(v) = \beta(\beta + \tfrac{\sigma^2}{2\alpha})\tfrac{1}{v} + 2(\beta + \tfrac{\sigma^2}{2\alpha})(X_s - \beta)\tfrac{1}{v+\alpha} + \Big[X_s^2 + (\beta + \tfrac{\sigma^2}{2\alpha})(\beta - 2X_s)\Big]\tfrac{1}{v+2\alpha}.$$

Applying the inverse Laplace transform:

$$\mathcal{L}^{-1}\{M_2(v)\} = \mathcal{L}^{-1}\{\beta(\beta + \tfrac{\sigma^2}{2\alpha})\tfrac{1}{v} + 2(\beta + \tfrac{\sigma^2}{2\alpha})(X_s - \beta)\tfrac{1}{v+\alpha} + \Big[X_s^2 + (\beta + \tfrac{\sigma^2}{2\alpha})(\beta - 2X_s)\Big]\tfrac{1}{v+2\alpha}\}.$$

The solution to the ODE in Equation 54 immediately follows:

$$E[X_t^2|X_s] = m_2(t) = X_s^2 e^{-2\alpha t} + (\beta + \tfrac{\sigma^2}{2\alpha})(\beta + 2(X_s - \beta)e^{-\alpha t} + (\beta - 2X_s)e^{-2\alpha t}).$$

Consider

$$m_3^{'}(t) = 3(\alpha\beta m_2(t) - \alpha m_3(t)) + 3\sigma^2 m_2(t). \tag{55}$$

Applying the Laplace transform throughout Equation 55:

$$\mathcal{L}\{m_3^{'}(t)\} = \mathcal{L}\{3(\alpha\beta m_2(t) - \alpha m_3(t)) + 3\sigma^2 m_2(t)\},$$

$$vM_1(v) - m_2(s) = 3(\alpha\beta + \sigma^2)M_2(v) - 3\alpha M_3(v),$$

$$\text{s.t } m_3(s) = X_s^3.$$

Using partial fractions and simplifying:

$$M_3(v) = X_s^3\tfrac{1}{v+3\alpha} + 3(\alpha\beta + \sigma^2)\Big[A\tfrac{1}{v} + B\tfrac{1}{v+\alpha} + C\tfrac{1}{v+2\alpha} + D\tfrac{1}{v+3\alpha}\Big],$$

where:

$$\begin{aligned}
A =& \frac{\beta(\beta + \frac{\sigma^2}{2\alpha})}{3\alpha},\\
C =& -4(\frac{1}{4\alpha^2}(\Upsilon - 9\alpha^2 A) - \frac{1}{2\alpha}(\Phi - 3\alpha A)),\\
\Upsilon =& \alpha(X_s^2 + (\beta + \frac{\sigma^2}{2\alpha})(\beta - 2X_s)) + 3\alpha\beta(\beta + \frac{\sigma^2}{2\alpha}) + 4\alpha(\beta + \frac{\sigma^2}{2\alpha})(X_s - \beta),\\
\Phi =& X_s^2 + (\beta + \frac{\sigma^2}{2\alpha})(\beta - 2X_s) + \beta(\beta + \frac{\sigma^2}{2\alpha}) + 2(\beta + \frac{\sigma^2}{2\alpha})(X_s - \beta),\\
B =& \frac{\Phi - 3\alpha\Phi - \alpha C}{2\alpha},\\
D =& -A - B - C.
\end{aligned}$$

Applying the inverse Laplace transform:

$$\mathcal{L}^{-1}\{M_3(v)\} = \mathcal{L}^{-1}\{X_s^3\tfrac{1}{v+3\alpha} + 3(\alpha\beta + \sigma^2)\Big[A\tfrac{1}{v} + B\tfrac{1}{v+\alpha} + C\tfrac{1}{v+2\alpha} + D\tfrac{1}{v+3\alpha}\Big]\}.$$

The solution to the ODE in Equation 55 immediately follows:

$$E[X_t^3|X_s] = m_3(t) = X_s^3 e^{-3\alpha t} + 3(\alpha\beta + \sigma^2)(A + Be^{-\alpha t} + Ce^{-2\alpha t} + De^{-3\alpha t}).$$

Consider:

$$m_4^{'}(t) = 4(\alpha\beta m_3(t) - \alpha m_4(t)) + 6\sigma^2 m_3(t). \tag{56}$$

Applying the Laplace transform throughout Equation 56:

$$\mathcal{L}\{m_4^{'}(t)\} = \mathcal{L}\{4(\alpha\beta m_3(t) - \alpha m_4(t)) + 6\sigma^2 m_3(t)\},$$

$$\upsilon M_4(v) - m_4(s) = (4\alpha\beta + 6\sigma^2)M_3(v) - 4\alpha M_4(v),$$

$$\text{s.t } m_4(s) = X_s^4.$$

Using partial fractions and simplifying:

$$M_3(v) = X_s^4 \tfrac{1}{v+4\alpha} + (4\alpha\beta + 6\sigma^2)\Big[E\tfrac{1}{v} + F\tfrac{1}{v+\alpha} + G\tfrac{1}{v+2\alpha} + H\tfrac{1}{v+3\alpha} + I\tfrac{1}{v+4\alpha}\Big],$$

where:

$$\begin{aligned}
E =& \frac{\nu^*}{24\alpha^4},\\
I =& -\frac{1}{6\alpha^3}\bigg(\Big(\Omega^* - \frac{13}{12}\frac{\nu^*}{\alpha}\Big) - 12\alpha^2\Big(\gamma^* - \frac{\nu^*}{24\alpha^3}\Big) - 4\alpha\Big(\Big(\lambda^* - \frac{3}{8}\frac{\nu^*}{\alpha}\Big) - 7\alpha\Big(\gamma^* - \frac{\nu^*}{24\alpha^3}\Big)\Big)\bigg),\\
H =& \frac{1}{2\alpha^2}\bigg(\Big(\lambda^* - \frac{3}{8}\frac{\nu^*}{\alpha}\Big) - 7\alpha\Big(\gamma^* - \frac{\nu^*}{24\alpha^3}\Big) - 6\alpha^2 I\bigg),\\
F =& -E - G - H - I,\\
\nu^* =& 3(\alpha\beta + \sigma^2)(6\alpha^3 A),\\
\gamma^* =& 5\alpha X_s^3 + 3(\alpha\beta + \sigma^2)\big[6\alpha A + 5\alpha B + 4\alpha C + 3\alpha D\big],\\
\lambda^* =& 5\alpha X_s^3 + 3(\alpha\beta + \sigma^2)\big[6\alpha A + 5\alpha B + 4\alpha C + 3\alpha D\big],\\
\Omega^* =& 4\alpha^2 X_s^3 + 3(\alpha\beta + \sigma^2)\big[11\alpha^2 A + 6\alpha^2 B + 3\alpha^2 C + 2\alpha^2 D\big],\\
A =& \frac{\beta(\beta + \frac{\sigma^2}{2\alpha})}{3\alpha},\\
C =& -4(\frac{1}{4\alpha^2}(\Upsilon - 9\alpha^2 A) - \frac{1}{2\alpha}(\Phi - 3\alpha A)),\\
\Upsilon =& \alpha(X_s^2 + (\beta + \frac{\sigma^2}{2\alpha})(\beta - 2X_s)) + 3\alpha\beta(\beta + \frac{\sigma^2}{2\alpha}) + 4\alpha(\beta + \frac{\sigma^2}{2\alpha})(X_s - \beta),\\
\Phi =& X_s^2 + (\beta + \frac{\sigma^2}{2\alpha})(\beta - 2X_s) + \beta(\beta + \frac{\sigma^2}{2\alpha}) + 2(\beta + \frac{\sigma^2}{2\alpha})(X_s - \beta),\\
B =& \frac{\Phi - 3\alpha\Phi - \alpha C}{2\alpha},\\
D =& -A - B - C.
\end{aligned}$$

Applying the inverse Laplace transform:

$$\mathcal{L}^{-1}\{M_4(v)\} = \mathcal{L}^{-1}\{X_s^4 \tfrac{1}{v+4\alpha} + (4\alpha\beta + 6\sigma^2)\Big[E\tfrac{1}{v} + F\tfrac{1}{v+\alpha} + G\tfrac{1}{v+2\alpha} + H\tfrac{1}{v+3\alpha} + I\tfrac{1}{v+4\alpha}\Big]\}.$$

The solution to the ODE in Equation 56 immediately follows:

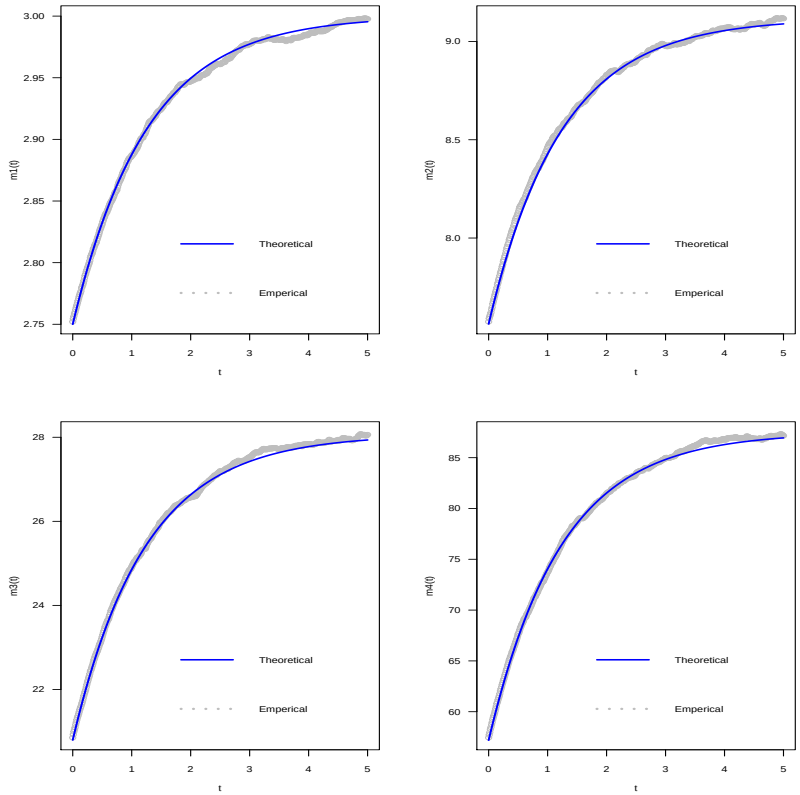


Figure 7: The first four theoretical and empirical moments of the scalar CIR process.

$$E[X_t^4|X_s] = m_4(t) = X_s^4 e^{-4\alpha t} + (4\alpha\beta + 6\sigma^2)\Big[E + Fe^{-\alpha t} + Ge^{-2\alpha t} + He^{-3\alpha t} + Ie^{-4\alpha t}\Big].$$

Therefore the first four theoretical moments of the scalar CIR process, given as the solutions of the ODEs in Equation 52, is as follows:

$$\begin{aligned}
E[X_t|X_s] =& m_1(t) = X_s e^{-\alpha t} + \beta(1 - e^{-\alpha t}),\\
E[X_t^2|X_s] =& m_2(t) = X_s^2 e^{-2\alpha t} + (\beta + \frac{\sigma^2}{2\alpha})(\beta + 2(X_s - \beta)e^{-\alpha t} + (\beta - 2X_s)e^{-2\alpha t}),\\
E[X_t^3|X_s] =& m_3(t) = X_s^3 e^{-3\alpha t} + 3(\alpha\beta + \sigma^2)(A + Be^{-\alpha t} + Ce^{-2\alpha t} + De^{-3\alpha t}),\\
E[X_t^4|X_s] =& m_4(t) = X_s^4 e^{-4\alpha t} + (4\alpha\beta + 6\sigma^2)\Big[E + Fe^{-\alpha t} + Ge^{-2\alpha t} + He^{-3\alpha t} + Ie^{-4\alpha t}\Big].
\end{aligned} \tag{57}$$

with A, B, C, D, E, F, G, H and I as above. In Figure 7, the theoretical moments of the scalar CIR process, given in Equation 57, together with the empirical moments are plotted with the corresponding code in Algorithm 5.

However to derive a saddlepoint transition density function approximation for the transition density function of the scalar CIR process, it is ideal to derive and utilize the theoretical cumulants. Let $K_j(t)$ denote the $j-th$ cumulant of the scalar CIR process. Therefore the first four cumulants of the of the

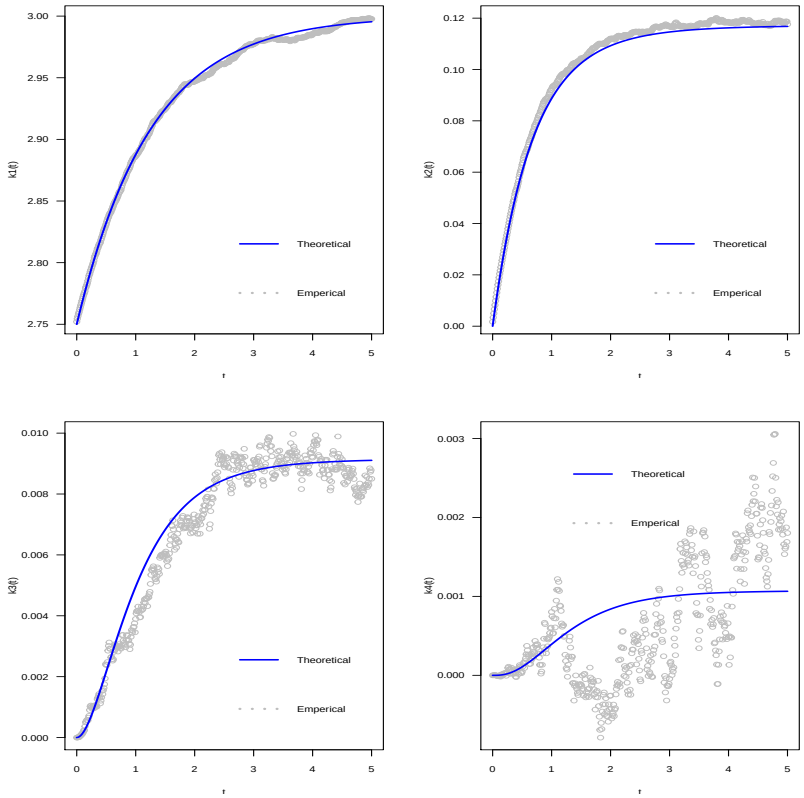


Figure 8: The first four theoretical and empirical cumulants of the scalar CIR process.

scalar CIR process are given by:

$$\begin{aligned} K_1(t) &= m_1(t), \\ K_2(t) &= m_2(t) - (m_1(t))^2, \\ K_3(t) &= 2(m_1(t))^3 - 3(m_1(t))(m_2(t)) + m_3(t), \\ K_3(t) &= -6(m_1(t))^4 + 12(m_1(t))^2(m_2(t)) - 3(m_2(t))^2 - 4(m_1(t))(m_3(t)) + m_4(t). \end{aligned} \tag{58}$$

In Figure 8 and in Algorithm 6 the theoretical cumulants of the CIR process, given in Equation 58, together with the empirical moments are plotted. As the order of the cumulants increase the empirical cumulants become much more volatile. But since the saddlepoint transition density function approximation utilizes a Taylor-series expansion, the weight put on the higher order cumulants in the transition density function approximation becomes less significant as the order of the cumulants increase.

Applying the methodology, as provided in [8], the saddlepoint transition density function approximation of the CIR diffusion process is derived as follows. Consider the approximate cumulant generating function with $N = 4$, given in Equation 18 as:

$$K_X(t) \approx \widetilde{K}_X(t) = tK_1(t) + \frac{1}{2!}t^2K_2(t) + \frac{1}{3!}t^3K_3(t) + \frac{1}{4!}t^4K_4(t), \tag{59}$$

for $K_i(t)$ for $i = 1, 2, 3, 4$ as in Equation 58. A Taylor-series is applied to get $\widetilde{K}_{X_t}(t)$. Where the exact cumulant generating function of the CIR process is $K_X(t) = ln(M_X(t))$, provided $M_{X_t}(t)$ exists and $M_{X_t}(t) > 0$ for all values of t, where $M_{X_t}(t)$ is the exact moment generating function of the CIR process. Consider the first and second order partial derivatives of Equation 59, in terms of t:

$$\widetilde{K}_X^{'}(t) = \frac{\partial}{\partial t}\widetilde{K}_X(t) = K_1(t) + tK_2(t) + \frac{1}{2}t^2K_3(t) + \frac{1}{6}t^3K_4(t), \tag{60}$$

$$\widetilde{K}_X^{''}(r) = \frac{\partial^2}{\partial t^2}\widetilde{K}_X(t) = K_2(t) + tK_3(t) + \frac{1}{2}t^2K_4(t). \tag{61}$$

Setting $X_t = \widetilde{K}_X^{'}(t)$, t is determined as a function of X_t:

$$t = \frac{-K_2(t) + \sqrt{(K_2(t))^2 - 2K_3(t)(K_1(t) - X_t)}}{K_3(t)} \tag{62}$$

Using the result given in [8], and the results in Equation 60,61 and 62, the closed-form saddlepoint transition density function approximation, $h_X(X_t, t|X_s, s; \boldsymbol{\theta})$, for the true transition density function of the CIR process is obtained as:

$$h_X(x_t, t|x_s, s; \boldsymbol{\theta}) = \exp(\widetilde{K}_X(t) - tx_t)\sqrt{(2\pi\widetilde{K}_X^{''}(t))^{-1}},$$

which gives:

$$\begin{aligned} h_X(x_t, t|x_s, s; \boldsymbol{\theta}) =& \sqrt{(2\pi(K_2(t) + tK_3(t) + \frac{1}{2}t^2K_4(t)))^{-1}} \\ & \times \exp\left[tK_1(t) + \frac{1}{2!}t^2K_2(t) + \frac{1}{3!}t^3K_3(t) + \frac{1}{4!}t^4K_4(t)\right] \\ & \times \exp\Big[-\left[\frac{-K_2(t) + \sqrt{(K_2(t))^2 - 2K_3(t)(K_1(t) - x_t)}}{K_3(t)}\right]X_t\Big] \end{aligned} \tag{63}$$

From Figure 9 it can be seen that the saddlepoint transition density function approximation is extremely accurate in approximating the true transition density function, from $X_t \approx 2.3$ onward. Since an efficient and accurate closed-form approximation has been developed, this transition density function approximation can now be applied to actual financial data. The S&P 500's volatility index (VIX) will be considered and analyzed.

4.1.8 Maximum likelihood estimation of parameters for the scalar CIR process

In this section, actual financial data will be analyzed. Consider the daily S&P 500 volatility index (VIX), which give a measure of the day-to-day variation in the S&P 500's index, which many investors use as a barometer of the actual market performance. Figure 10 gives a time-plot for the daily VIX-values for

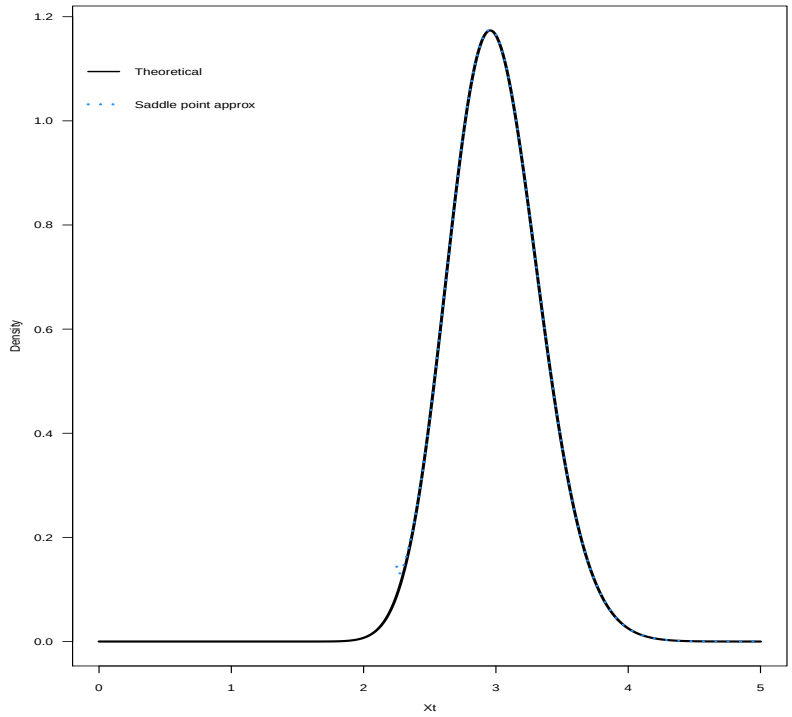


Figure 9: True transition density function and the moment-truncated saddlepoint transition density function approximation of the transition density function of the scalar CIR process.

30 December 2011 to 1 January 2017. Note that it is assumed that there are 250 trading days in a year. Therefore $\Delta t = \frac{1}{250}$. The 5 year S&P500VIX time series is plotted in Algorithm 8.

Firstly, maximum likelihood estimation, assuming a normal error-distribution, is done based on the true transition density function of the scalar CIR process. Consider

$$\max_{\boldsymbol{\theta}} \Big[\ln(L(\boldsymbol{\theta}|\boldsymbol{X})) \Big] = \max_{\boldsymbol{\theta}} \Big[\ln \prod_{i=1}^{1250} (p_X(X_t, t|X_s, s; \boldsymbol{\theta})) \Big] \tag{64}$$

$$= \max_{\boldsymbol{\theta}} \Big[\ln \prod_{i=1}^{1250} (c \exp(-(u+v)) \big(\frac{v}{u}\big)^{\frac{q}{2}} I_q(2(uv)^{\frac{1}{2}})) \Big] \tag{65}$$

Maximization of Equation 64, in Algorithm 9 gives the theoretical maximum likelihood estimates as:

$$\hat{\boldsymbol{\theta}} = (\hat{a}, \hat{\beta}, \hat{\sigma}) = (22.43, 15.74, 5.13)$$

Secondly, maximum likelihood estimation is done based on the saddlepoint transition density function approximation of the transition density function of the scalar CIR process. Consider:

$$\max_{\boldsymbol{\theta}} \Big[\ln(L(\boldsymbol{\theta}|\boldsymbol{X})) \Big] = \max_{\boldsymbol{\theta}} \Big[\ln \prod_{i=1}^{1250} (h_X(x_t, t|x_s, s; \boldsymbol{\theta})) \Big], \tag{66}$$

where $h_X(x_t, t|x_s, s; \boldsymbol{\theta})$ is given as the saddlepoint transition density function approximation in Equation

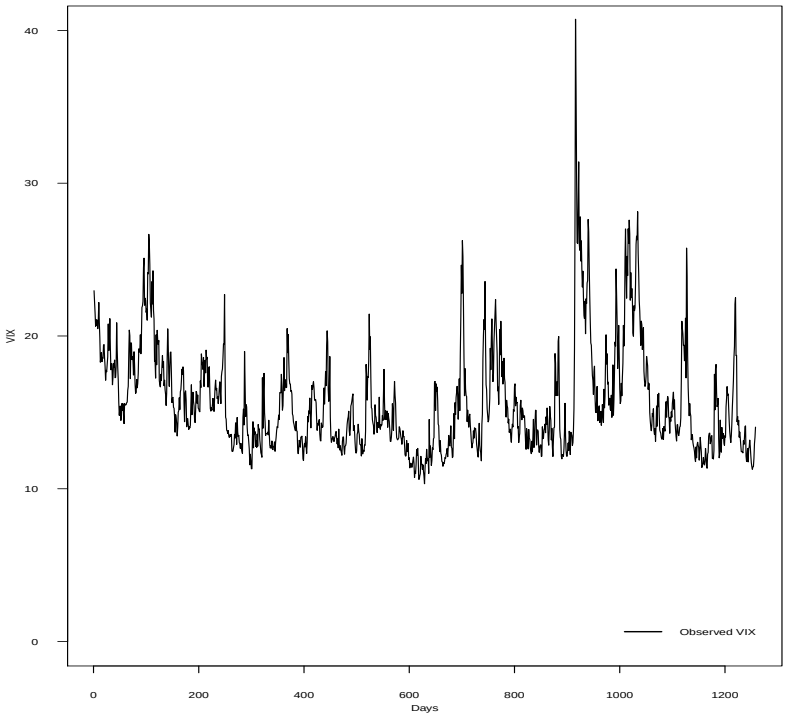


Figure 10: VIX time-plot for 30 December 2011 to 1 January 2017.

63. Maximization of Equation 66, in Algorithm 10, gives the saddlepoint transition density function approximation maximum likelihood estimates as:

$$\hat{\boldsymbol{\theta}}_s = (\hat{a}_s, \hat{\beta}_s, \hat{\sigma_s}) = (22.27, 15.55, 5.27)$$

Figure 11 gives the simulated trajectories of the scalar CIR process based on the mle's of the true transition density function and saddlepoint transition density function approximation of the scalar CIR transition density function.

4.2 Application to the mixed-effects CIR process

Consider the following mixed-effects CIR process, a generalization of the scalar Cox, Ingersoll and Ross process :

$$dX_t = \alpha(\beta - X_t)dt + \mathring{\sigma}\sqrt{X_t}dW_t, \tag{67}$$

where

$$\mathring{\sigma} \sim N(0.25, 0.15^2).$$

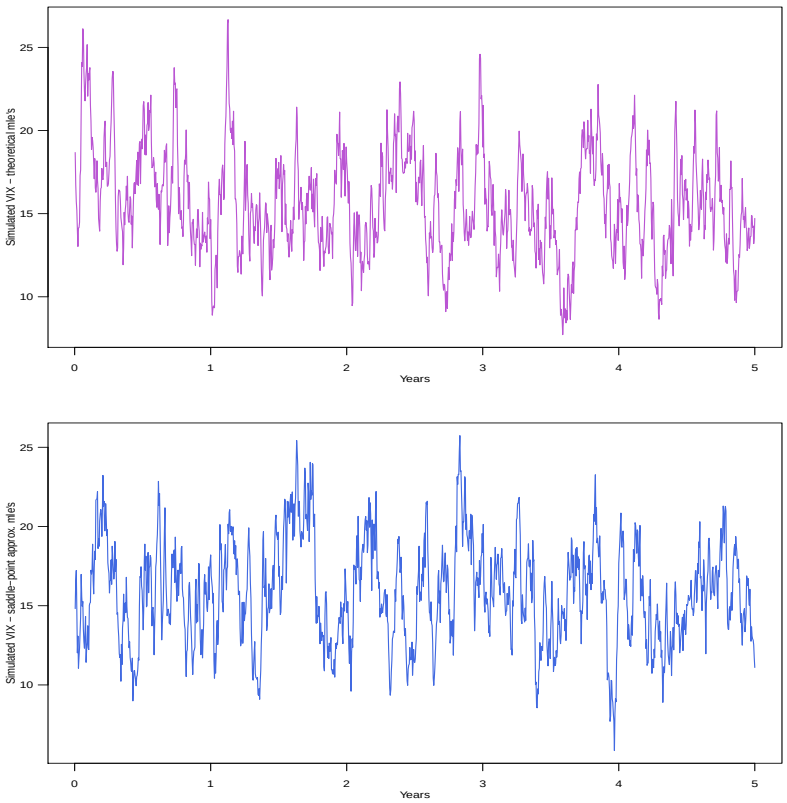


Figure 11: Simulated trajectories of the scalar CIR process based on the mle's of the true- and saddlepoint transition density functions.

4.2.1 Parameter values to be used in the present paper for the mixed-effects CIR process

Throughout the following sections the parameter values and domains used to analyze the mixed-effects CIR process in Equation 67 are given by the time-dimension $[s,T] = [0,5]$, state-space $[X_I, X_T] = [0,5]$, with $X_s = 2.75$ as the initial state and parameter vector $\boldsymbol{\mathring{\theta}} = (\alpha, \beta, \sigma) = (0.8, 3, \mathring{\sigma})$ where $\mathring{\sigma} \sim N(0.25, 0.15^2)$, hence $\boldsymbol{\gamma} = (0.8, 3, 0.25, 0.15)$.

4.2.2 Simulated trajectory of the mixed-effects CIR process

The function **rnorm()** in R is used to simulate the random Brownian motion or Wiener Process values,$\Delta_{W_t} \sim N(0, \Delta_t^2)$. These values contribute to the stochastic or random part of the trajectory, where $dt \approx \Delta_t = 0.01$ is used as a step length in the time-dimension $[0,5]$. The mixed-effects CIR process as given in Equation 67 are discretized by the implementation of the following recursive algorithm:

$$X_{s+\Delta_t} = X_s + \alpha(\beta - X_s)\Delta_t + \mathring{\sigma}\sqrt{X_s}\Delta_{W_t}, \tag{68}$$

for all $t > s + \Delta_t$.

The trajectory is simulated λ times, with a new simulated value of $\mathring{\sigma}$ from a $N(\nu, \varrho^2)$ distribution at each simulation. The average of λtrajectories is calculated and also plotted as a trajectory. As can be seen in Figure 12 various values of $\lambda$are plotted and that the average trajectory smooths out towards the mean-reverted value of β as λincreases.

The code for the plot in Figure 12 is provided in Algorithm 12.

4.2.3 Euler-Maruyama scheme of the scalar CIR process

Consider the time-dimension $[0,5]$ s.t $t \in [0,5]$, the dicretization of the time-dimension is given by: $\Delta_t = \frac{T}{N} = \frac{5}{500} = 0.01$ for $N = 500 \in \mathbb{N}$. The Euler-Maruyama scheme is now given by the following recursive relationship:

$$X_i = X_{i-1} + \alpha(\beta - X_{i-1})\Delta_t + \mathring{\sigma}\sqrt{X_{i-1}}\Delta_{W_i} \tag{69}$$

for all $i = 1, 2, ..., 499, 500$ and with $X_s = 2.75$ as initial state and where $\Delta_{W_t} \sim N(0, \Delta_t^2)$. The Euler-Maruyama scheme is simulated λ times, with a new simulated value of $\mathring{\sigma}$ from a $N(\nu, \varrho^2)$ distribution at each simulation. As can be seen in Figure 13 the Euler-Maruyama schemes have been overlayed to give a better idea of the impact the random effect has on the distribution. As the average is calculated over a larger amount of simulations it can be seen in Figure that the process clearly reverts towards $\beta = 3$. The frequency plots in Figure 14 also shows that with increasing simulations the process tends towards β. The code for Figure 13 is provided in Algorithm 14 and the code for Figure 14 is provided in Algorithm 21.

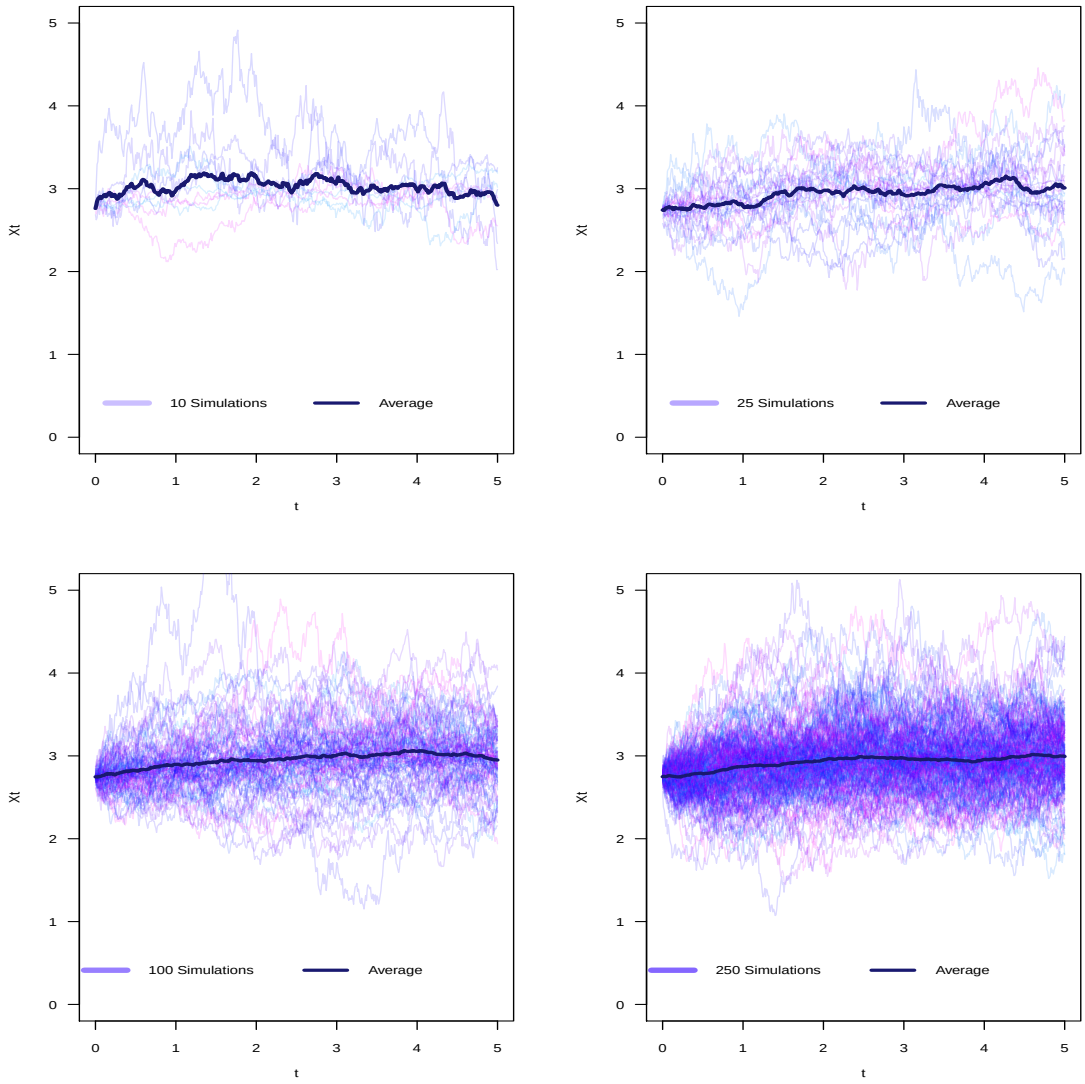


Figure 12: Simulated trajectories of the mixed-effects CIR diffusion process with various number of simulations.

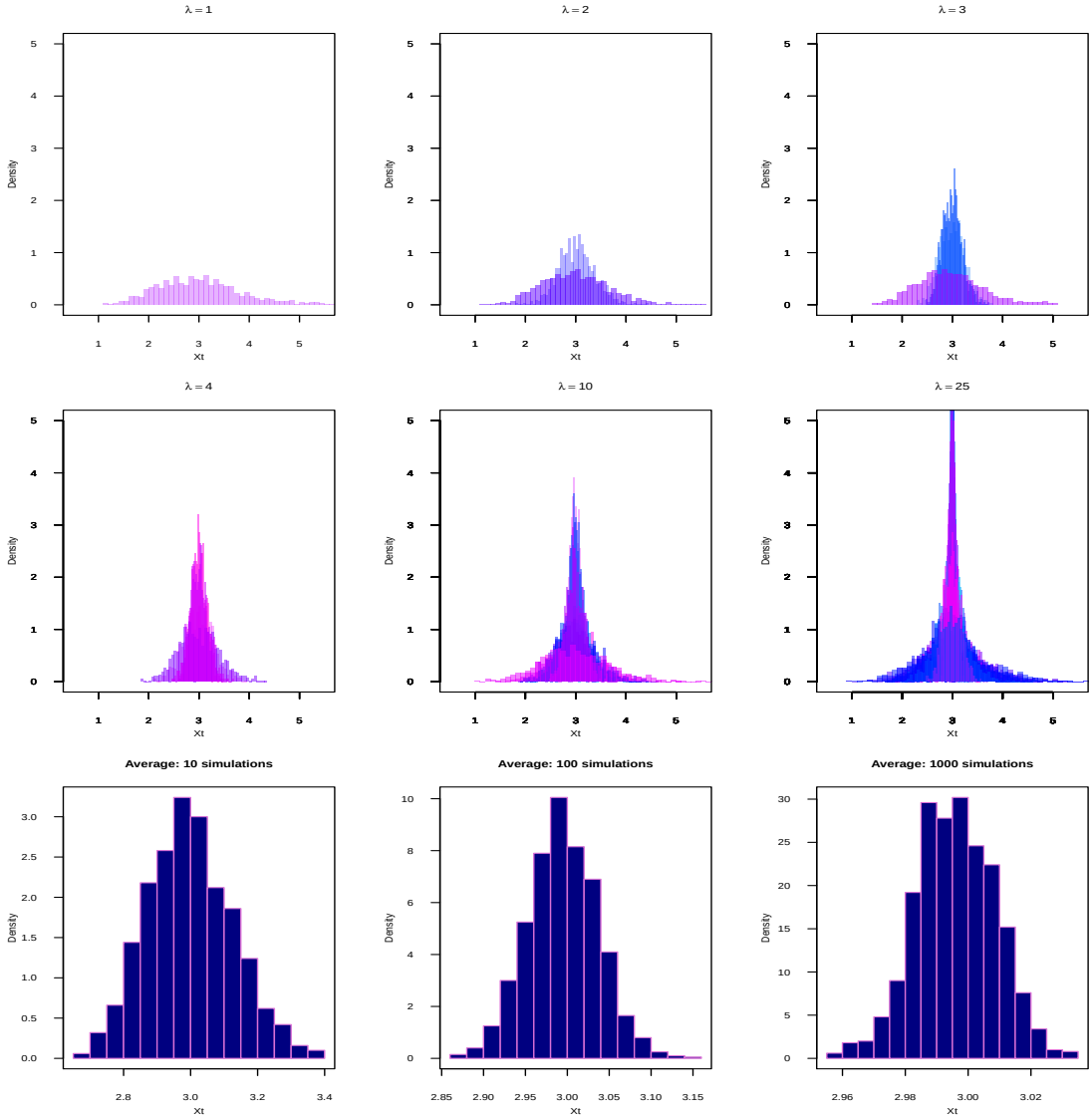


Figure 13: Simulated Euler-Maruyama schemes for various $\mathring{\sigma}$ simulations.

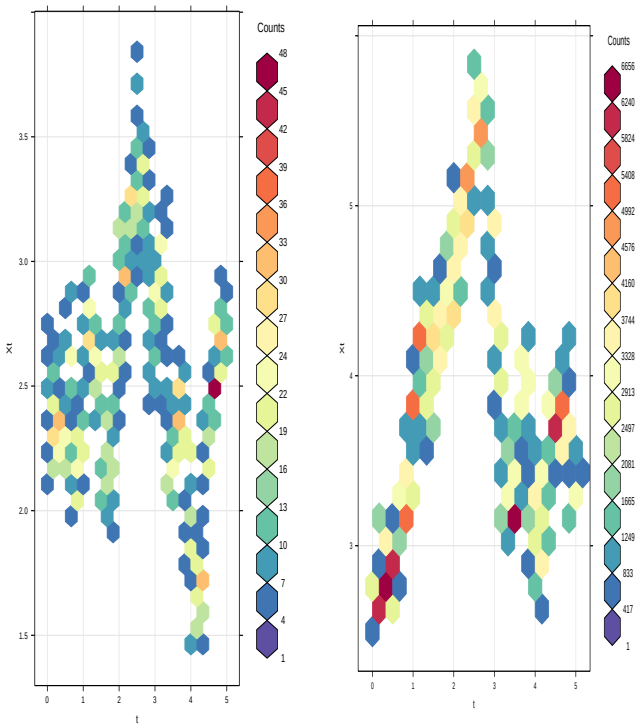


Figure 14: Frequency plots for 1 and 10 $\mathring{\sigma}$ simulations respectively.

4.2.4 Perspective plots for the mixed-effects CIR process

Plotting the true transition density for the scalar CIR process:

$$p_X(X_t, t|X_s, s; \mathring{\boldsymbol{\theta}}) = c\exp(-(u+v))\left(\frac{v}{u}\right)^{\frac{q}{2}} I_q(2(uv)^{\frac{1}{2}}), \tag{70}$$

for a various number of simulations and random values of $\mathring{\sigma}$ from $N(\upsilon, \varrho^2)$ and averaging the results, the movement of the CIR short-rate can be analyzed visually for various $\mathring{\sigma}$ values. Figure 15 shows these overlayed plots and average result. The corresponding code can be found in Algorithm 13. Again it is clear X_t tends towards $\beta = 3$, but at different densities due to the random effect.

4.2.5 Moment-truncated saddlepoint transition density function approximation of the mixed-effects CIR process

Consider the moment trajectories derived for the **scalar** CIR process:

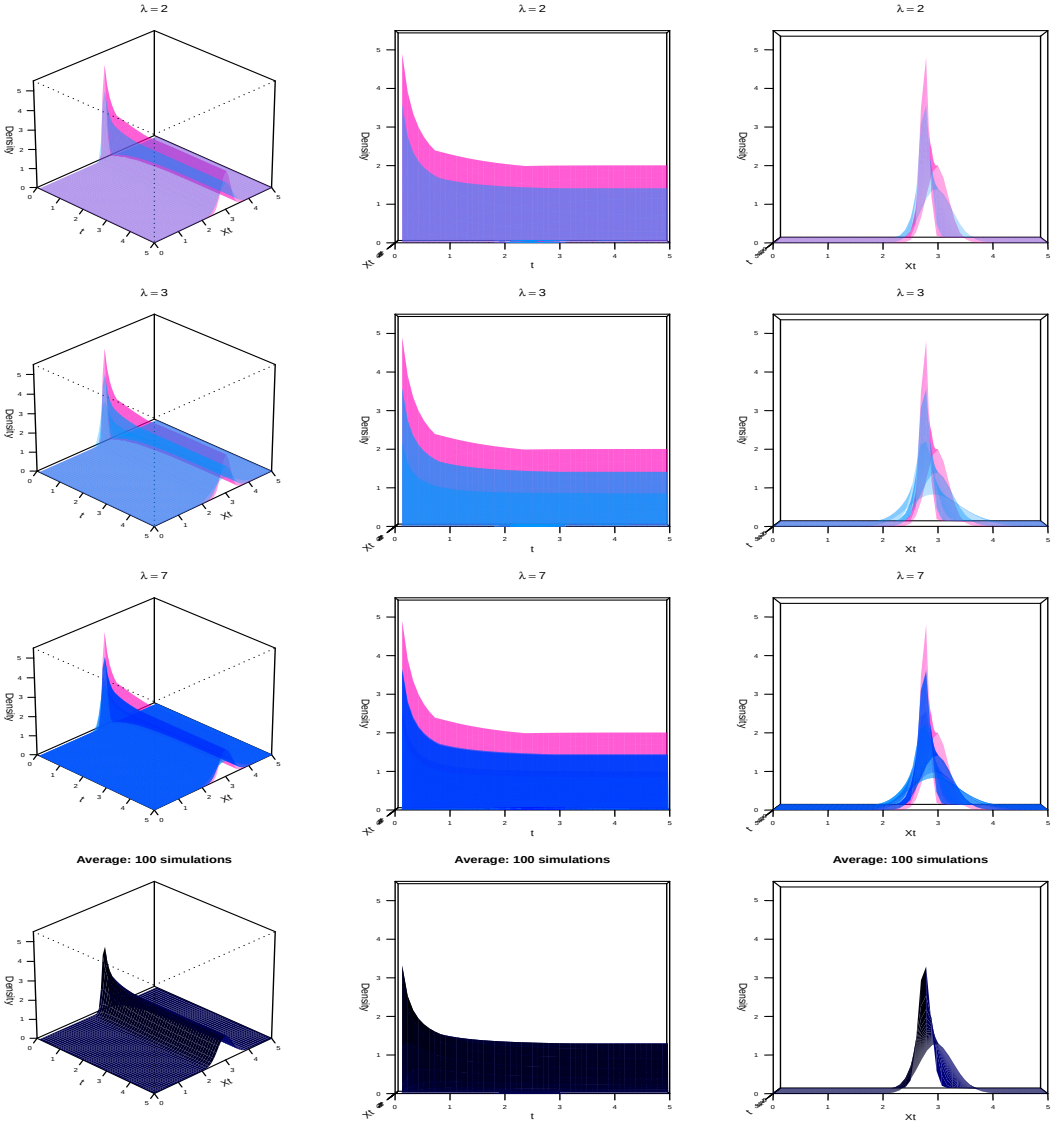


Figure 15: Perspective plots of the transition density function for various $\mathring{\sigma}$ simulations.

$$
\begin{aligned}
E[X_t|X_s] =& m_1(t) = X_s e^{-\alpha t} + \beta(1 - e^{-\alpha t}),\\
E[X_t^2|X_s] =& m_2(t) = X_s^2 e^{-2\alpha t} + (\beta + \frac{\sigma^2}{2\alpha})(\beta + 2(X_s - \beta)e^{-\alpha t} + (\beta - 2X_s)e^{-2\alpha t}),\\
E[X_t^3|X_s] =& m_3(t) = X_s^3 e^{-3\alpha t} + 3(\alpha\beta + \sigma^2)(A + Be^{-\alpha t} + Ce^{-2\alpha t} + De^{-3\alpha t}),\\
E[X_t^4|X_s] =& m_4(t) = X_s^4 e^{-4\alpha t} + (4\alpha\beta + 6\sigma^2)\Big[E + Fe^{-\alpha t} + Ge^{-2\alpha t} + He^{-3\alpha t} + Ie^{-4\alpha t}\Big].
\end{aligned}
\tag{71}
$$

To simplify proceedings, Equation 71 can in accordance to [12], be written in the following general form:

$$
m_j(t) = X_s^j \mathcal{L}^{-1}\Big\{\frac{1}{k + i\alpha}\Big\} + A_j(\alpha, \beta, \sigma)\mathcal{L}^{-1}\Big\{\frac{\ell^{j-1}(k)}{k + i\alpha}\Big\}, \tag{72}
$$

$$
\text{for } j = 1, 2, 3, 4,
$$

$$
\text{s.t } \ell^{j-1}(k) = \mathcal{L}\Big\{m_{j-i}(t)\Big\} = M_{j-1}(k),
$$

$$
\text{where } \ell^0(k) = \mathcal{L}\Big\{m_0(t)\Big\} = \mathcal{L}\Big\{\int \exp(0) p_X(x_t, t|x_s, s; \boldsymbol{\theta})dx\Big\} = \mathcal{L}\Big\{1\Big\} = \tfrac{1}{k},
$$

$$
\text{with scalar coefficient function } A_j(\alpha, \beta, \sigma) = j\alpha\beta + \tfrac{j(j-2)}{2}\sigma^2.
$$

Equation 71 then simplifies to:

$$
\begin{aligned}
E[X_t|X_s] =& m_1(t) = X_s\mathcal{L}^{-1}\Big\{\frac{1}{k + \alpha}\Big\} + A_1(\alpha, \beta, \sigma)\mathcal{L}^{-1}\Big\{\frac{\ell^0(k)}{k + \alpha}\Big\},\\
E[X_t^2|X_s] =& m_2(t) = X_s^2\mathcal{L}^{-1}\Big\{\frac{1}{k + 2\alpha}\Big\} + A_2(\alpha, \beta, \sigma)\mathcal{L}^{-1}\Big\{\frac{\ell^1(k)}{k + 2\alpha}\Big\},\\
E[X_t^3|X_s] =& m_3(t) = X_s^3\mathcal{L}^{-1}\Big\{\frac{1}{k + 3\alpha}\Big\} + A_3(\alpha, \beta, \sigma)\mathcal{L}^{-1}\Big\{\frac{\ell^2(k)}{k + 3\alpha}\Big\},\\
E[X_t^4|X_s] =& m_4(t) = X_s^4\mathcal{L}^{-1}\Big\{\frac{1}{k + 4\alpha}\Big\} + A_4(\alpha, \beta, \sigma)\mathcal{L}^{-1}\Big\{\frac{\ell^3(k)}{k + 4\alpha}\Big\},
\end{aligned}
\tag{73}
$$

s.t

$$
\begin{aligned}
A_1(\alpha, \beta, \sigma) =& \alpha\beta,\\
A_2(\alpha, \beta, \sigma) =& 2\alpha\beta + \sigma^2,\\
A_3(\alpha, \beta, \sigma) =& 3\alpha\beta + 3\sigma^2,\\
A_4(\alpha, \beta, \sigma) =& 4\alpha\beta + 6\sigma^2,
\end{aligned}
\tag{74}
$$

and

$$
\begin{aligned}
\ell^0(k) &= \mathcal{L}\Big\{m_0(t)\Big\} = M_0(k) = \frac{1}{k},\\
\ell^1(k) &= \mathcal{L}\Big\{m_1(t)\Big\} = M_1(k) = X_s\Big(\frac{1}{k+\alpha}\Big) + A_1(\alpha,\beta,\sigma)\Big(\frac{\frac{1}{k}}{k+\alpha}\Big),\\
\ell^2(k) &= \mathcal{L}\Big\{m_2(t)\Big\} = M_2(k) = X_s^2\Big(\frac{1}{k+2\alpha}\Big) + A_2(\alpha,\beta,\sigma)\Bigg[\frac{X_s\Big(\frac{1}{k+\alpha}\Big) + A_1(\alpha,\beta,\sigma)\Big(\frac{\frac{1}{k}}{k+\alpha}\Big)}{k+2\alpha}\Bigg],\\
\ell^3(k) &= \mathcal{L}\Big\{m_3(t)\Big\} = M_3(k) = X_s^3\Big(\frac{1}{k+3\alpha}\Big)\\
&\qquad + A_3(\alpha,\beta,\sigma)\Bigg[\frac{X_s^2\Big(\frac{1}{k+2\alpha}\Big) + A_2(\alpha,\beta,\sigma)\mathcal{L}^{-1}\Big(\frac{X_s\Big(\frac{1}{k+\alpha}\Big)+A_1(\alpha,\beta,\sigma)\Big(\frac{\frac{1}{k}}{k+\alpha}\Big)}{k+2\alpha}\Big)}{k+3\alpha}\Bigg].
\end{aligned}
\tag{75}
$$

Substituting Equation 74 and Equation 75 into Equation 73 yields:

$$
\begin{aligned}
E[X_t|X_s] =& m_1(t) = X_s\mathcal{L}^{-1}\Big\{\frac{1}{k+\alpha}\Big\} + \alpha\beta\mathcal{L}^{-1}\Big\{\frac{\frac{1}{k}}{k+\alpha}\Big\},\\
E[X_t^2|X_s] =& m_2(t) = X_s^2\mathcal{L}^{-1}\Big\{\frac{1}{k+2\alpha}\Big\} + (2\alpha\beta+\sigma^2)\mathcal{L}^{-1}\Big\{\frac{X_s\Big(\frac{1}{k+\alpha}\Big) + A_1(\alpha,\beta,\sigma)\Big(\frac{\frac{1}{k}}{k+\alpha}\Big)}{k+2\alpha}\Big\},\\
E[X_t^3|X_s] =& m_3(t) = X_s^3\mathcal{L}^{-1}\Big\{\frac{1}{k+3\alpha}\Big\}\\
&+(3\alpha\beta+3\sigma^2)\mathcal{L}^{-1}\Bigg\{\frac{X_s^2\Big(\frac{1}{k+2\alpha}\Big) + A_2(\alpha,\beta,\sigma)\Bigg[\frac{X_s\Big(\frac{1}{k+\alpha}\Big)+A_1(\alpha,\beta,\sigma)\Big(\frac{\frac{1}{k}}{k+\alpha}\Big)}{k+2\alpha}\Bigg]}{k+3\alpha}\Bigg\},\\
E[X_t^4|X_s] =& m_4(t) = X_s^4\mathcal{L}^{-1}\Big\{\frac{1}{k+4\alpha}\Big\}\\
&+(4\alpha\beta+6\sigma^2)\mathcal{L}^{-1}\Bigg\{\frac{X_s^3\Big(\frac{1}{k+3\alpha}\Big) + A_3(\alpha,\beta,\sigma)\Bigg[\frac{X_s^2\Big(\frac{1}{k+2\alpha}\Big)+A_2(\alpha,\beta,\sigma)\mathcal{L}^{-1}\Big(\frac{X_s\Big(\frac{1}{k+\alpha}\Big)+A_1(\alpha,\beta,\sigma)\Big(\frac{\frac{1}{k}}{k+\alpha}\Big)}{k+2\alpha}\Big)}{k+3\alpha}\Bigg]}{k+4\alpha}\Bigg\}.
\end{aligned}
\tag{76}
$$

By factorizing out the scalar coefficients and parameters, the following inverse Laplace transforms comes to the foreground; where solving the Laplace transforms yields:

$$
\begin{aligned}
\mathcal{L}^{-1}\left\{\frac{1}{k}\right\} &= 1, \\
\mathcal{L}^{-1}\left\{\frac{1}{k+\alpha}\right\} &= \exp(-\alpha t), \\
\mathcal{L}^{-1}\left\{\frac{1}{k+2\alpha}\right\} &= \exp(-2\alpha t), \\
\mathcal{L}^{-1}\left\{\frac{1}{k+3\alpha}\right\} &= \exp(-3\alpha t), \\
\mathcal{L}^{-1}\left\{\frac{1}{k+4\alpha}\right\} &= \exp(-4\alpha t), \\
\mathcal{L}^{-1}\left\{\frac{1}{k(k+\alpha)}\right\} &= \frac{1}{\alpha}\exp(-\alpha t), \\
\mathcal{L}^{-1}\left\{\frac{1}{k(k+\alpha)(k+2\alpha)}\right\} &= \frac{1}{2\alpha^2}\exp(-2\alpha t)(\exp(\alpha t)-1)^2, \\
\mathcal{L}^{-1}\left\{\frac{1}{k(k+\alpha)(k+2\alpha)(k+3\alpha)}\right\} &= \frac{1}{6\alpha^3}\exp(-3\alpha t)(\exp(\alpha t)-1)^3, \\
\mathcal{L}^{-1}\left\{\frac{1}{k(k+\alpha)(k+2\alpha)(k+3\alpha)(k+4\alpha)}\right\} &= \frac{1}{24\alpha^4}\exp(-4\alpha t)(\exp(\alpha t)-1)^4, \\
\mathcal{L}^{-1}\left\{\frac{1}{(k+\alpha)(k+2\alpha)(k+3\alpha)(k+4\alpha)}\right\} &= \frac{1}{6\alpha^3}\exp(-4\alpha t)(\exp(\alpha t)-1)^3, \\
\mathcal{L}^{-1}\left\{\frac{1}{(k+2\alpha)(k+3\alpha)(k+4\alpha)}\right\} &= \frac{1}{2\alpha^2}\exp(-4\alpha t)(\exp(\alpha t)-1)^2, \\
\mathcal{L}^{-1}\left\{\frac{1}{(k+3\alpha)(k+4\alpha)}\right\} &= \frac{1}{\alpha}\exp(-4\alpha t)(\exp(\alpha t)-1), \\
\mathcal{L}^{-1}\left\{\frac{1}{(k+2\alpha)(k+3\alpha)}\right\} &= \frac{1}{\alpha}\exp(-3\alpha t)(\exp(\alpha t)-1), \\
\mathcal{L}^{-1}\left\{\frac{1}{(k+\alpha)(k+2\alpha)}\right\} &= \frac{1}{\alpha}\exp(-2\alpha t)(\exp(\alpha t)-1).
\end{aligned}
\tag{77}
$$

Simplifying Equation 76, using Equation 77, yields the original moments obtained for the scalar CIR process, but in a form which is more appropriate to generalize to the mixed-effects CIR process.

Moment trajectories of the **mixed-effects** CIR process:

Note that scalar σ now becomes a random effect with an assumed Normal distribution, i.e. $\mathring{\sigma} \sim N(\upsilon, \varrho^2)$. The mixed-effects coefficients are now calculated by taking the expectation over $\mathring{\sigma}$:

$$
\begin{aligned}
B_1(\alpha,\beta,\mathring{\sigma}) =& E_{\mathring{\sigma}}[\alpha\beta] = \alpha\beta, \\
B_2(\alpha,\beta,\mathring{\sigma}) =& E_{\mathring{\sigma}}[2\alpha\beta + \mathring{\sigma}^2] = 2\alpha\beta + E_{\mathring{\sigma}}[\mathring{\sigma}^2] = 2\alpha\beta + (\varrho^2 + \upsilon^2), \\
B_3(\alpha,\beta,\mathring{\sigma}) =& E_{\mathring{\sigma}}[3\alpha\beta + 3\mathring{\sigma}^2] = 3\alpha\beta + 3E_{\mathring{\sigma}}[\mathring{\sigma}^2] = 3\alpha\beta + 3(\varrho^2 + \upsilon^2), \\
B_4(\alpha,\beta,\mathring{\sigma}) =& E_{\mathring{\sigma}}[4\alpha\beta + 6\mathring{\sigma}^2] = 4\alpha\beta + 6E_{\mathring{\sigma}}[\mathring{\sigma}^2] = 4\alpha\beta + 6(\varrho^2 + \upsilon^2).
\end{aligned}
\tag{78}
$$

Note that

$$E_{\mathring{\sigma}}[\mathring{\sigma}^2] = VAR(\mathring{\sigma}) + \Big(E[\mathring{\sigma}]\Big)^2 = \varrho^2 + \upsilon^2.$$

From the moment trajectories of the scalar CIR process the A_j-coefficients can be identified and replaced by the generalized B_j-coefficients. Finally the moment trajectories of the mixed-effects CIR process becomes:

$$\begin{aligned}
E[X_t|X_s] =& m_1(t) = X_s \exp(-\alpha t) \\
&+ B_1(\alpha,\beta,\mathring{\sigma})\left[\frac{1}{\alpha}(1-\exp(-\alpha t))\right], \\
E[X_t^2|X_s] =& m_2(t) = X_s^2 \exp(-2\alpha t) \\
&+ B_2(\alpha,\beta,\mathring{\sigma})\left[X_S\frac{1}{\alpha}\exp(-2\alpha t)(\exp(\alpha t)-1) + B_1(\alpha,\beta,\mathring{\sigma})\frac{1}{2\alpha^2}\exp(-2\alpha t)(\exp(\alpha t)-1)^2\right], \\
E[X_t^3|X_s] =& m_3(t) = X_s^3 \exp(-3\alpha t) \\
&+ B_3(\alpha,\beta,\mathring{\sigma})\left[X_s^2\frac{1}{\alpha}\exp(-3\alpha t)(\exp(\alpha t)-1) + B_2(\alpha,\beta,\mathring{\sigma})X_s\frac{1}{2\alpha^2}\exp(-2\alpha t)(\exp(\alpha t)-1)^2\right. \\
&\left.+ B_2(\alpha,\beta,\mathring{\sigma})B_1(\alpha,\beta,\mathring{\sigma})\frac{1}{6\alpha^3}\exp(-3\alpha t)(\exp(\alpha t)-1)^3\right], \\
E[X_t^4|X_s] =& m_4(t) = X_s^4 \exp(-4\alpha t) \\
&+ B_4(\alpha,\beta,\mathring{\sigma})\left[X_s^3\frac{1}{\alpha}\exp(-4\alpha t)(\exp(\alpha t)-1) + B_3(\alpha,\beta,\mathring{\sigma})X_s^2\frac{1}{2\alpha^2}\exp(-4\alpha t)(\exp(\alpha t)-1)^2\right. \\
&+ B_3(\alpha,\beta,\mathring{\sigma})B_2(\alpha,\beta,\mathring{\sigma})Xs\frac{1}{6\alpha^3}\exp(-4\alpha t)(\exp(\alpha t)-1)^3 \\
&\left.+ B_3(\alpha,\beta,\mathring{\sigma})B_2(\alpha,\beta,\mathring{\sigma})B_1(\alpha,\beta,\mathring{\sigma})\frac{1}{24\alpha^4}\exp(-4\alpha t)(\exp(\alpha t)-1)^4\right].
\end{aligned} \tag{79}$$

The empirical moment trajectories and the theoretical moment trajectories of the mixed-effects CIR process as in Equation 79, with time-dimension $[s,T] = [0,5]$, state-space $[X_s, X_T] = [0,5]$, $X_s = 2.75$ as the initial state and parameter vector $\mathring{\boldsymbol{\theta}} = (\alpha,\beta,\sigma) = (0.8, 3, \mathring{\sigma})$ where $\mathring{\sigma} \sim N(\nu = 0.25, \varrho^2 = 0.15^2)$ (i.e $\boldsymbol{\delta} = (\alpha = 0.8, \beta = 3, \upsilon = 0.25, \varrho = 0.15)$), are plotted in Figure 16 with the code provided in Algorithm 15.

The first four cumulants of the of the mixed-effects CIR process are given by:

$$\begin{aligned}
K_1(t) &= m_1(t), \\
K_2(t) &= m_2(t) - (m_1(t))^2, \\
K_3(t) &= 2(m_1(t))^3 - 3(m_1(t))(m_2(t)) + m_3(t), \\
K_3(t) &= -6(m_1(t))^4 + 12(m_1(t))^2(m_2(t)) - 3(m_2(t))^2 - 4(m_1(t))(m_3(t)) + m_4(t).
\end{aligned} \tag{80}$$

Due to the extra variation provided by the random effect, the third and fourth order empirical and

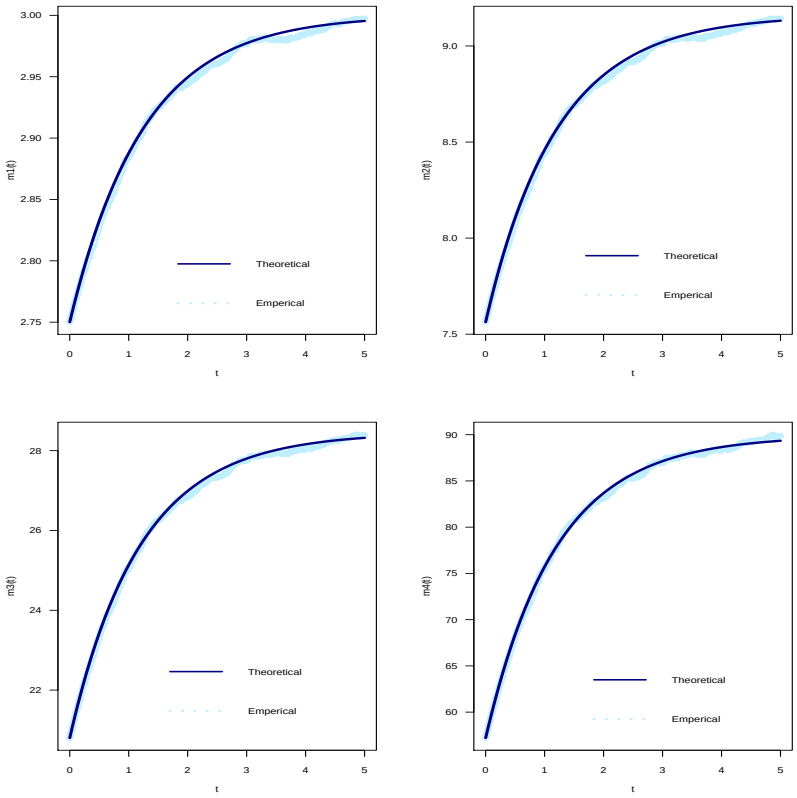


Figure 16: The first four theoretical and empirical moments of the mixed-effects CIR process.

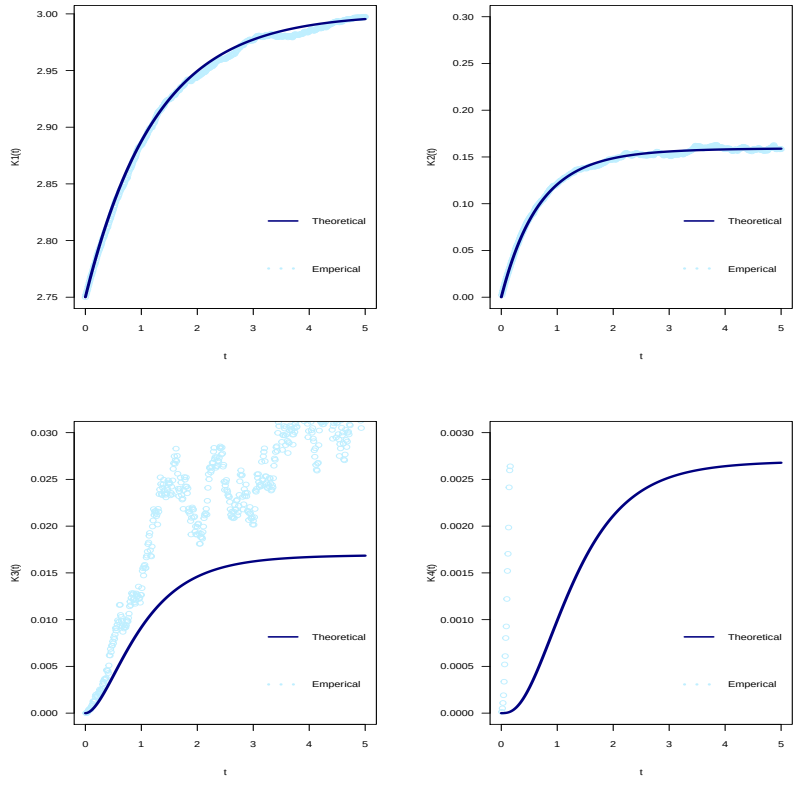


Figure 17: The first four theoretical and empirical cumulants of the mixed-effects CIR process.

theoretical cumulant trajectories is not closely related, however considering how small the cumulant value for the $K_3(t)$ and $K_4(t)$ is, the impact of the difference is insignificant. Applying the methodology, as provided in [8], the saddlepoint transition density function approximation of the mixed-effects CIR process is derived as follows. Consider the approximate cumulant generating function with $N = 4$, given in Equation 18 as:

$$K_X(t) \approx \widetilde{K}_X(t) = tK_1(t) + \frac{1}{2!}t^2K_2(t) + \frac{1}{3!}t^3K_3(t) + \frac{1}{4!}t^4K_4(t), \tag{81}$$

for $K_i(t)$ for all $i = 1, 2, 3, 4$ as in Equation 58. A Taylor series is applied to get $\widetilde{K}_{X_t}(t)$. Where the exact cumulant generating function of the CIR process is $K_X(t) = ln(M_X(t))$, provided $M_{X_t}(t)$ exists and $M_{X_t}(t) > 0$ for all values of t, where $M_{X_t}(t)$ is the exact moment generating function of the mixed-effects CIR process. Consider the first and second order partial derivatives of Equation 81, in terms of t:

$$\widetilde{K}_X^{'}(t) = \frac{\partial}{\partial t}\widetilde{K}_X(t) = K_1(t) + tK_2(t) + \frac{1}{2}t^2K_3(t) + \frac{1}{6}t^3K_4(t), \tag{82}$$

$$\widetilde{K}_X^{''}(r) = \frac{\partial^2}{\partial t^2}\widetilde{K}_X(t) = K_2(t) + tK_3(t) + \frac{1}{2}t^2K_4(t). \tag{83}$$

Setting $X_t = \widetilde{K}_X^{'}(t)$, t is determined as a function of X_t:

$$t = \frac{-K_2(t) + \sqrt{(K_2(t))^2 - 2K_3(t)(K_1(t) - X_t)}}{K_3(t)}. \tag{84}$$

Using the result given in [8], and the results in Equation 82,83 and 84, the closed-form saddlepoint transition density function approximation, $w_X(X_t, t|X_s, s; \mathring{\boldsymbol{\theta}})$, for the mixed-effects CIR process is obtained as:

$$w_X(x_t, t|x_s, s; \mathring{\boldsymbol{\theta}}) = \exp(\widetilde{K}_X(t) - tx_t)\sqrt{(2\pi\widetilde{K}_X^{''}(t))^{-1}},$$

which gives:

$$\begin{aligned} w_X(x_t, t|x_s, s; \mathring{\boldsymbol{\theta}}) =& \sqrt{(2\pi(K_2(t) + tK_3(t) + \frac{1}{2}t^2K_4(t)))^{-1}} \\ & \times \exp\left[tK_1(t) + \frac{1}{2!}t^2K_2(t) + \frac{1}{3!}t^3K_3(t) + \frac{1}{4!}t^4K_4(t)\right] \\ & \times \exp\Big[-\left[\frac{-K_2(t) + \sqrt{(K_2(t))^2 - 2K_3(t)(K_1(t) - x_t)}}{K_3(t)}\right]X_t\Big]. \end{aligned} \tag{85}$$

From Figure 18 it can be seen that the saddlepoint transition density function approximation is an efficient density function, from $X_t \approx 2.3$ onward. Although the approximation breaks between $0 \leq X_t < 2.3$,

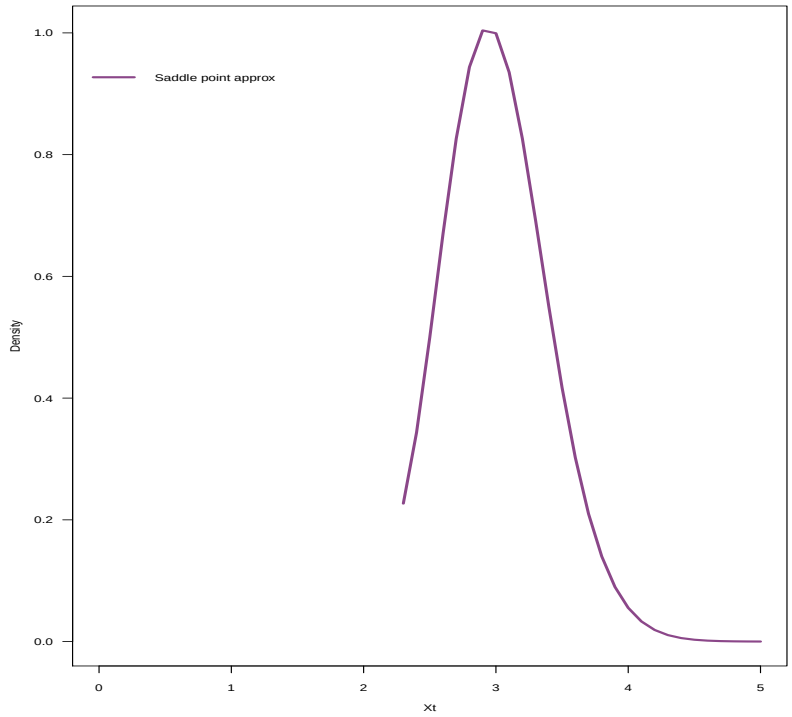


Figure 18: The moment-truncated saddlepoint transition density function approximation of the mixed-effects CIR process.

from $X_t \approx 2.3$ onward the moment-truncated saddlepoint is very accurate. Since the approximation is very close to symmetric, interpolation techniques can be utilized in the range $0 \leq X_t < 2.3$.

From Figure 19 it can be seen that the mixed-effects saddlepoint approximation provides higher density approximation to short-rate values near the mean-reverted value of $\beta = 3$, which is a clear improvement on the saddlepoint approximation of the scalar CIR process, since it provides a higher probability of observing the mean-reverted value.

4.2.6 Maximum likelihood estimation for the parameters of the mixed-effects CIR process

Consider the daily S&P 500 volatility index (VIX), which give a measure of the day-to-day variation in the S&P 500's index. Maximum likelihood estimation is now done based on the moment-truncated saddlepoint transition density function approximation of the transition density function of the mixed-effects CIR process. Consider:

$$\max_{\boldsymbol{\delta}} \Big[\ln(L(\boldsymbol{\delta}|\boldsymbol{X})) \Big] = \max_{\boldsymbol{\delta}} \Big[\ln \prod_{i=1}^{1250} (w_X(x_t, t|x_s, s; \boldsymbol{\delta})) \Big], \tag{86}$$

where $w_X(x_t, t|x_s, s; \boldsymbol{\delta})$ is given as the moment-truncated saddlepoint transition density function approximation in Equation 85. Maximization of Equation 86, in Algorithm 19 gives the saddlepoint transition density function approximation maximum likelihood estimates as:

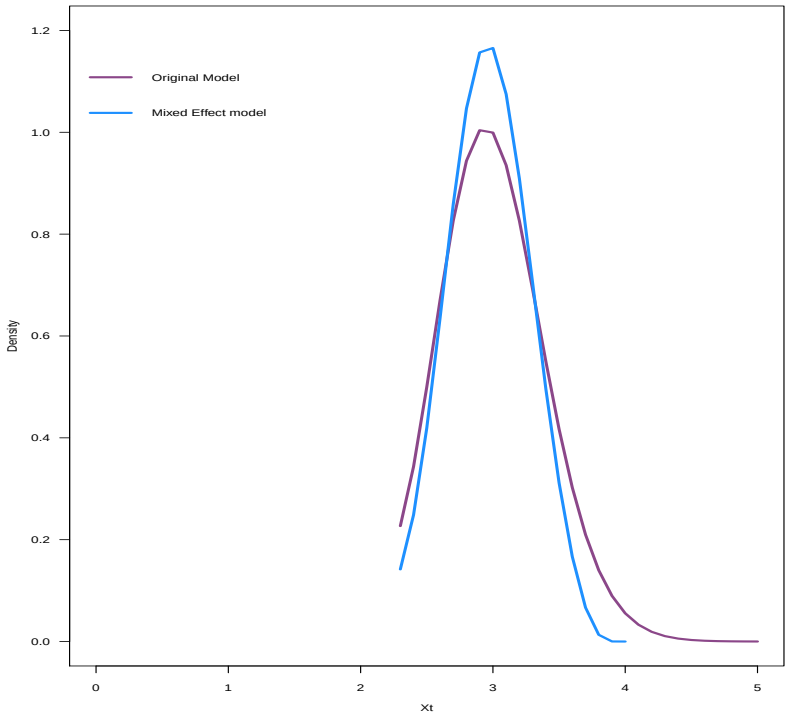


Figure 19: Overlayed moment-truncated saddlepoint transition density function approximations of the scalar CIR process and mixed-effects CIR process.

$$\hat{\boldsymbol{\delta}} = (\hat{\alpha}, \hat{\beta}, \hat{\nu}, \hat{\varrho}), = (22.15, 14.70, 4.96, 1.95)$$

4.3 Forecasting done in SAS and R based on 5 year S&P500VIX daily data

Figure 20 gives day 150-day forecast of the S&P500 volatility index, with the coding done in SAS, given in Algorithm 22, where an autoregressive model of order 1, AR(1), was fitted to the observed 5 year daily data.

Figure 21 gives a 250-day-forward simulated trajectory for the S&P500 volatility index, based on the mle's of the moment-truncated saddlepoint transition density approximation of the mixed-effects CIR process; with the coding done in R, given in Algorithm 20. The average of all the trajectories is used as the prediction, and as the number of simulations for $\mathring{\sigma}$ increases, the average trajectory closely resembles the forecast provided by the AR(1)-fitted model. The forward simulated trajectory done with the mixed-effects saddlepoint transition density approximation seems to be more efficient than an AR(1)-fitted model forecast, since it seems to take more volatility into account in the prediction, whereas the forecast based on the AR(1)-fitted model seems to rely solely on the first moment of the process.

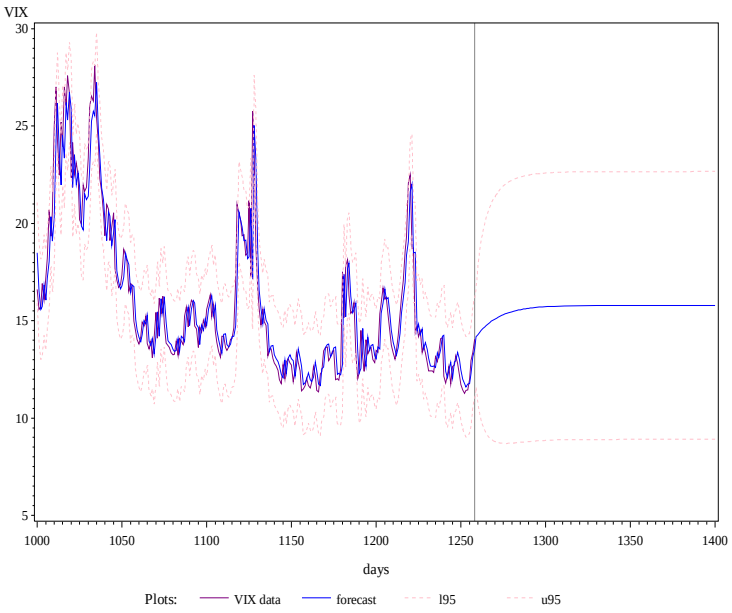


Figure 20: S&P500VIX 150-day forecast, programmed in SAS, by fitting an AR(1) model to the 5-year daily data.

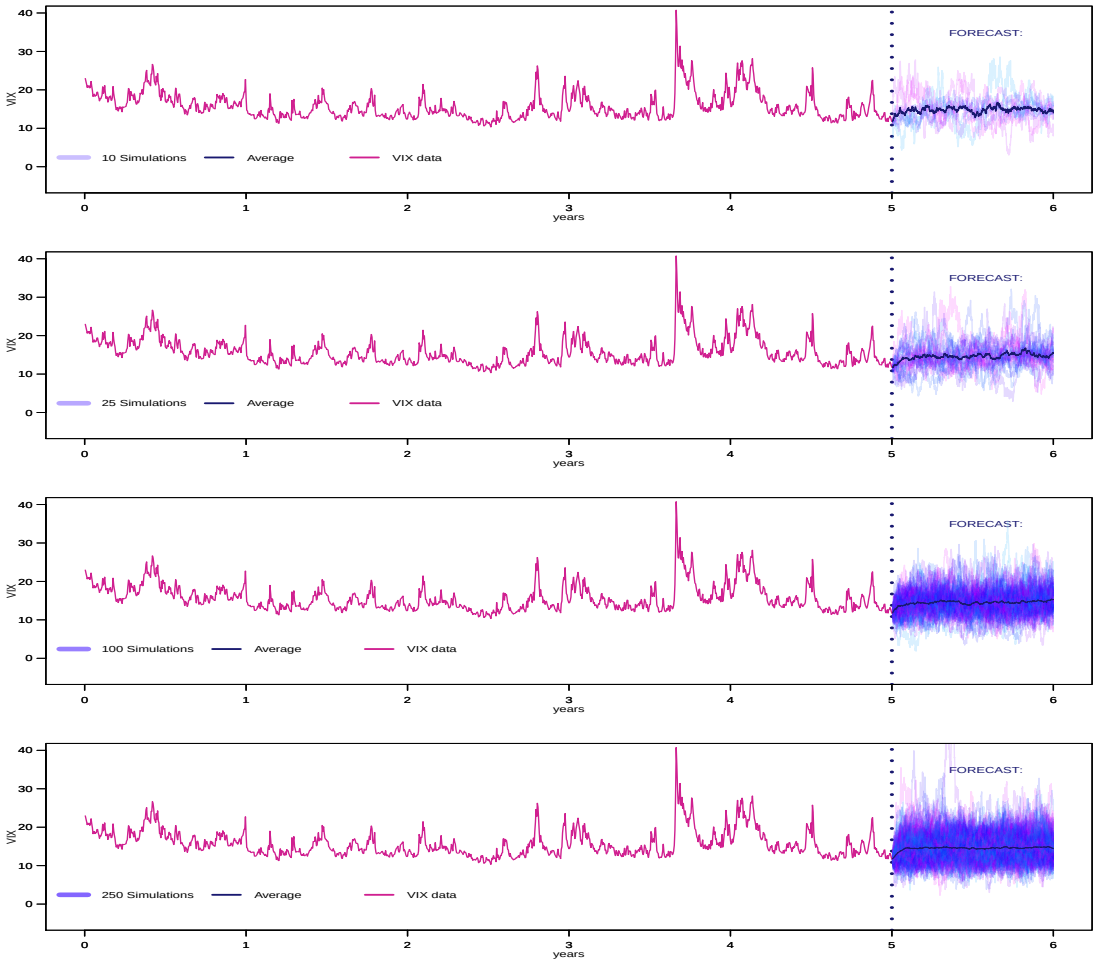


Figure 21: SP500VIX 250-day forward simulated prediction trajectory based on the mixed-effects CIR process moment-truncated saddlepoint transition density approximation's maximum likelihood estimates.

5 Conclusion

In the present paper an efficient closed-form transition density function approximation method for both scalar diffusion processes and mixed-effects diffusion processes have been developed, namely the moment-truncated saddlepoint transition density function approximation. It was shown that the moment-truncated saddlepoint transition density function approximation provides a significant improvement in approximation accuracy than that provided by the Hermite-series transition density function approximation. This approximation can efficiently be applied to infer on financial data and improve financial decisions, especially in the absence of a true transition density function. In the process of obtaining the moment-truncated saddlepoint transition density function approximation various insights into the CIR process has been made, for example that the CIR process is a mean-reversion model, a behavior which the moment-truncated saddlepoint transition density function approximation emulates this specific behavior with greater accuracy than the Hermite-series transition density function approximation. Not only could an appropriate and relatively accurate approximation be found for a scalar diffusion process, this paper has shown how to generalize such a scalar model to a mixed-effects model with a random-effect. The moment-truncated saddlepoint transition density function approximation can also be effectively applied to the mixed-effects diffusion process.

References

[1] Yacine Aït-Sahalia. Transition densities for interest rate and other nonlinear diffusions. *The Journal of Finance*, 54(4):1361–1395, 1999.

[2] Kalok C Chan, G Andrew Karolyi, Francis A Longstaff, and Anthony B Sanders. An empirical comparison of alternative models of the short-term interest rate. *The Journal of Finance*, 47(3):1209–1227, 1992.

[3] David A Chapman and Neil D Pearson. Is the short rate drift actually nonlinear? *The Journal of Finance*, 55(1):355–388, 2000.

[4] John C Cox, Jonathan E Ingersoll Jr, and Stephen A Ross. A theory of the term structure of interest rates. *Econometrica: Journal of the Econometric Society*, pages 385–407, 1985.

[5] Susanne Ditlevsen and Andrea De Gaetano. Mixed effects in stochastic differential equation models. *REVSTAT-Statistical Journal*, 3(2):137–153, 2005.

[6] Rick Durrett. *Probability: Theory and Examples*. Cambridge University Press, 2010.

[7] Patrick Flandrin, Pierre Borgnat, and Pierre-Olivier Amblard. From stationarity to self-similarity, and back: variations on the lamperti transformation. *Processes with Long-Range Correlations*, pages 88–117, 2003.

[8] Constantino Goutis and George Casella. Explaining the saddlepoint approximation. *The American Statistician*, 53(3):216–224, 1999.

[9] Desmond J Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 43(3):525–546, 2001.

[10] George J Jiang and John L Knight. A nonparametric approach to the estimation of diffusion processes, with an application to a short-term interest rate model. *Econometric Theory*, 13(5):615–645, 1997.

[11] Grigorios A Pavliotis. *Stochastic Processes and Applications*. Springer, 2016.

[12] Etienne AD Pienaar. *Non-linear Diffusion Processes and Applications*. PhD thesis, University of Cape Town, 2016.

[13] Gareth O Roberts and Osnat Stramer. On inference for partially observed nonlinear diffusion models using the metropolis–hastings algorithm. *Biometrika*, 88(3):603–621, 2001.

[14] Steven E Shreve. *Stochastic Calculus for Finance II: Continuous-time Models*, volume 11. Springer Science & Business Media, 2004.

[15] Johan Swart. Arbitrage theory- the partial differential equations approach. Unpublished manuscript, University of Pretoria, 2016.

[16] Melvin M Varughese. Parameter estimation for multivariate diffusion systems. *Computational Statistics & Data Analysis*, 57(1):417–428, 2013.

[17] George Neville Watson. *A Treatise on the Theory of Bessel Functions*. Cambridge University Press, 1995.

Appendix

A Fundamental results

The theorems, results and definitions given in this section, unless otherwise stated, are obtained and adapted from the works of [14].

A1 Stochastic processes

Let Ω be the non-empty set of all attainable elements, where E and its complement E^C possible events/-subsets.

Definition 5. A collection of subsets, ξ, of Ω is called a $\sigma - algebra$ or $\sigma - field$ if:

1. $\varnothing \in \Omega$ where $\varnothing$ is called the empty set,
2. if $E \in \xi$ then $E^C \in \xi$,
3. if $E_1, E_2, E_3, ...$ is a sequence of sets in ξ, then $\bigcup_{i=1}^{\infty} E_i \in \xi$.

Definition 6. $\mathbb{P}$ is called a probability measure function if it maps ξ into $[0, 1]$, where the following conditions hold:

1. $\mathbb{P}(\Omega) = 1$,
2. $\mathbb{P}(E) \geq 0 \; \forall \; E \in \xi$,
3. if $E_1, E_2, E_3, ...$ is a sequence of mutually disjoint subsets in ξ, then $\mathbb{P}\left[\bigcup_{i=1}^{\infty} E_i\right] = \sum_{i=1}^{\infty} \mathbb{P}(E_i)$.

Definition 7. For Ω, a non-empty finite set, a sequence of $\sigma - algebras$, $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, ...$, is called a filtration if $\mathcal{F}_k \subseteq \mathcal{F}_{k+1}$ for all $k = 0, 1, 2, ...$
The filtration can be denoted as $\{\mathcal{F}_t\}_{t=0}^{\infty}$. It can be written that $E \in \mathcal{F}_t$, if it is known that at time t whether or not event E has occurred. If our time-dimension is finite, i.e $[0, t^*]$ then $\mathcal{F}_{t^*} = \mathcal{F}$

Define$\mathbb{R}$ to be the set of all real numbers.

Definition 8. For Ω a non-empty finite set, and ξ a $\sigma - algebra$ of all possible subsets of Ω, a random variable defined as a function mapping Ω onto $\mathbb{R}$, i.e X is a random variable if it can be written as $X : \Omega \to \mathbb{R}$.

A probability space can now be defined.

Definition 9. $(\Omega, \mathcal{F}, \mathbb{P})$ is called a probability space, where the elements are defined as

1. A non-empty set, Ω, called the sample space which include all possible outcomes,
2. $\mathcal{F}$ is a $\sigma - algebra$, consisting of subsets of Ω,
3. $\mathbb{P}$ is called a probability measure function on $(\Omega, \mathcal{F})$, where $(\Omega, \mathcal{F})$ is a measurable space.

Stochastic processes are now introduced. Let S be the state-space consisting of all possible states, and T^* an ordered set known as the time-dimension over which the stochastic process is defined.

Definition 10. A collection of random variables $\{X_t : t \in T^*\}$, defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, is called a stochastic process.

For an individual occurrence or sample path of Ω it is given that $\omega \in \Omega$, s.t. $X_t(\omega) \in \mathbb{R}$.

Definition 11. A $\sigma - algebra$ is called a Borel $\sigma - algebra$ if it is the $\sigma - algebra$ which contains all the open intervals in $\mathbb{R}$, and is denoted $\mathcal{B}(\mathbb{R})$.

Borel-measurable functions in $\mathbb{R}$ are of interest.

Definition 12. For $g : \mathbb{R} \to \mathbb{R}$, g is Borel-measurable if $\{y \in \mathbb{R} : g(y) \in G\} \in \mathcal{B}(\mathbb{R})$ when $G \in \mathcal{B}(\mathbb{R})$.

A very important class of stochastic processes are Markov processes:

Definition 13. Consider only Borel-measurable functions in $\mathbb{R}$ and consider $\{\mathcal{F}_t\}_{t=0}^n$ a filtration under $\mathcal{F}$. The stochastic process $\{X_t\}_{t=0}^n$, defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, is said to be Markov if:

1. $\{X_t\}$ is adapted to $\{\mathcal{F}_t\}$
2. $\{X_t\}$ is said to have the Markov Property if the distribution of X_{t+1}, conditioned upon $\mathcal{F}_t$ is equal in distribution of X_{t+1}, conditioned upon X_t, for all $t = 0, 1, 2, ..., n-1$; i.e. $\mathbb{P}(X_{t+1} \in E|\mathcal{F}_t) = \mathbb{P}(X_{t+1} \in E|X_t)$, where $\mathcal{F}_t$ depends on $X_0, X_1, ..., X_{t-1}, X_t$.

Throughout this text, S will represent the state-space and T^* the time-dimension over which a given diffusion process is defined. In this document our continuous-time Markov processes, diffusion processes, will need to be discretized, therefore the interest will lie in Markov chains, which can be seen as stochastic processes with both a discrete time-dimension and discrete state-space. The definitions of transition probabilities, under the Markov property, can now, according to [11], be introduced:

Definition 14. The transition probability for a transition in the stochastic process, in a discrete state-space, from $X_s = x$ at time s to $X_t = y$ at time t can be defined as:

Time-in homogeneous case: transition probabilities are time-dependent

$\mathbb{P}(X_t = y|X_s = x) = p_{xy}(s,t) = p(y,t|x,s)$

Time-homogeneous case: transition probabilities are invariant to shifts in time:

$\mathbb{P}(X_t = y|X_s = x) = \mathbb{P}(X_{t+k} = y|X_{s+k} = x) = p_{xy}(t-s)$ for all k s.t $s+k, t+k \in T$.

Lemma 15. *The Chapman-Kolmogorov equation (for a time-in homogeneous case) is given as:*

$p_{xy}(s,t) = \sum_{\forall \tau} p_{x\tau}(s,r)p_{\tau y}(r,t)$

Now, by [**?**], consider the case of a continuous state-space, and continuous time-dimension, s.t. $t \in [s,T]$.

Definition 16. $\mathbb{P}(X_t \in \zeta|X_s = x) = p(\zeta,t|x,s)$ is called the transition function, which is a probability measure on A s.t. $\mathbb{P}(X_t \in A|X_s = x) = 1$. $p(\zeta,t|x,s)$ is Borel-Measurable ($\mathcal{B}(A)$) in x where ζ, s and t are fixed and satisfies to the continuous-space Chapman Kolmogorov Equation:

$p(\zeta,t|x,s) = \int_A p(\zeta,t|x,u)p(dy,u|x,s)$

Fact 17. *Given that the all information regarding $\mathcal{F}_t$ is known, it is said that X_t is $\mathcal{F}_t - adapted$ if the value of the random variable X_t is known at time t*

Definition 18. Strictly stationary process: according to [11] for a stochastic process $\{X_t\}$ to be strictly stationary the joint distribution of $X_{t_1}, X_{t_2}, ..., X_{t_n}$ should be the same as the joint distribution of $X_{t_{1+k}}, X_{t_{2+k}}, ..., X_{t_{n+k}}$, for all k s.t. $t_{i+k} \in T$.

A2 Stochastic calculus

The results and theory given in this section is a combination based on the works of [15], [9] and [14]. Consider a standard Brownian motion or Wiener Process to be the continuous counterpart of a Random Walk.

Lemma 19. $\mathbb{E}[W_s W_t] = min\{s,t\}$

Proof. Assume that $s \leq t$, then $[0,s]$ and $[0,t]$ is overlapping and hence W_s and W_t are dependent. Since $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$ if X and Y are independent random variables, let $W_s W_t = W_s(W_t - W_s + W_s) = W_s(W_t - W_s) + W_s^2$. Since the distribution of $W_t - W_s$ and W_{t-s} is the same, i.e $N(0, t-s)$, the intervals $[0,s]$ and $[0,t-s]$ are non-overlapping and hence W_s and W_{t-s} are independent. Therefore

$$\begin{aligned}
\mathbb{E}[W_s W_t] &= \mathbb{E}[W_s(W_t - W_s + W_s)] \\
&= \mathbb{E}[W_s(W_t - W_s) + W_s^2] \\
&= \mathbb{E}[W_s(W_t - W_s)] + \mathbb{E}[W_s^2] \\
&= \mathbb{E}[W_s]\mathbb{E}[(W_t - W_s)] + VAR(W_s) + \big(E[W_s]\big)^2 \\
&= s
\end{aligned}$$

Since $s \leq t$ it is clear that $\mathbb{E}[W_s W_t] = s = min\{s,t\}$.

Similarly if it is assumed that $t \leq s$, it follows that $\mathbb{E}[W_t W_s] = t = min\{t,s\}$. Therefore the result follows that $\mathbb{E}[W_s W_t] = min\{s,t\}$ □

Suppose $\{X_n\}_{n=0}^{\infty}$ is a sequence of random variables, which in turn is also random.

Proposition 20. *X_n converges in mean square to X if* $\lim_{n\to\infty} \mathbb{E}[(X_n - X)^2] = 0$.

The stochastic integral and its meaning can now be introduced. Let X_t be a process be driven by Brownian motion, then a stochastic integral can be denoted as $\int_0^t X_s(\omega)dW_s(\omega)$, where ω denotes the specific path followed by the stochastic process. If this path is assumed to be known, which is assumed for this section, then $\int_0^t X_s(\omega)dW_s(\omega)$ is equivalent to $\int_0^t X_s dW_s$. Let f be a continuous real valued function, s.t. $X_s = f(W_s)$, i.e. X_t is a stochastic process driven by Brownian motion. Let ϱ_n be a partitioning of $[0,t]$ s.t $\varrho_n : 0 = s_0 < s_1 < ... < s_{n-1} < s_n = t$, where the mesh of the partitioning is given by: mesh(ϱ_n) $= \max_{1\leq i\leq n}(s_i - s_{i-1})$. Consider the series:

$$\Upsilon_n = \sum_{i=0}^{n-1} f(W_{s_j})(W_{s_{j+1}} - W_{s_j}), \text{ where } s_j \text{ is the left end-point of the interval.}$$

Taking the mean square, which will always exist if $\mathbb{E}\Big(\int_0^t |f(W_s)|^2 ds\Big) < \infty$, and noting that $f(W_s)$ has continuous sample paths since f is continuous. It follows that:

$$\lim_{n\to\infty} \Upsilon_n = \lim_{n\to\infty} \sum_{i=0}^{n-1} f(W_{s_j})(W_{s_{j+1}} - W_{s_j}) = \int_0^t X_s dW_s \ .$$

Theorem 21. *Let f be a continuous function, where $f(W_s)$ is a stochastic process driven by Brownian motion. If $\int_0^t \mathbb{E}[f(W_s)^2]ds < \infty$then:*

1. $\mathbb{E}\Big(\int_0^t f(W_s)dW_s\Big) = 0$,
2. $\mathbb{E}\Big(\Big|\int_0^t f(W_s)dW_s\Big|^2\Big) = \int_0^t \Big[\mathbb{E}|f(W_s)|^2\Big]dW_s$.

The concept of stochastic differential equations and *Itô* processes can now be discussed.

Definition 22. A stochastic process X_t is called an *Itô* process if the process can be given in the form of:

$$X_t = X_0 + \int_0^t \Gamma_s ds + \int_0^t \Psi_s dW_s \tag{87}$$

s.t. Γ and Ψ are $\mathcal{F}_t - adapted$ processes driven by Brownian motion and where the following conditions hold:

$$\int_0^t |\Gamma_s|^2 ds < \infty and \int_0^t \mathbb{E}[\Psi_s^2]ds < \infty.$$

Now if f is assumed to be a real-valued function, which is integrable. Let $F(t)$ be defined as follows:

$$F(t) = F(0) + \int_0^t f(s)ds \tag{88}$$

Differentiating Equation 88 with respect to t:

$$\frac{d}{dt}F(t) = \frac{d}{dt}F(0) + \frac{d}{dt}\int_0^t f(s)ds,$$

it follows from the Fundamental theorem of Calculus that:

$$\frac{d}{dt}F(t) = f(t), \text{ or}$$

$$F'(t) = f(t).$$

Equation 87 can now be written in stochastic differential form:

$$dX_t = \Gamma_t dt + \Psi_t dW_t \tag{89}$$

Remark 23. The short-hand notion will be used throughout the text, but it is important to take note that: $X_s \equiv X(s, W_s)$, $\Gamma_s \equiv \Gamma(s, W_s)$ and $\Psi_s \equiv \Psi(s, W_s)$.

A fundamental result in stochastic calculus can now be introduced, namely *Itô*'s Lemma:

Theorem 24. *Itô's Lemma for a one-dimensional Brownian motion: for W_t a Brownian motion defined on $[0, T]$ and real-valued function $g(x)$, which is twice differentiable, then for all $t \leq T$, it follows that:*

$$g(W_t) = g(0) + \frac{1}{2}\int_0^t g"(W_s)ds + \int_0^t g'(W_s)dW_s$$

Proof. Consider a partition of $[0, t] \subseteq [0, T]$ namely $\varrho_n : 0 = t_0 < t_1 < ... < t_{n-1} < t_n = t$. Let

$$g(W_t) = g(0) + \sum_{j=0}^{n-1}[g(W_{t_{j+1}}) - g(W_{t_j})] \tag{90}$$

As an implication of Taylor's Theorem it is given that:

$$g(W_{t_{j+1}}) - g(W_{t_j}) = g'(W_{t_j})(W_{t_{j+1}} - W_{t_j}) + \frac{1}{2}g"(W^*_{t_j})(W_{t_{j+1}} - W_{t_j})^2 \tag{91}$$

for some $W^*_{t_j} \in (W_{t_j}, W_{t_{j+1}})$. Through the substitution of Equation 6 into Equation 5, Equation 7 is obtained:

$$g(W_t) = g(0) + \sum_{j=0}^{n-1} g'(W_t)(W_{t_{j+1}} - W_{t_j}) + \frac{1}{2}\sum_{j=0}^{n-1} g"(W^*_{t_j})(W_{t_{j+1}} - W_{t_j})^2 \tag{92}$$

For $\text{mesh}(\varrho_n) = \underset{1\leq i\leq n}{max}(t_i - t_{i-1})$ and letting $\text{mesh}(\varrho_n) \to 0$, then taking the limit as n tends to infinity over Equation 92, the required result is obtained:

$$g(W_t) = g(0) + \frac{1}{2}\int_0^t g"(W_s)ds + \int_0^t g'(W_s)dW_s.$$

□

The general version of *Itô*'s lemma can now be given:

Theorem 25. *Itô's Lemma for a one-dimensional Itô process: for a one-dimensional Itô process, X_t, satisfying the following stochastic differential equation:*

$$dX_t = a_t dt + b_t dW_t \tag{93}$$

For $g(t, X_t) : [0, \infty) \times \mathbb{R} \to \mathbb{R}$ and letting $Y_t = g(t, X_t)$ then

$$dY_t = \frac{\partial}{\partial t} g(t, X_t) dt + \frac{\partial}{\partial X_t} g(t, X_t) dX_t + \frac{1}{2} \frac{\partial^2}{\partial X_t^2} g(t, X_t)(dX_t)^2 \tag{94}$$

Substituting Equation 93 into Equation 94 yields:

$$dY_t = \left(\frac{\partial}{\partial t} g(t, X_t) + \frac{\partial}{\partial t} g(t, X_t) a_t + \frac{1}{2} \frac{\partial^2}{\partial X_t^2} g(t, X_t) b_t^2 \right) dt + \frac{\partial}{\partial X_t} g(t, X_t) b_t dW_t \tag{95}$$

Remark 26. $dt dW_t = (dt)^2 = 0$ and $(dW_t)^2 = dt$

B Algorithms

B2 R algorithms

Algorithm 1 Trajectory, Euler-Maruyama scheme, perspective plot, theoretical density, Hermite-series transition density function approximation and moment truncation approximation of the scalar CIR diffusion process.

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)*dt + sigma(Xt,t)*dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt
rm(list=ls(all=TRUE))

#Seed:
set.seed(7)

#Parameters
s              = 0
t              = 5
Xs             = 2.75
alpha          = 0.8
beta           = 3
sigma          = 0.25
delta_t        = 0.01    #step length
startingstate  = 0
endstate       = 5
numbsims       = 10000
timespace      = seq(s,t,delta_t)
statespace     = seq(startingstate,endstate,delta_t)

par(ps=10,cex.lab=1,cex.axis=1,mar=c(3.5,3.5,3.5,2.5), mgp=c(2.5, 1, 0), las=1)

#Simulating the trajectory

CIR_trajectory = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
{

  timeseq        = (seq(s,t,delta_t))
  datamatrix     = matrix(0,nrow = length(timeseq), ncol = 1)
  Z1             = rnorm(1,mean = 0, sd = sqrt(delta_t))
  Xt             = Xs + alpha*(beta-Xs)*delta_t + sigma*sqrt(Xs)*Z1
  datamatrix[1] = Xt
```

```
35
36    for(i in 2:length(timeseq))
37    {
38      dWt            = rnorm(1,mean = 0, sd = sqrt(delta_t))
39      Xtplus1        = Xt  + alpha*(beta-Xt)*delta_t + sigma*sqrt(Xt)*dWt
40      Xt             = Xtplus1
41      datamatrix[i] = Xtplus1
42    }
43
44    X = datamatrix
45
46    plot(X~seq(s,t,delta_t),type ='l', col = "royalblue4",xlab="t",ylab = "Xt")
47  }
48
49  trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
50
51  #Perspective Plot
52
53  CIR_perpective = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
54  {
55    timespace  = seq(s,t,delta_t)
56    statespace = seq(startingstate,endstate,delta_t)
57
58    datamatrix = matrix(0,length(timespace),length(statespace))
59
60    for (t in s:length(timespace))
61    {
62      for (state in startingstate:length(statespace))
63      {
64        c                    = (2*alpha)/((sigma^2)*(1-exp(-alpha*(timespace[t]-s))))
65        u                    = c*Xs*exp(-alpha*(timespace[t]-s))
66        v                    = c*statespace[state]
67        q                    = 2*alpha*beta/(sigma^2) - 1
68        besselparameter      = 2*(u*v)^(0.5)
69        logbessel            = log(besselI(besselparameter,q,expon.scaled = TRUE))+
              besselparameter
70        logfXt_t             = log(c) - (u+v) + (q/2)*log(v/u) + logbessel
71        datamatrix[t,state]  = exp(logfXt_t)
72      }
73
74    }
75
76
```

```
77    persp(timespace,statespace,datamatrix, col = "royalblue4",xlab="t", ylab="Xt",zlab="
          Density", border = NA, shade = 0.9 , theta = 45, phi = 35, r = 35, ticktype = "
          detailed")
78  }
79
80  perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
81
82  #Euler-Maruyama Scheme
83
84  CIR_EM = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
85  {
86    mufunc = function(Xt,t)
87    {
88      return(alpha*(beta - Xt))
89    }
90
91    sigfunc = function(Xt,t)
92    {
93      return(sigma*sqrt(Xt))
94    }
95
96    histfunc = function(Xs,s,t,delta_t,numbsims)
97    {
98
99      Xt         = rep(Xs,numbsims)
100     timespace = seq(s,t,delta_t)
101
102     for(i in 2:length(timespace))
103     {
104       dWt   = sqrt(delta_t)*rnorm(numbsims)
105       Xt    = Xt + mufunc(Xt, timespace[i])*delta_t + sigfunc(Xt,timespace[i])*dWt
106       hist(Xt, freq = FALSE, col = 'royalblue4', breaks = 50, ylim = c(0,1.4), main = NA,
              border = "mediumpurple", xlab = "Xt", ylab = "Density")
107     }
108
109     return(list(Xt=Xt,time = t))
110
111   }
112
113   plot = histfunc(Xs,s,t,delta_t,numbsims)
114 }
115
116 EM_plot = CIR_EM(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
```

```
117
118 #Theoretical density (Sahalia 1999)
119
120 CIR_theoretical1 = function(s,t,Xs,Xt,alpha,beta,sigma)
121 {
122   c               = (2*alpha)/((sigma^2)*(1-exp(-alpha*(t-s))))
123   u               = c*Xs*exp(-alpha*(t-s))
124   v               = c*Xt
125   q               = 2*alpha*beta/(sigma^2) - 1
126   besselparameter = 2*(u*v)^(0.5)
127   besselfunction  = besselI(besselparameter,q,expon.scaled = TRUE)
128   logbessel       = log(besselI(besselparameter,q,expon.scaled = TRUE))+besselparameter
129   logfXt          = log(c) - (u+v) + (q/2)*log(v/u) + logbessel
130   return(exp(logfXt))
131 }
132
133 Xt = statespace
134 plot_theoretical1 = CIR_theoretical1(s,t,Xs,Xt,alpha,beta,sigma)
135
136 lines(plot_theoretical1~Xt,col = "black",lwd = 3)
137
138 #Hermite-series transition density function approximation:
139
140 CIR_Hermite-series= function(s,t,Xs,Xt,alpha,beta,sigma,K)
141 {
142
143     invsigxt    = 1/(sigma*sqrt(Xt))
144     gamxt       = ((2*sqrt(Xt))/sigma) # = Yt
145     gamxs       = ((2*sqrt(Xs))/sigma) # = Ys
146     p1          = 1/sqrt(2*pi*(t-s))
147     p2          = exp(-((gamxt-gamxs)^2)/(2*(t-s))-(alpha*(gamxt^2)/4)+(alpha*(gamxs^2)
          /4))*(gamxt^(-0.5+2*alpha*beta/sigma^2))*(gamxs^(0.5-2*alpha*beta/sigma^2))
148     p           = p1*p2
149     c1          = -1/(24*gamxt*gamxs*sigma^4)*(48*(alpha*beta)^2-48*alpha*beta*(sigma^2)
          +9*(sigma^4)+gamxt*(alpha^2)*(sigma^2)*gamxs*(-24*beta+(gamxt^2)*(sigma^2))+(
          gamxt^2)*(alpha^2)*(sigma^4)*(gamxs^2)+gamxt*(alpha^2)*(sigma^4)*(gamxs^3))
150     hermitedens = invsigxt*p
151
152     if (K>0)
153       {
154        hermitedens = invsigxt*p*(1+(t-s)*c1)
155       }
156
```

```
  return(hermitedens)
}

K              = 1
Xt             = statespace
plot_Hermite-series = CIR_hermite(s,t,Xs,Xt,alpha,beta,sigma,K)

lines(plot_hermite~Xt,lty = 3,col = "gray47", lwd = 3)

#Method of Moment Truncation

#Theoretical Moments
del   = Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs)+beta*(beta + ((sigma^2)/(2*
    alpha)))+2*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
gamma = alpha*(Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs))+3*alpha*beta*(beta + ((
    sigma^2)/(2*alpha)))+4*alpha*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
kappa = 2*(alpha^2)*beta*(beta + ((sigma^2)/(2*alpha)))
A     = kappa/(6*alpha^3)
C     = -4*((1/(4*alpha^2))*(gamma-9*A*alpha^2)-(1/(2*alpha))*(del-3*alpha*A))
B     = (1/(2*alpha))*(del-3*alpha*A-alpha*C)
D     = -A-B-C

gamma_star  = Xs^3 + 3*(alpha*beta + sigma^2)*(A + B + C + D)
lambda_star = 3*alpha*Xs^3 + 3*(alpha*beta + sigma^2)*(6*alpha*A + 5*alpha*B + 4*alpha*C
    + 3*alpha*D)
omega_star  = 2*(alpha^2)*Xs^3 + 3*(alpha*beta + sigma^2)*(11*(alpha^2)*A + 6*(alpha^2)*B
     + 3*(alpha^2)*C + 2*(alpha^2)*D)
nu_star     = 3*(alpha*beta + sigma^2)*(6*A*alpha^3)

E  = nu_star/(24*alpha^4)
I  = (-1/(6*alpha^3))*(((omega_star-(13*nu_star/(12*alpha)))-12*(alpha^2)*(gamma_star-(nu
    _star/(24*alpha^3))))-4*alpha*((lambda_star-(3*nu_star/(8*alpha^2)))-7*alpha*((gamma_
    star-(nu_star/(24*alpha^3))))))
H  = (1/(2*alpha^2))*(((lambda_star-(3*nu_star/(8*alpha^2)))-7*alpha*((gamma_star-(nu_
    star/(24*alpha^3))))) - 6*(alpha^2)*I)
G  = (-1/alpha)*((gamma_star-(nu_star/(24*alpha^3)))+ 2*alpha*H + 3*alpha*I)
FF = -E - G - H - I

theomoment1 = Xs*exp(-alpha*(t-s)) + beta*(1 - exp(-alpha*(t-s)))
theomoment2 = (Xs^2)*exp(-2*alpha*(t-s)) + (beta + (sigma^2)/(2*alpha))*(beta + 2*(Xs-
    beta)*exp(-alpha*(t-s)) + (beta - 2*Xs)*exp(-2*alpha*(t-s)))
theomoment3 = (Xs^3)*exp(-3*alpha*(t-s)) + (3*alpha*beta+3*sigma^2)*(A + B*exp(-alpha*(t-
    s)) + C*exp(-2*alpha*(t-s)) + D*exp(-3*alpha*(t-s)))
```

```
theomoment4 = (Xs^4)*exp(-4*alpha*(t-s)) + (4*alpha*beta + 6*sigma^2)*(E + FF*exp(-1*
    alpha*(t-s)) + G*exp(-2*alpha*(t-s)) + H*exp(-3*alpha*(t-s)) + I*exp(-4*alpha*(t-s)))

#theoretical Cumulants
theocumulant1 = theomoment1
theocumulant2 = theomoment2-(theomoment1)^2
theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
theocumulant4 = -6*(theomoment1^4) + 12*(theomoment1^2)*(theomoment2) - 3*(theomoment2^2)
     - 4*theomoment1*theomoment3 + theomoment4

X = statespace

#saddlepoint transition density approximation
s = (1/theocumulant3)*(sqrt(theocumulant2^2 - 2*theocumulant3*(theocumulant1-X)) -
    theocumulant2)
Ksapprox = theocumulant1*s + theocumulant2*((1/2)*s^2) + theocumulant3*((1/6)*s^3) +
    theocumulant4*((1/24)*s^4)
Ks2approx = theocumulant2 + theocumulant3*s + 0.5*theocumulant4*s^2

saddle_pt_approx = exp(Ksapprox-s*X)*sqrt(1/(2*pi*Ks2approx))
#print(saddle_pt_approx)

lines(saddle_pt_approx~statespace,lty = 3,col = "deepskyblue1", lwd = 3)

#Transition Density Plots
labels = c("Theoretical", "Hermite-seriesapprox", "Euler-Maruyama", "Saddle pt approx")
legend("topright", title = NA,labels,lty = c(1,3,2,3), lwd = c(3,3,6,3) ,col=c("black", "
    gray47","royalblue4","deepskyblue1"), bty = 'n')
box()
```

Algorithm 2 Simulated trajectories indicating the effect of parameter changes for the CIR process

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)*dt + sigma(Xt,t)*dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt

rm(list=ls(all=TRUE))

#Seed:
set.seed(7)

#Parameters
s              = 0
```

```
t              = 5
Xs             = 2.75
delta_t        = 0.01    #step length
startingstate  = 0
endstate       = 5
numbsims       = 10000
timespace      = seq(s,t,delta_t)
statespace     = seq(startingstate,endstate,delta_t)


#Simulating the trajectory

CIR_trajectory = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
{

  timeseq        = (seq(s,t,delta_t))
  datamatrix     = matrix(0,nrow = length(timeseq), ncol = 1)
  Z1             = rnorm(1,mean = 0, sd = sqrt(delta_t))
  Xt             = Xs + alpha*(beta-Xs)*delta_t + sigma*sqrt(Xs)*Z1
  datamatrix[1] = Xt

  for(i in 2:length(timeseq))
  {
    dWt            = rnorm(1,mean = 0, sd = sqrt(delta_t))
    Xtplus1        = Xt  + alpha*(beta-Xt)*delta_t + sigma*sqrt(Xt)*dWt
    Xt             = Xtplus1
    datamatrix[i] = Xtplus1
  }

  X = datamatrix

  plot(X~seq(s,t,delta_t),type ='l', col = rainbow(1, start = runif(1,0.55,0.8), end =
      runif(1,0.55,0.7), alpha = 1) ,xlab="t",ylab = "Xt", ylim=c(0,5))
}


par(mfrow=c(3,4),ps=9,cex.lab=1,cex.axis=0.75,mar=c(1, 1, 2, 1), mgp=c(1.5, 0.8, 0), las
    =1)
#Alpha change - mean reversion speed
beta = 3
sigma = 0.25

alpha = -1
```

```
trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(alpha == .(alpha)))
alpha = 0.2
trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(alpha == .(alpha)))
alpha = 0.8
trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(alpha == .(alpha)))
alpha = 1.4
trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(alpha == .(alpha)))

#Beta change - to where mean revert
alpha = 0.8
sigma = 0.25

beta = 0
trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(beta == .(beta)))
beta = 2
trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(beta == .(beta)))
beta = 3
trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(beta == .(beta)))
beta = 4
trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(beta == .(beta)))

#Sigma change - diffusion coefficient
alpha = 0.8
beta  = 3

sigma = 0
trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(sigma == .(sigma)))
sigma = 0.15
trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(sigma == .(sigma)))
sigma = 0.25
trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(sigma == .(sigma)))
sigma = 0.5
```

```
trajectory_plot = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(sigma == .(sigma)))
```

Algorithm 3 Perspective plots indicating the effect of parameter changes for the CIR process.

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)*dt + sigma(Xt,t)*dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt

rm(list=ls(all=TRUE))

#Seed:
set.seed(7)

#Parameters
s              = 0
t              = 5
Xs             = 2.75
delta_t        = 0.01    #step length
startingstate  = 0
endstate       = 5
numbsims       = 1000
timespace      = seq(s,t,delta_t)
statespace     = seq(startingstate,endstate,delta_t)


#Perspective Plot

CIR_perpective = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
{
  timespace  = seq(s,t,delta_t)
  statespace = seq(startingstate,endstate,delta_t)

  datamatrix = matrix(0,length(timespace),length(statespace))

  for (t in s:length(timespace))
  {
    for (state in startingstate:length(statespace))
    {
      c                    = (2*alpha)/((sigma^2)*(1-exp(-alpha*(timespace[t]-s))))
      u                    = c*Xs*exp(-alpha*(timespace[t]-s))
      v                    = c*statespace[state]
      q                    = 2*alpha*beta/(sigma^2) - 1
```

```
      besselparameter      = 2*(u*v)^(0.5)
      logbessel            = log(besselI(besselparameter,q,expon.scaled = TRUE))+
          besselparameter
      logfXt_t             = log(c) - (u+v) + (q/2)*log(v/u) + logbessel
      datamatrix[t,state]  = exp(logfXt_t)
    }

  }


  persp(timespace,statespace,datamatrix, col = rainbow(1, start = runif(1,0.55,0.8), end
      = runif(1,0.55,0.7), alpha = 1),xlab="t", ylab="Xt",zlab="density", border = NA,
      shade = 0.9 , theta = 45, phi = 35, r = 35, ticktype = "detailed")
}


par(mfrow=c(3,4),ps=9,cex.lab=1,cex.axis=0.6,mar=c(0.25, 0.25, 2, 0.25), mgp=c(1.5, 0.8,
    0), las=1)
#Alpha change - mean reversion speed
beta = 3
sigma = 0.25

alpha = -1
perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(alpha == .(alpha)))
alpha = 0.2
perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(alpha == .(alpha)))
alpha = 0.8
perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(alpha == .(alpha)))
alpha = 1.4
perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(alpha == .(alpha)))

#Beta change - to where mean revert
alpha = 0.8
sigma = 0.25

beta = 0
perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(beta == .(beta)))
beta = 2
```

```
perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(beta == .(beta)))
beta = 3
perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(beta == .(beta)))
beta = 4
perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(beta == .(beta)))


#Sigma change - diffusion coefficient
alpha = 0.8
beta  = 3

sigma = 0.025
perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(sigma == .(sigma)))
sigma = 0.15
perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(sigma == .(sigma)))
sigma = 0.25
perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(sigma == .(sigma)))
sigma = 0.5
perspective_plot = CIR_perpective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
title(main=bquote(sigma == .(sigma)))
```

Algorithm 4 Hermite-series density approximation of order $K = 0, 1, 2$ for the scalar CIR process.

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)*dt + sigma(Xt,t)*dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt

rm(list=ls(all=TRUE))


#Seed:
set.seed(7)

#Parameters
s              = 0
t              = 5
Xs             = 2.75
```

```
alpha          = 0.8
beta           = 3
sigma          = 0.25
delta_t        = 0.01    #step length
startingstate  = 0
endstate       = 5
numbsims       = 10000
timespace      = seq(s,t,delta_t)
statespace     = seq(startingstate,endstate,delta_t)


par(mfrow=c(2,3), ps=10,cex.lab=1,cex.axis=1,mar=c(3.5,3.5,3.5,2.5), mgp=c(2.5, 1, 0),
    las=1)

#Hermite-series transition density function approximation:

CIR_main = function(Xt,K)
{

  Xt = statespace

  #Theoretical density (Sahalia 1999)

  CIR_theoretical1 = function(s,t,Xs,Xt,alpha,beta,sigma)
  {
    c               = (2*alpha)/((sigma^2)*(1-exp(-alpha*(t-s))))
    u               = c*Xs*exp(-alpha*(t-s))
    v               = c*Xt
    q               = 2*alpha*beta/(sigma^2) - 1
    besselparameter = 2*(u*v)^(0.5)
    besselfunction  = besselI(besselparameter,q,expon.scaled = TRUE)
    logbessel       = log(besselI(besselparameter,q,expon.scaled = TRUE))+besselparameter
    logfXt          = log(c) - (u+v) + (q/2)*log(v/u) + logbessel
    return(exp(logfXt))
  }


  plot_theoretical1 = CIR_theoretical1(s,t,Xs,Xt,alpha,beta,sigma)

  plot(plot_theoretical1~Xt,col = "black", type = "l" , lwd = 2, ylab = "Density" ,ylim =
       c(0,2),xlab="t",xlim=c(startingstate,endstate))


```

```
CIR_Hermite-series= function(s,t,Xs,Xt,alpha,beta,sigma,K)
{

  Xt           = statespace

  invsigxt     = 1/(sigma*sqrt(Xt))
  gamxt        = ((2*sqrt(Xt))/sigma) # = Yt
  gamxs        = ((2*sqrt(Xs))/sigma) # = Ys
  p1           = 1/sqrt(2*pi*(t-s))
  p2           = exp(-((gamxt-gamxs)^2)/(2*(t-s))-(alpha*(gamxt^2)/4)+(alpha*(gamxs^2)/4))
      *(gamxt^(-0.5+2*alpha*beta/sigma^2))*(gamxs^(0.5-2*alpha*beta/sigma^2))
  p            = p1*p2
  c1           = -1/(24*gamxt*gamxs*sigma^4)*(48*(alpha*beta)^2-48*alpha*beta*(sigma^2)
      +9*(sigma^4)+gamxt*(alpha^2)*(sigma^2)*gamxs*(-24*beta+(gamxt^2)*(sigma^2))+(gamxt
      ^2)*(alpha^2)*(sigma^4)*(gamxs^2)+gamxt*(alpha^2)*(sigma^4)*(gamxs^3))
  c2           = (1/(576*gamxt^2*gamxs^2))*(9*(256*alpha^4*beta^4-512*alpha^3*beta^3*sigma
      ^2+224*alpha*beta*sigma^6-15*sigma^8)+6*gamxt*alpha^2*sigma^2*(-24*beta+gamxt^2*
      sigma^2)*(16*beta^2*alpha^2-16*beta*alpha*sigma^2+3*sigma^4)*gamxs+gamxt^2*alpha^2*
      sigma^4*(672*beta^2*alpha^2-48*beta*alpha*(2+gamxt^2*alpha)*sigma^2+(-6+gamxt^4*
      alpha^2)*sigma^4)*gamxs^2+2*gamxt*alpha^2*sigma^4*(48*beta^2*alpha^2-24*beta*alpha
      *(2+gamxt^2*alpha)*sigma^2+(9+gamxt^4*alpha^2)*sigma^4)*gamxs^3+3*gamxt^2*alpha^4*
      sigma^6*(-16*beta+gamxt^2*sigma^2)*gamxs^4+2*gamxt^3*alpha^4*sigma^8*gamxs^5+gamxt
      ^2*alpha^4*sigma^8*gamxs^6)


  if (K==0)
  {
    hermitedens = invsigxt*p
  }

  if (K==1)
  {
    hermitedens = invsigxt*p*(1+(t-s)*c1)
  }

  if (K==2)
  {
    hermitedens = invsigxt*p*(1+(t-s)*c1 + (((t-s)^2)/2)*c2)
  }

  hermite_plot = lines(hermitedens~Xt, lty = 3, col = "azure4", lwd = 3)

  return(hermite_plot)
```

```
89
90  }
91
92  call_Hermite-series = CIR_hermite(s,t,Xs,Xt,alpha,beta,sigma,K)
93  return(call_hermite)
94
95  }
96
97
98  call_main = CIR_main(Xt,0)
99  labels = c("Theoretical", "Hermite: K=0")
100 legend("top", inset = -0.095, title = NA,labels,lty = c(1,3), lwd = c(2,3) , col=c("black
        ","azure4"), bty = 'n')
101
102 call_main = CIR_main(Xt,1)
103 labels = c("Theoretical", "Hermite: K=1")
104 legend("top", inset = -0.095, title = NA,labels,lty = c(1,3), lwd = c(2,3) , col=c("black
        ","azure4"), bty = 'n')
105
106 call_main = CIR_main(Xt,2)
107 labels = c("Theoretical", "Hermite: K=2")
108 legend("top", inset = -0.095, title = NA,labels,lty = c(1,3), lwd = c(2,3) , col=c("black
        ","azure4"), bty = 'n')
109
110
111 CIR_hermite_diff = function(s,t,Xs,Xt,alpha,beta,sigma,K1,K2)
112 {
113
114   Xt           = statespace
115
116   invsigxt     = 1/(sigma*sqrt(Xt))
117   gamxt        = ((2*sqrt(Xt))/sigma) # = Yt
118   gamxs        = ((2*sqrt(Xs))/sigma) # = Ys
119   p1           = 1/sqrt(2*pi*(t-s))
120   p2           = exp(-((gamxt-gamxs)^2)/(2*(t-s))-(alpha*(gamxt^2)/4)+(alpha*(gamxs^2)/4))
        *(gamxt^(-0.5+2*alpha*beta/sigma^2))*(gamxs^(0.5-2*alpha*beta/sigma^2))
121   p            = p1*p2
122   c1           = -1/(24*gamxt*gamxs*sigma^4)*(48*(alpha*beta)^2-48*alpha*beta*(sigma^2)
        +9*(sigma^4)+gamxt*(alpha^2)*(sigma^2)*gamxs*(-24*beta+(gamxt^2)*(sigma^2))+(gamxt
        ^2)*(alpha^2)*(sigma^4)*(gamxs^2)+gamxt*(alpha^2)*(sigma^4)*(gamxs^3))
123   c2           = (1/(576*gamxt^2*gamxs^2))*(9*(256*alpha^4*beta^4-512*alpha^3*beta^3*sigma
        ^2+224*alpha*beta*sigma^6-15*sigma^8)+6*gamxt*alpha^2*sigma^2*(-24*beta+gamxt^2*
        sigma^2)*(16*beta^2*alpha^2-16*beta*alpha*sigma^2+3*sigma^4)*gamxs+gamxt^2*alpha^2*
```

```
    sigma^4*(672*beta^2*alpha^2-48*beta*alpha*(2+gamxt^2*alpha)*sigma^2+(-6+gamxt^4*
    alpha^2)*sigma^4)*gamxs^2+2*gamxt*alpha^2*sigma^4*(48*beta^2*alpha^2-24*beta*alpha
    *(2+gamxt^2*alpha)*sigma^2+(9+gamxt^4*alpha^2)*sigma^4)*gamxs^3+3*gamxt^2*alpha^4*
    sigma^6*(-16*beta+gamxt^2*sigma^2)*gamxs^4+2*gamxt^3*alpha^4*sigma^8*gamxs^5+gamxt
    ^2*alpha^4*sigma^8*gamxs^6)


if (K1==0)
{
  hermitedens1 = invsigxt*p
}

if (K1==1)
{
  hermitedens1 = invsigxt*p*(1+(t-s)*c1)
}

if (K1==2)
{
  hermitedens1 = invsigxt*p*(1+(t-s)*c1 + (((t-s)^2)/2)*c2)
}


if (K2==0)
{
  hermitedens2 = invsigxt*p
}

if (K2==1)
{
  hermitedens2 = invsigxt*p*(1+(t-s)*c1)
}

if (K2==2)
{
  hermitedens2 = invsigxt*p*(1+(t-s)*c1 + (((t-s)^2)/2)*c2)
}



plot(hermitedens1~Xt, type = "l" , col = "azure4", lwd =2,ylim=c
    (0.5000000002,0.5000000004), xlim = c(2.4941,2.49418),ylab="Density",axes = F)
axis(1, xaxp=c(2.4941, 2.49418, 1), las=2)
```

```
  #axis(2, yaxp=c(0.5000000002, 0.5000000004, 1),outer = F, las=2)
  title(main="Density in [0.5000000002, 0.5000000004]")
  box()
  lines(hermitedens2~Xt, lty = 1 , col = "cornflowerblue", lwd = 2, ylim=c
      (0.5000000002,0.5000000004), xlim = c(2.4941,2.49418),ylab=NA)

}

CIR_hermite_diff(s,t,Xs,Xt,alpha,beta,sigma,1,2)
labels = c("Hermite: K=1", "Hermite: K=2")
legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) , col=c("azure4","
    cornflowerblue"),pch = 1)

CIR_hermite_coeff = function(s,t,Xs,Xt,alpha,beta,sigma,coeff)
{

  Xt          = statespace

  invsigxt    = 1/(sigma*sqrt(Xt))
  gamxt       = ((2*sqrt(Xt))/sigma) # = Yt
  gamxs       = ((2*sqrt(Xs))/sigma) # = Ys
  p1          = 1/sqrt(2*pi*(t-s))
  p2          = exp(-((gamxt-gamxs)^2)/(2*(t-s))-(alpha*(gamxt^2)/4)+(alpha*(gamxs^2)/4))
      *(gamxt^(-0.5+2*alpha*beta/sigma^2))*(gamxs^(0.5-2*alpha*beta/sigma^2))
  p           = p1*p2
  c1          = -1/(24*gamxt*gamxs*sigma^4)*(48*(alpha*beta)^2-48*alpha*beta*(sigma^2)
      +9*(sigma^4)+gamxt*(alpha^2)*(sigma^2)*gamxs*(-24*beta+(gamxt^2)*(sigma^2))+(gamxt
      ^2)*(alpha^2)*(sigma^4)*(gamxs^2)+gamxt*(alpha^2)*(sigma^4)*(gamxs^3))
  c2          = (1/(576*gamxt^2*gamxs^2))*(9*(256*alpha^4*beta^4-512*alpha^3*beta^3*sigma
      ^2+224*alpha*beta*sigma^6-15*sigma^8)+6*gamxt*alpha^2*sigma^2*(-24*beta+gamxt^2*
      sigma^2)*(16*beta^2*alpha^2-16*beta*alpha*sigma^2+3*sigma^4)*gamxs+gamxt^2*alpha^2*
      sigma^4*(672*beta^2*alpha^2-48*beta*alpha*(2+gamxt^2*alpha)*sigma^2+(-6+gamxt^4*
      alpha^2)*sigma^4)*gamxs^2+2*gamxt*alpha^2*sigma^4*(48*beta^2*alpha^2-24*beta*alpha
      *(2+gamxt^2*alpha)*sigma^2+(9+gamxt^4*alpha^2)*sigma^4)*gamxs^3+3*gamxt^2*alpha^4*
      sigma^6*(-16*beta+gamxt^2*sigma^2)*gamxs^4+2*gamxt^3*alpha^4*sigma^8*gamxs^5+gamxt
      ^2*alpha^4*sigma^8*gamxs^6)

  if (coeff == 1)
  {
  plot(c1~Xt, type = "p" , col = "blue", lwd = 2)
  }

  if (coeff == 2)
```

```
  {
    plot(c2~Xt, type = "p" , col = "skyblue", lwd = 2)
  }

}

CIR_hermite_coeff(s,t,Xs,Xt,alpha,beta,sigma,1)
CIR_hermite_coeff(s,t,Xs,Xt,alpha,beta,sigma,2)
```

Algorithm 5 Empirical and theoretical moments of the scalar CIR process.

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)dt + sigma(Xt,t)dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt

rm(list=ls(all=TRUE))

#Seed:
set.seed(7)

#Parameters
s              = 0
t              = 5
Xs             = 2.75
alpha          = 0.8
beta           = 3
sigma          = 0.25
delta_t        = 0.01    #step length
startingstate  = 0
endstate       = 5
numbsims       = 10000
timespace      = seq(s,t,delta_t)
statespace     = seq(startingstate,endstate,delta_t)


par(mfrow=c(2,2),ps=10,cex.lab=1,cex.axis=1,mar=c(3.5,3.5,3.5,2.5), mgp=c(2.5, 1, 0), las
    =1)

#m1t emperical and theoretical:
#Using Euler-Maruyama

CIR_moment1 = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
{
```

```
  mufunc = function(Xt,t)
  {
    return(alpha*(beta - Xt))
  }

  sigfunc = function(Xt,t)
  {
    return(sigma*sqrt(Xt))
  }

  momentfunc = function(Xs,s,t,delta_t,numbsims)
  {

    Xt = rep(Xs,numbsims)
    timespace = seq(s,t,delta_t)

    moment1mat = matrix(Xs,nrow=length(timespace),ncol=1)

    for(i in 1:length(timespace))
    {
      dWt = sqrt(delta_t)*rnorm(numbsims)
      Xt = Xt + mufunc(Xt, timespace[i])*delta_t + sigfunc(Xt,timespace[i])*dWt
      moment1mat[i] = mean(Xt)
    }

    plot(moment1mat~timespace,xlab='t',ylab='m1(t)',type = 'p',lwd = 1 ,col = 'grey')

    theomoment1 = Xs*exp(-alpha*timespace) + beta*(1 - exp(-alpha*timespace))
    lines(theomoment1~timespace,col="blue",lwd = 2)

  }

  m = momentfunc(Xs,s,t,delta_t,numbsims)
}

#m2t emperical and theoretical:
#Using Euler-Maruyama

CIR_moment2 = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
{
  mufunc = function(Xt,t)
  {
    return(alpha*(beta - Xt))
```

```
  }

  sigfunc = function(Xt,t)
  {
    return(sigma*sqrt(Xt))
  }

  momentfunc = function(Xs,s,t,delta_t,numbsims)
  {

    Xt = rep(Xs,numbsims)
    timespace = seq(s,t,delta_t)

    moment2mat = matrix(Xs,nrow=length(timespace),ncol=1)

    for(i in 1:length(timespace))
    {
      dWt = sqrt(delta_t)*rnorm(numbsims)
      Xt = Xt + mufunc(Xt, timespace[i])*delta_t + sigfunc(Xt,timespace[i])*dWt
      moment2mat[i] = mean(Xt^2)
    }

    plot(moment2mat~timespace,xlab='t',ylab='m2(t)',type = 'p',lwd = 1 ,col = 'grey')

    theomoment2 = (Xs^2)*exp(-2*alpha*timespace) + (beta + (sigma^2)/(2*alpha))*(beta +
        2*(Xs-beta)*exp(-alpha*timespace) + (beta - 2*Xs)*exp(-2*alpha*timespace))
    lines(theomoment2~timespace,col="blue",lwd = 2)

  }

  m = momentfunc(Xs,s,t,delta_t,numbsims)
}


#m3t emperical and theoretical:
#Using Euler-Maruyama

CIR_moment3 = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
{
  mufunc = function(Xt,t)
  {
    return(alpha*(beta - Xt))
  }
```

```
117
118   sigfunc = function(Xt,t)
119   {
120     return(sigma*sqrt(Xt))
121   }
122
123   momentfunc = function(Xs,s,t,delta_t,numbsims)
124   {
125
126     Xt = rep(Xs,numbsims)
127     timespace = seq(s,t,delta_t)
128
129     moment3mat = matrix(Xs,nrow=length(timespace),ncol=1)
130     for(i in 1:length(timespace))
131     {
132       dWt = sqrt(delta_t)*rnorm(numbsims)
133       Xt = Xt + mufunc(Xt, timespace[i])*delta_t + sigfunc(Xt,timespace[i])*dWt
134       moment3mat[i] = mean(Xt^3)
135     }
136
137     plot(moment3mat~timespace,xlab='t',ylab='m3(t)',type = 'p',lwd = 1 ,col = 'grey')
138
139
140     del = Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs)+beta*(beta + ((sigma^2)/(2*
          alpha)))+2*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
141     gamma = alpha*(Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs))+3*alpha*beta*(beta
          + ((sigma^2)/(2*alpha)))+4*alpha*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
142     kappa = 2*(alpha^2)*beta*(beta + ((sigma^2)/(2*alpha)))
143     A = kappa/(6*alpha^3)
144     C = -4*((1/(4*alpha^2))*(gamma-9*A*alpha^2)-(1/(2*alpha))*(del-3*alpha*A))
145     B = (1/(2*alpha))*(del-3*alpha*A-alpha*C)
146     D = -A-B-C
147     theomoment3 = (Xs^3)*exp(-3*alpha*timespace) + (3*alpha*beta+3*sigma^2)*(A + B*exp(-
          alpha*timespace) + C*exp(-2*alpha*timespace) + D*exp(-3*alpha*timespace))
148     lines(theomoment3~timespace,col="blue",lwd = 2)
149
150   }
151
152   m = momentfunc(Xs,s,t,delta_t,numbsims)
153 }
154
155 #mt4 emperical and theoretical:
156 #Using Euler-Maruyama
```

```
157
158 CIR_moment4 = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
159 {
160   mufunc = function(Xt,t)
161   {
162     return(alpha*(beta - Xt))
163   }
164
165   sigfunc = function(Xt,t)
166   {
167     return(sigma*sqrt(Xt))
168   }
169
170   momentfunc = function(Xs,s,t,delta_t,numbsims)
171   {
172
173     Xt = rep(Xs,numbsims)
174     timespace = seq(s,t,delta_t)
175
176     moment4mat = matrix(Xs,nrow=length(timespace),ncol=1)
177     for(i in 1:length(timespace))
178     {
179       dWt = sqrt(delta_t)*rnorm(numbsims)
180       Xt = Xt + mufunc(Xt, timespace[i])*delta_t + sigfunc(Xt,timespace[i])*dWt
181       moment4mat[i] = mean(Xt^4)
182     }
183
184     plot(moment4mat~timespace,xlab='t',ylab='m4(t)',type = 'p',lwd = 1 ,col = 'grey')
185
186
187     #theoretical Moment
188     del = Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs)+beta*(beta + ((sigma^2)/(2*
          alpha)))+2*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
189     gamma = alpha*(Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs))+3*alpha*beta*(beta
          + ((sigma^2)/(2*alpha)))+4*alpha*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
190     kappa = 2*(alpha^2)*beta*(beta + ((sigma^2)/(2*alpha)))
191     A = kappa/(6*alpha^3)
192     C = -4*((1/(4*alpha^2))*(gamma-9*A*alpha^2)-(1/(2*alpha))*(del-3*alpha*A))
193     B = (1/(2*alpha))*(del-3*alpha*A-alpha*C)
194     D = -A-B-C
195     gamma_star = Xs^3 + 3*(alpha*beta + sigma^2)*(A + B + C + D)
196     lambda_star = 3*alpha*Xs^3 + 3*(alpha*beta + sigma^2)*(6*alpha*A + 5*alpha*B + 4*
          alpha*C + 3*alpha*D)
```

```
    omega_star = 2*(alpha^2)*Xs^3 + 3*(alpha*beta + sigma^2)*(11*(alpha^2)*A + 6*(alpha
        ^2)*B + 3*(alpha^2)*C + 2*(alpha^2)*D)
    nu_star = 3*(alpha*beta + sigma^2)*(6*A*alpha^3)

    E = nu_star/(24*alpha^4)
    I = (-1/(6*alpha^3))*(((omega_star-(13*nu_star/(12*alpha)))-12*(alpha^2)*(gamma_star
        -(nu_star/(24*alpha^3))))-4*alpha*((lambda_star-(3*nu_star/(8*alpha^2)))-7*alpha
        *((gamma_star-(nu_star/(24*alpha^3))))))
    H = (1/(2*alpha^2))*(((lambda_star-(3*nu_star/(8*alpha^2)))-7*alpha*((gamma_star-(nu_
        star/(24*alpha^3))))) - 6*(alpha^2)*I)
    G = (-1/alpha)*((gamma_star-(nu_star/(24*alpha^3)))+ 2*alpha*H + 3*alpha*I)
    FF = -E - G - H - I


    theomoment4 = (Xs^4)*exp(-4*alpha*timespace) + (4*alpha*beta + 6*sigma^2)*(E + FF*exp
        (-1*alpha*timespace) + G*exp(-2*alpha*timespace) + H*exp(-3*alpha*timespace) + I*
        exp(-4*alpha*timespace))
    lines(theomoment4~timespace,col="blue",lwd = 2)

  }

  m = momentfunc(Xs,s,t,delta_t,numbsims)
}

#Plots


M1_plot = CIR_moment1(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
labels = c("Theoretical", "Emperical")
legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("blue","grey"),
     bty = 'n', inset = -0.025)

M2_plot = CIR_moment2(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
labels = c("Theoretical", "Emperical")
legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("blue","grey"),
     bty = 'n', inset = -0.025)

M3_plot = CIR_moment3(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
labels = c("Theoretical", "Emperical")
legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("blue","grey"),
     bty = 'n', inset = -0.025)

M4_plot = CIR_moment4(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
```

```
231 labels = c("Theoretical", "Emperical")
232 legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("blue","grey"),
        bty = 'n', inset = -0.025)
```

Algorithm 6 Empirical and theoretical cumulants of the scalar CIR process.

```
1  #CIR Diffusion Process Analysis
2  #General: dXt = mu(Xt,t)dt + sigma(Xt,t)*dWt
3  #dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt
4
5  rm(list=ls(all=TRUE))
6
7  #Seed:
8  set.seed(7)
9
10 #Parameters
11 s              = 0
12 t              = 5
13 Xs             = 2.75
14 alpha          = 0.8
15 beta           = 3
16 sigma          = 0.25
17 delta_t        = 0.01    #step length
18 startingstate  = 0
19 endstate       = 5
20 numbsims       = 10000
21 timespace      = seq(s,t,delta_t)
22 statespace     = seq(startingstate,endstate,delta_t)
23
24
25 par(mfrow=c(2,2),ps=10,cex.lab=1,cex.axis=1,mar=c(3.5,3.5,3.5,2.5), mgp=c(2.8, 1, 0), las
       =1)
26
27 #k1t emperical and theoretical:
28 CIR_cumulant1 = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
29 {
30   mufunc = function(Xt,t)
31   {
32     return(alpha*(beta - Xt))
33   }
34
35   sigfunc = function(Xt,t)
36   {
```

```
    return(sigma*sqrt(Xt))
  }

  cumulantfunc = function(Xs,s,t,delta_t,numbsims)
  {

    Xt = rep(Xs,numbsims)
    timespace = seq(s,t,delta_t)

    cumulant1mat = matrix(Xs,nrow=length(timespace),ncol=1)
    for(i in 1:length(timespace))
    {
      dWt = sqrt(delta_t)*rnorm(numbsims)
      Xt = Xt + mufunc(Xt, timespace[i])*delta_t + sigfunc(Xt,timespace[i])*dWt
      cumulant1mat[i] = mean(Xt)
    }

    plot(cumulant1mat~timespace,xlab='t',ylab='k1(t)',type = 'p',lwd = 1 ,col = 'grey')

    #theoretical Moment
    theomoment1 = Xs*exp(-alpha*timespace) + beta*(1 - exp(-alpha*timespace))

    #theoretical Cumulant
    theocumulant1 = theomoment1
    lines(theocumulant1~timespace,col="blue",lwd = 2)

  }

  c = cumulantfunc(Xs,s,t,delta_t,numbsims)
}


#k2t emperical and theoretical:
CIR_cumulant2 = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
{
  mufunc = function(Xt,t)
  {
    return(alpha*(beta - Xt))
  }

  sigfunc = function(Xt,t)
  {
    return(sigma*sqrt(Xt))
```

```
  }

  cumulantfunc = function(Xs,s,t,delta_t,numbsims)
  {

    Xt = rep(Xs,numbsims)
    timespace = seq(s,t,delta_t)

    cumulant2mat =  matrix(Xs,nrow=length(timespace),ncol=1)
    for(i in 1:length(timespace))
    {
      dWt = sqrt(delta_t)*rnorm(numbsims)
      Xt = Xt + mufunc(Xt, timespace[i])*delta_t + sigfunc(Xt,timespace[i])*dWt
      cumulant2mat[i] = mean(Xt^2) - (mean(Xt))^2
    }

    plot(cumulant2mat~timespace,xlab='t',ylab='k2(t)',type = 'p',lwd = 1 ,col = 'grey')

    #theoretical Moment
    theomoment1 = Xs*exp(-alpha*timespace) + beta*(1 - exp(-alpha*timespace))
    theomoment2 = (Xs^2)*exp(-2*alpha*timespace) + (beta + (sigma^2)/(2*alpha))*(beta +
        2*(Xs-beta)*exp(-alpha*timespace) + (beta - 2*Xs)*exp(-2*alpha*timespace))

    #theoretical Cumulant
    theocumulant2 = theomoment2-(theomoment1)^2
    lines(theocumulant2~timespace,col="blue",lwd = 2)

  }

  c = cumulantfunc(Xs,s,t,delta_t,numbsims)
}

#k3t emperical and theoretical:
CIR_cumulant3 = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
{
  mufunc = function(Xt,t)
  {
    return(alpha*(beta - Xt))
  }

  sigfunc = function(Xt,t)
  {
    return(sigma*sqrt(Xt))
```

```
}

cumulantfunc = function(Xs,s,t,delta_t,numbsims)
{

  Xt = rep(Xs,numbsims)
  timespace = seq(s,t,delta_t)

  cumulant3mat =  matrix(Xs,nrow=length(timespace),ncol=1)
  for(i in 1:length(timespace))
  {
    dWt = sqrt(delta_t)*rnorm(numbsims)
    Xt = Xt + mufunc(Xt, timespace[i])*delta_t + sigfunc(Xt,timespace[i])*dWt
    #theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
     cumulant3mat[i] = mean(Xt^3) - 3*mean(Xt)*(mean(Xt^2)) + 2*mean(Xt)^3
  }

  plot(cumulant3mat~timespace,xlab='t',ylab='k3(t)',type = 'p',lwd = 1 ,col = 'grey')

  #theoretical Moment
  del = Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs)+beta*(beta + ((sigma^2)/(2*
      alpha)))+2*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
  gamma = alpha*(Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs))+3*alpha*beta*(beta
      + ((sigma^2)/(2*alpha)))+4*alpha*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
  kappa = 2*(alpha^2)*beta*(beta + ((sigma^2)/(2*alpha)))
  A = kappa/(6*alpha^3)
  C = -4*((1/(4*alpha^2))*(gamma-9*A*alpha^2)-(1/(2*alpha))*(del-3*alpha*A))
  B = (1/(2*alpha))*(del-3*alpha*A-alpha*C)
  D = -A-B-C

  theomoment1 = Xs*exp(-alpha*timespace) + beta*(1 - exp(-alpha*timespace))
  theomoment2 = (Xs^2)*exp(-2*alpha*timespace) + (beta + (sigma^2)/(2*alpha))*(beta +
      2*(Xs-beta)*exp(-alpha*timespace) + (beta - 2*Xs)*exp(-2*alpha*timespace))
  theomoment3 = (Xs^3)*exp(-3*alpha*timespace) + (3*alpha*beta+3*sigma^2)*(A + B*exp(-
      alpha*timespace) + C*exp(-2*alpha*timespace) + D*exp(-3*alpha*timespace))

  #theoretical Cumulant
  theocumulant1 = theomoment1
  theocumulant2 = theomoment2-(theomoment1)^2
  theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
  lines(theocumulant3~timespace,col="blue",lwd = 2)

}
```

```
161
162   c = cumulantfunc(Xs,s,t,delta_t,numbsims)
163 }
164
165 #k4t emperical and theoretical:
166 CIR_cumulant4 = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
167 {
168   mufunc = function(Xt,t)
169   {
170     return(alpha*(beta - Xt))
171   }
172
173   sigfunc = function(Xt,t)
174   {
175     return(sigma*sqrt(Xt))
176   }
177
178   cumulantfunc = function(Xs,s,t,delta_t,numbsims)
179   {
180
181     Xt = rep(Xs,numbsims)
182     timespace = seq(s,t,delta_t)
183
184     cumulant4mat =  matrix(Xs,nrow=length(timespace),ncol=1)
185     for(i in 1:length(timespace))
186     {
187       dWt = sqrt(delta_t)*rnorm(numbsims)
188       Xt = Xt + mufunc(Xt, timespace[i])*delta_t + sigfunc(Xt,timespace[i])*dWt
189       #theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
190       cumulant4mat[i] = -6*(mean(Xt)^4) + 12*(mean(Xt)^2)*(mean(Xt^2)) - 3*(mean(Xt^2)^2)
              - 4*mean(Xt)*mean(Xt^3) + mean(Xt^4)
191     }
192
193     plot(cumulant4mat~timespace,xlab='t',ylab='k4(t)',type = 'p',lwd = 1 ,col = 'grey')
194
195     #theoretical Moment
196     del = Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs)+beta*(beta + ((sigma^2)/(2*
          alpha)))+2*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
197     gamma = alpha*(Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs))+3*alpha*beta*(beta
          + ((sigma^2)/(2*alpha)))+4*alpha*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
198     kappa = 2*(alpha^2)*beta*(beta + ((sigma^2)/(2*alpha)))
199     A = kappa/(6*alpha^3)
200     C = -4*((1/(4*alpha^2))*(gamma-9*A*alpha^2)-(1/(2*alpha))*(del-3*alpha*A))
```

```
    B = (1/(2*alpha))*(del-3*alpha*A-alpha*C)
    D = -A-B-C
    gamma_star = Xs^3 + 3*(alpha*beta + sigma^2)*(A + B + C + D)
    lambda_star = 3*alpha*Xs^3 + 3*(alpha*beta + sigma^2)*(6*alpha*A + 5*alpha*B + 4*
        alpha*C + 3*alpha*D)
    omega_star = 2*(alpha^2)*Xs^3 + 3*(alpha*beta + sigma^2)*(11*(alpha^2)*A + 6*(alpha
        ^2)*B + 3*(alpha^2)*C + 2*(alpha^2)*D)
    nu_star = 3*(alpha*beta + sigma^2)*(6*A*alpha^3)

    E = nu_star/(24*alpha^4)
    I = (-1/(6*alpha^3))*(((omega_star-(13*nu_star/(12*alpha)))-12*(alpha^2)*(gamma_star
        -(nu_star/(24*alpha^3))))-4*alpha*((lambda_star-(3*nu_star/(8*alpha^2)))-7*alpha
        *((gamma_star-(nu_star/(24*alpha^3))))))
    H = (1/(2*alpha^2))*(((lambda_star-(3*nu_star/(8*alpha^2)))-7*alpha*((gamma_star-(nu_
        star/(24*alpha^3))))) - 6*(alpha^2)*I)
    G = (-1/alpha)*((gamma_star-(nu_star/(24*alpha^3)))+ 2*alpha*H + 3*alpha*I)
    FF = -E - G - H - I

    theomoment1 = Xs*exp(-alpha*timespace) + beta*(1 - exp(-alpha*timespace))
    theomoment2 = (Xs^2)*exp(-2*alpha*timespace) + (beta + (sigma^2)/(2*alpha))*(beta +
        2*(Xs-beta)*exp(-alpha*timespace) + (beta - 2*Xs)*exp(-2*alpha*timespace))
    theomoment3 = (Xs^3)*exp(-3*alpha*timespace) + (3*alpha*beta+3*sigma^2)*(A + B*exp(-
        alpha*timespace) + C*exp(-2*alpha*timespace) + D*exp(-3*alpha*timespace))
    theomoment4 = (Xs^4)*exp(-4*alpha*timespace) + (4*alpha*beta + 6*sigma^2)*(E + FF*exp
        (-1*alpha*timespace) + G*exp(-2*alpha*timespace) + H*exp(-3*alpha*timespace) + I*
        exp(-4*alpha*timespace))

    #theoretical Cumulant
    theocumulant1 = theomoment1
    theocumulant2 = theomoment2-(theomoment1)^2
    theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
    theocumulant4 = -6*(theomoment1^4) + 12*(theomoment1^2)*(theomoment2) - 3*(
        theomoment2^2) - 4*theomoment1*theomoment3 + theomoment4
    lines(theocumulant4~timespace,col="blue",lwd = 2)

  }

  c = cumulantfunc(Xs,s,t,delta_t,numbsims)
}

#Plots


```

```
C1_plot = CIR_cumulant1(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
labels = c("Theoretical", "Emperical")
legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("blue","grey"),
     bty = 'n', inset = -0.025)

C2_plot = CIR_cumulant2(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
labels = c("Theoretical", "Emperical")
legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("blue","grey"),
     bty = 'n', inset = -0.025)

C3_plot = CIR_cumulant3(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
labels = c("Theoretical", "Emperical")
legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("blue","grey"),
     bty = 'n', inset = -0.025)

C4_plot = CIR_cumulant4(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
labels = c("Theoretical", "Emperical")
legend("topleft", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("blue","grey"), bty
     = 'n',inset=-0.14)
```

Algorithm 7 True transition density and the Moment-truncated saddlepoint transition density approximation of the transition density of the scalar CIR process.

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)dt + sigma(Xt,t)dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt

rm(list=ls(all=TRUE))

#Seed:
set.seed(7)

#Parameters
s             = 0
t             = 5
Xs            = 2.75
alpha         = 0.8
beta          = 3
sigma         = 0.25
delta_t       = 0.01   #step length
startingstate = 0
endstate      = 5
```

```
numbsims      = 10000
timespace     = seq(s,t,delta_t)
statespace    = seq(startingstate,endstate,delta_t)

par(ps=10,cex.lab=1,cex.axis=1,mar=c(3.5,3.5,3.5,2.5), mgp=c(2.5, 1, 0), las=1)

#Theoretical density 1: PLotted from the density given in Sahalia-paper

CIR_theoretical1 = function(s,t,Xs,Xt,alpha,beta,sigma)
{
  c               = (2*alpha)/((sigma^2)*(1-exp(-alpha*(t-s))))
  u               = c*Xs*exp(-alpha*(t-s))
  v               = c*Xt
  q               = 2*alpha*beta/(sigma^2) - 1
  besselparameter = 2*(u*v)^(0.5)
  besselfunction  = besselI(besselparameter,q,expon.scaled = TRUE)
  logbessel       = log(besselI(besselparameter,q,expon.scaled = TRUE))+besselparameter
  logfXt          = log(c) - (u+v) + (q/2)*log(v/u) + logbessel
  return(exp(logfXt))
}

Xt = statespace
plot_theoretical1 = CIR_theoretical1(s,t,Xs,Xt,alpha,beta,sigma)

plot(plot_theoretical1~Xt,col = "black",lwd = 3, type="l", ylab = "Density", xlab='Xt')

#saddlepoint Approx

#theoretical Moment
del = Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs)+beta*(beta + ((sigma^2)/(2*alpha)
    ))+2*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
gamma = alpha*(Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs))+3*alpha*beta*(beta + ((
    sigma^2)/(2*alpha)))+4*alpha*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
kappa = 2*(alpha^2)*beta*(beta + ((sigma^2)/(2*alpha)))
A = kappa/(6*alpha^3)
C = -4*((1/(4*alpha^2))*(gamma-9*A*alpha^2)-(1/(2*alpha))*(del-3*alpha*A))
B = (1/(2*alpha))*(del-3*alpha*A-alpha*C)
D = -A-B-C

gamma_star = Xs^3 + 3*(alpha*beta + sigma^2)*(A + B + C + D)
lambda_star = 3*alpha*Xs^3 + 3*(alpha*beta + sigma^2)*(6*alpha*A + 5*alpha*B + 4*alpha*C
    + 3*alpha*D)
omega_star = 2*(alpha^2)*Xs^3 + 3*(alpha*beta + sigma^2)*(11*(alpha^2)*A + 6*(alpha^2)*B
```

```
    + 3*(alpha^2)*C + 2*(alpha^2)*D)
nu_star = 3*(alpha*beta + sigma^2)*(6*A*alpha^3)

E = nu_star/(24*alpha^4)
I = (-1/(6*alpha^3))*(((omega_star-(13*nu_star/(12*alpha)))-12*(alpha^2)*(gamma_star-(nu_
    star/(24*alpha^3))))-4*alpha*((lambda_star-(3*nu_star/(8*alpha^2)))-7*alpha*((gamma_
    star-(nu_star/(24*alpha^3))))))
H = (1/(2*alpha^2))*(((lambda_star-(3*nu_star/(8*alpha^2)))-7*alpha*((gamma_star-(nu_star
    /(24*alpha^3))))) - 6*(alpha^2)*I)
G = (-1/alpha)*((gamma_star-(nu_star/(24*alpha^3)))+ 2*alpha*H + 3*alpha*I)
FF = -E - G - H - I

theomoment1 = Xs*exp(-alpha*(t-s)) + beta*(1 - exp(-alpha*(t-s)))
theomoment2 = (Xs^2)*exp(-2*alpha*(t-s)) + (beta + (sigma^2)/(2*alpha))*(beta + 2*(Xs-
    beta)*exp(-alpha*(t-s)) + (beta - 2*Xs)*exp(-2*alpha*(t-s)))
theomoment3 = (Xs^3)*exp(-3*alpha*(t-s)) + (3*alpha*beta+3*sigma^2)*(A + B*exp(-alpha*(t-
    s)) + C*exp(-2*alpha*(t-s)) + D*exp(-3*alpha*(t-s)))
theomoment4 = (Xs^4)*exp(-4*alpha*(t-s)) + (4*alpha*beta + 6*sigma^2)*(E + FF*exp(-1*
    alpha*(t-s)) + G*exp(-2*alpha*(t-s)) + H*exp(-3*alpha*(t-s)) + I*exp(-4*alpha*(t-s)))

#theoretical Cumulant
theocumulant1 = theomoment1
theocumulant2 = theomoment2-(theomoment1)^2
theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
theocumulant4 = -6*(theomoment1^4) + 12*(theomoment1^2)*(theomoment2) - 3*(theomoment2^2)
     - 4*theomoment1*theomoment3 + theomoment4

X = statespace

s = (1/theocumulant3)*(sqrt(theocumulant2^2 - 2*theocumulant3*(theocumulant1-X)) -
    theocumulant2)
Ksapprox = theocumulant1*s + theocumulant2*((1/2)*s^2) + theocumulant3*((1/6)*s^3) +
    theocumulant4*((1/24)*s^4)
Ks2approx = theocumulant2 + theocumulant3*s + 0.5*theocumulant4*s^2

saddle_pt_approx = exp(Ksapprox-s*X)*sqrt(1/(2*pi*Ks2approx))
#print(saddle_pt_approx)

lines(saddle_pt_approx~statespace,lty = 3,col = "dodgerblue1", lwd = 3)
labels = c("Theoretical", "saddlepoint approx")
legend("topleft", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("black","
    dodgerblue1"), bty = 'n')
```

Algorithm 8 Trajectory of the 5 year observed S&P500VIX dataset.

```
rm(list = ls(all = TRUE))
set.seed(7)

library(readxl)
SP500VIX <- read_excel("D:/ResearchUSB/Report_Draft_Final/Code/R_code/final_draft_code/
    original_model/SP500VIX.xlsx",
                       col_types = c("date", "numeric"))


X = as.matrix(na.omit(SP500VIX$VIX),nrow(na.omit(SP500VIX$VIX)),1) #1Jan2012-30Dec2016


#Simulating the trajectory of saddlepoint mles and theoretical mles

set.seed(7)

par(ps=9,cex.lab=1,cex.axis=1,mar=c(3.5,3.5,3.5,2.5), mgp=c(2.5, 1.5, 0), las=1)

plot(SP500VIX$VIX,SP500VIX$date,ylab = "VIX",xlab="Days",type = "l", col="black",ylim = c
    (0,40))
legend("bottomright", title = NA, "Observed VIX",lty = c(1), lwd = c(2) ,col="black", bty
     = 'n',horiz=F)
```

Algorithm 9 Theoretical maximum likelihood estimation of the scalar CIR process's parameters.

```
rm(list = ls(all = TRUE))

set.seed(7)

library(readxl)
SP500VIX <- read_excel("D:/ResearchUSB/Report_Draft_Final/Code/R_code/final_draft_code/
    original_model/SP500VIX.xlsx",
                       col_types = c("date", "numeric"))

Xt = as.matrix(na.omit(SP500VIX$VIX),nrow(na.omit(SP500VIX$VIX)),1) #1Jan2012-30Dec2016

n               = nrow(Xt)
s               = 1/250
t               = 5

```

```
timespace = seq(s,n/250,s)


likelihood = function(theta)
{
  alpha = theta[1]
  beta  = theta[2]
  sigma = theta[3]
  Xt    = Xt[-n]
  Xs    = Xt[-1]
  dt    = diff(timespace)[1] #1/250
  a = ((sigma^2)/alpha)*(exp(-alpha*(dt))-exp(-2*alpha*(dt)))
  b = (1-exp(-alpha*(dt)))
  mean = Xs*exp(-alpha*(dt)) + beta*b
  variance = Xs*a + beta*((sigma^2)/(2*alpha))*(b^2)
  theta = variance/mean
  kappa = (mean^2)/variance
  f = dgamma(Xt,scale=theta,shape=kappa,log=TRUE)
  minloglike    = -sum(f)
  return(minloglike)
}

mle_estimates = nlm(likelihood, c(0.5,3,0.2))

print(mle_estimates$estimate)
```

Algorithm 10 Moment-truncated saddlepoint transition density approximation maximum likelihood estimation of the scalar CIR process's parameters.

```

rm(list = ls(all = TRUE))

set.seed(7)

library(readxl)
SP500VIX <- read_excel("D:/ResearchUSB/Report_Draft_Final/Code/R_code/final_draft_code/
    original_model/SP500VIX.xlsx",
                       col_types = c("date", "numeric"))

#saddlepoint mle
X = as.matrix(na.omit(SP500VIX$VIX),nrow(na.omit(SP500VIX$VIX)),1) #1Jan2012-30Dec2016


```

```
#plot(SP500VIX, type = "l", ylab="VIX")

s                = 1/250
Xs               = mean(X)
Tt               = 5
numberparsim     = 1000
numberpar        = 3
sims             = 1
alpha_min        = 20
alpha_max        = 24
beta_min         = 13
beta_max         = 17
sigma_min        = 3
sigma_max        = 7
simulations      = 10
dt               = s

max_theta_matrix = matrix(0,simulations,numberpar)


for (m in 1:simulations)
{

  main_function = function(numberparsim,numberpar,s,t,Xs,alpha_min,alpha_max,beta_min,
      beta_max,sigma_min,sigma_max)
  {
    m = numberpar + 1

    theta_matrix = matrix(0,numberparsim,m)
    for (i in 1:numberparsim)
    {
      theta_matrix[i,1] = runif(1,alpha_min,alpha_max)
      theta_matrix[i,2] = runif(1,beta_min,beta_max)
      theta_matrix[i,3] = runif(1,sigma_min,sigma_max)

      alpha = theta_matrix[i,1]
      beta  = theta_matrix[i,2]
      sigma = theta_matrix[i,3]

      #theoretical Moment
      del = Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs)+beta*(beta + ((sigma^2)/(2*
          alpha)))+2*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
      gamma = alpha*(Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs))+3*alpha*beta*(
```

```
        beta + ((sigma^2)/(2*alpha)))+4*alpha*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
    kappa = 2*(alpha^2)*beta*(beta + ((sigma^2)/(2*alpha)))
    A = kappa/(6*alpha^3)
    C = -4*((1/(4*alpha^2))*(gamma-9*A*alpha^2)-(1/(2*alpha))*(del-3*alpha*A))
    B = (1/(2*alpha))*(del-3*alpha*A-alpha*C)
    D = -A-B-C

    gamma_star = Xs^3 + 3*(alpha*beta + sigma^2)*(A + B + C + D)
    lambda_star = 3*alpha*Xs^3 + 3*(alpha*beta + sigma^2)*(6*alpha*A + 5*alpha*B + 4*
        alpha*C + 3*alpha*D)
    omega_star = 2*(alpha^2)*Xs^3 + 3*(alpha*beta + sigma^2)*(11*(alpha^2)*A + 6*(alpha
        ^2)*B + 3*(alpha^2)*C + 2*(alpha^2)*D)
    nu_star = 3*(alpha*beta + sigma^2)*(6*A*alpha^3)

    E = nu_star/(24*alpha^4)
    I = (-1/(6*alpha^3))*(((omega_star-(13*nu_star/(12*alpha)))-12*(alpha^2)*(gamma_
        star-(nu_star/(24*alpha^3))))-4*alpha*((lambda_star-(3*nu_star/(8*alpha^2)))-7*
        alpha*((gamma_star-(nu_star/(24*alpha^3))))))
    H = (1/(2*alpha^2))*(((lambda_star-(3*nu_star/(8*alpha^2)))-7*alpha*((gamma_star-(
        nu_star/(24*alpha^3))))) - 6*(alpha^2)*I)
    G = (-1/alpha)*((gamma_star-(nu_star/(24*alpha^3)))+ 2*alpha*H + 3*alpha*I)
    FF = -E - G - H - I

    theomoment1 = Xs*exp(-alpha*dt) + beta*(1 - exp(-alpha*dt))
    theomoment2 = (Xs^2)*exp(-2*alpha*dt) + (beta + (sigma^2)/(2*alpha))*(beta + 2*(Xs-
        beta)*exp(-alpha*dt) + (beta - 2*Xs)*exp(-2*alpha*dt))
    theomoment3 = (Xs^3)*exp(-3*alpha*dt) + (3*alpha*beta+3*sigma^2)*(A + B*exp(-alpha*
        dt) + C*exp(-2*alpha*dt) + D*exp(-3*alpha*dt))
    theomoment4 = (Xs^4)*exp(-4*alpha*dt) + (4*alpha*beta + 6*sigma^2)*(E + FF*exp(-1*
        alpha*dt) + G*exp(-2*alpha*dt) + H*exp(-3*alpha*dt) + I*exp(-4*alpha*dt))


    #theoretical Cumulant
    theocumulant1 = theomoment1
    theocumulant2 = theomoment2-(theomoment1)^2
    theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
    theocumulant4 = -6*(theomoment1^4) + 12*(theomoment1^2)*(theomoment2) - 3*(
        theomoment2^2) - 4*theomoment1*theomoment3 + theomoment4

    n = nrow(X)
    cumul_sum_vec = matrix(0,n,1)

    r = (1/theocumulant3)*(sqrt(theocumulant2^2 - 2*theocumulant3*(theocumulant1-X[1]))
```

```
          - theocumulant2)
88      Krapprox = theocumulant1*r + theocumulant2*((1/2)*r^2) + theocumulant3*((1/6)*r^3)
            + theocumulant4*((1/24)*r^4)
89      Kr2approx = theocumulant2 + theocumulant3*r + 0.5*theocumulant4*r^2
90      saddle_pt_approx = exp(Krapprox-r*X[1])*sqrt(1/(2*pi*Kr2approx))
91      sum1 = log(saddle_pt_approx)
92
93      cumul_sum_vec[1] = sum1
94
95      for (l in 2:n)
96      {
97        r = (1/theocumulant3)*(sqrt(theocumulant2^2 - 2*theocumulant3*(theocumulant1-X[l
              ])) - theocumulant2)
98        Krapprox = theocumulant1*r + theocumulant2*((1/2)*r^2) + theocumulant3*((1/6)*r
              ^3) + theocumulant4*((1/24)*r^4)
99        Kr2approx = theocumulant2 + theocumulant3*r + 0.5*theocumulant4*r^2
100       saddle_pt_approx = exp(Krapprox-r*X[l])*sqrt(1/(2*pi*Kr2approx))
101       log_saddle = log(saddle_pt_approx)
102       sum_log_saddle = cumul_sum_vec[l-1] + log_saddle
103       cumul_sum_vec[l] = sum_log_saddle
104     }
105
106     theta_matrix[i,4] = cumul_sum_vec[n]
107
108   }
109
110   return(theta_matrix)
111
112 }
113
114
115 call_main_function = main_function(numberparsim,numberpar,s,t,Xs,alpha_min,alpha_max,
        beta_min,beta_max,sigma_min,sigma_max)
116 #print(call_main_function
117
118
119
120 for (i in 1:numberparsim)
121 {
122   if ((call_main_function[i,4]=="NaN"))
123   {call_main_function[i,4] = -10000000000000000000000000}
124 }
125
```

```
  maximum = max(call_main_function[,4])

  for (j in 1:numberparsim)
  {
    if (call_main_function[j,4]==maximum)
    {
      max_theta = call_main_function[j,1:3]
      max_alpha = max_theta[1]
      max_beta  = max_theta[2]
      max_sigma = max_theta[3]
    }
  }

  max_theta_matrix[m,1] = max_alpha
  max_theta_matrix[m,2] = max_beta
  max_theta_matrix[m,3] = max_sigma

}

print(max_theta_matrix)

one = matrix(1,1,simulations)
theta_mles = (1/simulations)*one%*%(max_theta_matrix)
names(theta_mles) = c("alpha","beta","sigma")

theta_mles_fin  = as.data.frame(rbind(names(theta_mles),theta_mles),1,3)

print(theta_mles_fin)

saddle_mlesa = as.matrix(theta_mles_fin,3,1)
```

Algorithm 11 Simulated trajectories using the theoretical and saddlepoint transition density approximation maximum likelihood estimates of the scalar CIR process.

```
rm(list = ls(all = TRUE))

library(readxl)
SP500VIX <- read_excel("D:/ResearchUSB/Report_Draft_Final/Code/R_code/final_draft_code/
    original_model/SP500VIX.xlsx",
                       col_types = c("date", "numeric"))


X = as.matrix(na.omit(SP500VIX$VIX),nrow(na.omit(SP500VIX$VIX)),1) #1Jan2012-30Dec2016
```

```
9
10
11  #Simulating the trajectory of saddlepoint mles and theoretical mles
12
13  set.seed(7)
14
15  par(mfrow=c(2,1),ps=9,cex.lab=1,cex.axis=1,mar=c(2.5,2.5,2.5,2.5), mgp=c(1.5, 0.75, 0),
        las=1)
16
17
18  #parameters:
19  s = 1/250
20  Tt = 5
21  dt = s
22  Xs = mean(X)
23
24
25  CIR_mle_theoretical_trajectory = function(s,Tt,Xs,alpha,beta,sigma,dt)
26  {
27
28    #Theoretical mle's
29    alpha    = 22.425318
30    beta     = 15.740536
31    sigma    = 5.127504
32
33    timeseq        = (seq(s,Tt,dt))
34    datamatrix     = matrix(0,nrow = length(timeseq), ncol = 1)
35    Z1             = rnorm(1,mean = 0, sd = sqrt(dt))
36    Xt             = Xs + alpha*(beta-Xs)*dt + sigma*sqrt(Xs)*Z1
37    datamatrix[1] = Xt
38
39    for(i in 2:length(timeseq))
40    {
41      dWt            = rnorm(1,mean = 0, sd = sqrt(dt))
42      Xtplus1        = Xt  + alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt
43      Xt             = Xtplus1
44      datamatrix[i] = Xtplus1
45    }
46
47    X = datamatrix
48
49    plot(X~seq(s,Tt,dt),type = "l", col="mediumorchid", xlab="Years", ylab = "Simulated VIX
        - theoretical mle's")
```

```
}

trajectory_plot = CIR_mle_theoretical_trajectory(s,Tt,Xs,alpha,beta,sigma,dt)

CIR_mle_Saddle_trajectory = function(s,Tt,Xs,alpha,beta,sigma,dt)
{

  #saddlepoint mles
  alpha          = 22.2687135545537
  beta           = 15.5513544885442
  sigma          = 5.2684889501892


  timeseq        = (seq(s,Tt,dt))
  datamatrix     = matrix(0,nrow = length(timeseq), ncol = 1)
  Z1             = rnorm(1,mean = 0, sd = sqrt(dt))
  Xt             = Xs + alpha*(beta-Xs)*dt + sigma*sqrt(Xs)*Z1
  datamatrix[1] = Xt

  for(i in 2:length(timeseq))
  {
    dWt            = rnorm(1,mean = 0, sd = sqrt(dt))
    Xtplus1        = Xt  + alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt
    Xt             = Xtplus1
    datamatrix[i] = Xtplus1
  }

  X = datamatrix

  plot(X~seq(s,Tt,dt),type="l",col="royalblue",xlab="Years", ylab = "Simulated VIX -
      saddlepoint approx. mle's")
}

trajectory_plot = CIR_mle_Saddle_trajectory(s,Tt,Xs,alpha,beta,sigma,dt)
```

Algorithm 12 Simulated trajectories for the mixed-effects CIR process

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)*dt + sigma(Xt,t)*dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt

rm(list=ls(all=TRUE))

```

```
#Seed:
set.seed(7)

#Parameters
s              = 0
t              = 5
Xs             = 2.75
alpha          = 0.8
beta           = 3
sigma          = 0.25
delta_t        = 0.01    #step length
startingstate  = 0
endstate       = 5
timespace      = seq(s,t,delta_t)
statespace     = seq(startingstate,endstate,delta_t)


#Simulating the trajectory
func = function(randsims, a,b)
{
  sigma_vec          = rnorm(randsims,a,b)
  trajectory_matrix  = matrix(0,nrow = length(timespace), ncol = randsims)
  one                = matrix(1,nrow = randsims, 1)

  for (r in 1:randsims)
  {
    sigma      = sigma_vec[r]

    CIR_trajectory = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
    {

      timespace                = seq(s,t,delta_t)
      datamatrix               = matrix(0,nrow = length(timespace), ncol = 1)
      Z1                       = rnorm(1,mean = 0, sd = sqrt(delta_t))
      Xt                       = Xs + alpha*(beta-Xs)*delta_t + sigma*sqrt(Xs)*Z1
      datamatrix[1]            = Xt
      trajectory_matrix[,1]    = datamatrix[1]

      for(i in 2:length(timespace))
      {
        dWt            = rnorm(1,mean = 0, sd = sqrt(delta_t))
        Xtplus1        = Xt  + alpha*(beta-Xt)*delta_t + sigma*sqrt(Xt)*dWt
        Xt             = Xtplus1
```

```
50          datamatrix[i] = Xtplus1
51        }
52        return(datamatrix)
53      }
54
55      trajectory_matrix[,r] = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,
            endstate)
56
57    }
58
59    #print(trajectory_matrix)
60
61    plot(trajectory_matrix[,1]~timespace,type ='l', col = rainbow(1, start = runif
          (1,0.55,0.85), end = runif(1,0.65,0.75),alpha = 0.15), xlab="t",ylab = "Xt",ylim=c
          (0,5))
62
63
64    color = rainbow(randsims-1, start = .5, end = .7)
65      for (plotnumb in 2:randsims)
66        {
67          lines(trajectory_matrix[,plotnumb]~timespace,col = rainbow(1, start = runif
                (1,0.55,0.85), end = runif(1,0.65,0.85),alpha = 0.15), type = 'l')
68        }
69
70    mean_trajectory = (1/randsims)*trajectory_matrix %*% one
71    #print(mean_trajectory)
72
73    lines(mean_trajectory~timespace, col = "midnightblue",  type = 'l', lwd = 3)
74
75  }
76
77  par(mfrow=c(2,2),ps=10,cex.lab=1,cex.axis=1,mar=c(3.5,3.5,3.5,2.5), mgp=c(2.5, 1, 0), las
        =1)
78
79  a               = 0.25
80  b               = 0.15
81
82  func(10,a,b)
83  labels = c("10 Simulations", "Average")
84  legend("bottomright", title = NA, labels,lty = c(1,1), lwd = c(5,3) ,col=c(rainbow(1,
        start = 0.7, end = 0.75 ,alpha = 0.25),"midnightblue"), bty = 'n',horiz=T)
85
86  func(25,a,b)
```

```
labels = c("25 Simulations", "Average")
legend("bottomright", title = NA, labels,lty = c(1,1), lwd = c(5,3) ,col=c(rainbow(1,
    start = 0.7, end = 0.75 ,alpha = 0.35),"midnightblue"), bty = 'n',horiz=T)

func(100,a,b)
labels = c("100 Simulations", "Average")
legend("bottomright", title = NA, labels,lty = c(1,1), lwd = c(5,3) ,col=c(rainbow(1,
    start = 0.7, end = 0.75 ,alpha = 0.5),"midnightblue"), bty = 'n',horiz=T)

func(250,a,b)
labels = c("250 Simulations", "Average")
legend("bottomright", title = NA, labels,lty = c(1,1), lwd = c(5,3) ,col=c(rainbow(1,
    start = 0.7, end = 0.75 ,alpha = 0.6),"midnightblue"), bty = 'n',horiz = T)
```

Algorithm 13 Overlayed perspective plots for the mixed-effects CIR diffusion process.

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)*dt + sigma(Xt,t)*dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt

rm(list=ls(all=TRUE))


set.seed(7)

#Parameters
s               = 0
t               = 5
Xs              = 2.75
alpha           = 0.8
beta            = 3
delta_t         = 0.1    #step length
startingstate   = 0
endstate        = 5
timespace       = seq(s,t,delta_t)
statespace      = seq(startingstate,endstate,delta_t)
a               = 0.25
b               = 0.15
randsimsactual = 100


simplotfunc = function(a,b,randsims,randsimsactual, theta_rv,phi_rv,r_rv)
{
```

```
28
29  #Perspective Plot
30  funcpersps = function(randsims,randsimsactual,a,b,theta_rv,phi_rv,r_rv)
31  {
32    set.seed(7)
33
34    CIR_perspective = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,shade
          _rv = 0.9, theta_rv , phi_rv, r_rv,seedval,transparency,make_axes)
35    {
36
37      set.seed(seedval)
38
39      timespace  = seq(s,t,delta_t)
40      statespace = seq(startingstate,endstate,delta_t)
41
42      datamatrix = matrix(0,length(timespace),length(statespace))
43
44      for (t in s:length(timespace))
45      {
46        for (state in startingstate:length(statespace))
47        {
48          c                    = (2*alpha)/((sigma^2)*(1-exp(-alpha*(timespace[t]-s))))
49          u                    = c*Xs*exp(-alpha*(timespace[t]-s))
50          v                    = c*statespace[state]
51          q                    = 2*alpha*beta/(sigma^2) - 1
52          besselparameter      = 2*(u*v)^(0.5)
53          logbessel            = log(besselI(besselparameter,q,expon.scaled = TRUE))+
                besselparameter
54          logfXt_t             = log(c) - (u+v) + (q/2)*log(v/u) + logbessel
55          datamatrix[t,state]  = exp(logfXt_t)
56        }
57
58      }
59
60
61        if (make_axes==1)
62        {
63          perspplot = persp(timespace,statespace,datamatrix, col = rainbow(1, start = runif
                (1,0.85,0.95), end = runif(1,0.30,0.90),alpha=transparency), xlab="t", ylab="
                Xt",zlab="Density", border = NA, shade = 0.7 , theta = theta_rv, phi = phi_rv
                , r = r_rv, ticktype = "detailed",zlim=c(0,5.5),box=T)
64        }
65        else
```

```
66        {
67          perspplot = persp(timespace,statespace,datamatrix, col = rainbow(1, start = runif
              (1,0.55,0.65), end = runif(1,0.45,0.65), alpha=transparency), xlab=NA, ylab=
              NA,zlab=NA, border = NA, shade = 0.7, theta = theta_rv, phi = phi_rv, r = r_
              rv, box = FALSE,zlim=c(0,5.5))
68        }
69
70
71      return(perspplot)
72
73
74  }
75
76        sigma_vec            = rnorm(randsimsactual,a,b)
77        transvec             = seq(0.4,0.7,length.out = randsims)
78        plotsigmavec         = matrix(0,randsims,1)
79        plot_selection_seed = round(runif(1,1,10000))
80        set.seed(plot_selection_seed)
81        plotsigmavec = sample(sigma_vec,randsims,replace=F)
82
83        #Renew Original Seed
84        set.seed(7)
85
86        CIR_perspective(s,t,Xs,alpha,beta,plotsigmavec[1],delta_t,startingstate,endstate,
              shade_rv = 0.8 , theta_rv, phi_rv,r_rv,seedval=1,transparency=transvec[1],1)
87        for (plotnumb in 2:randsims)
88        {
89          transparency = 1-transvec[plotnumb]
90          seedval=plotnumb
91          par(new=T)
92          CIR_perspective(s,t,Xs,alpha,beta,plotsigmavec[plotnumb],delta_t,startingstate,
              endstate,shade_rv = 0.8 , theta_rv, phi_rv,r_rv,seedval,transparency,0)
93        }
94
95  }
96
97  #Final Plotting
98
99    funcpersps(randsims,randsimsactual,a,b,theta_rv,phi_rv,r_rv)
100   title(main = bquote(lambda == .(randsims)))
101
102 }
103
```

```
par(mfrow=c(4,3),ps=9,cex.lab=1,cex.axis=0.6,mar=c(2, 2, 2, 0.25), mgp=c(2, 1.5, 0), las
    =1)

simplotfunc(a,b,randsims=2,randsimsactual,45,35,35)

simplotfunc(a,b,randsims=2,randsimsactual,0,0,90)

simplotfunc(a,b,randsims=2,randsimsactual,90,0,35)

######

simplotfunc(a,b,randsims=3,randsimsactual,45,35,35)

simplotfunc(a,b,randsims=3,randsimsactual,0,0,90)

simplotfunc(a,b,randsims=3,randsimsactual,90,0,35)

######

simplotfunc(a,b,randsims=7,randsimsactual,45,35,35)

simplotfunc(a,b,randsims=7,randsimsactual,0,0,90)

simplotfunc(a,b,randsims=7,randsimsactual,90,0,35)

######


aveplotfunc = function(a,b,randsimsactual,theta_ave_rv,phi_ave_rv,r_ave_rv)
{
  CIR_perspective = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
   {
    set.seed(7)
        timespace  = seq(s,t,delta_t)
        statespace = seq(startingstate,endstate,delta_t)

        datamatrix = matrix(0,length(timespace),length(statespace))

        sumdens = as.matrix(0,length(timespace),1)

        sigma_vec  = rnorm(randsimsactual,a,b)
        for (k in 1:randsimsactual)
        {
```

```
146
147           sigma        = sigma_vec[k]
148           for (t in s:length(timespace))
149           {
150             for (state in startingstate:length(statespace))
151             {
152               c                     = (2*alpha)/((sigma^2)*(1-exp(-alpha*(timespace[t]-s))
                      ))
153               u                     = c*Xs*exp(-alpha*(timespace[t]-s))
154               v                     = c*statespace[state]
155               q                     = 2*alpha*beta/(sigma^2) - 1
156               besselparameter       = 2*(u*v)^(0.5)
157               logbessel             = log(besselI(besselparameter,q,expon.scaled = TRUE))+
                      besselparameter
158               logfXt_t              = log(c) - (u+v) + (q/2)*log(v/u) + logbessel
159               datamatrix[t,state]   = exp(logfXt_t)
160
161             }
162
163           }
164
165           sumdens = sumdens + datamatrix[,length(statespace)]
166         }
167         average = sumdens/randsimsactual
168
169
170        datamatrix[,length(statespace)]=average
171       perspplot = persp(timespace,statespace,datamatrix, col = "navyblue", xlab="t", ylab=
            "Xt",zlab="Density", border = NA, shade = 0.7 , theta = theta_ave_rv, phi = phi_
            ave_rv, r = r_ave_rv, ticktype = "detailed",zlim = c(0,5.5))
172     }
173   par(new=F)
174   perspective_plot = CIR_perspective(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,
          endstate)
175   title(main="Average: 100 simulations")
176
177 }
178
179
180 aveplotfunc(a,b,randsimsactual,45,35,35)
181
182 aveplotfunc(a,b,randsimsactual,0,0,90)
183
```

```
aveplotfunc(a,b,randsimsactual,90,0,35)
```

Algorithm 14 Euler-Maruyama Schemes for the mixed-effects CIR process.

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)dt + sigma(Xt,t)dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt
#sigma ~ N(a,b^2)


rm(list=ls(all=TRUE))

#Seed:
set.seed(7)

#Parameters

s              = 0
t              = 5
Xs             = 2.75
alpha          = 0.8
beta           = 3
delta_t        = 0.01    #step length
startingstate  = 0
endstate       = 5
numbsims       = 1000
timespace      = seq(s,t,delta_t)
statespace     = seq(startingstate,endstate,delta_t)
N              = length(timespace)
X              = rep(Xs, numbsims)

a              = 0.25 #sigma ~ N(a,b^2)
b              = 0.15
randsims       = 10


#Euler-Maruyama Scheme

CIR_EM_ME = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims,
    transparency,add,ans)
{

  mufunc = function(Xt,t)
```

```
{
  return(alpha*(beta - Xt))
}

sigfunc = function(Xt,t)
{
  return(sigma*sqrt(Xt))
}

histfunc = function(Xs,s,t,delta_t,numbsims,ans)
{

  Xt         = rep(Xs,numbsims)
  timespace = seq(s,t,delta_t)

  for(i in 1:length(timespace))
  {
    if (i==1)
    {
      Xt = Xs
    }

    dWt   = sqrt(delta_t)*rnorm(numbsims)
    Xt    = Xt + mufunc(Xt, timespace[i])*delta_t + sigfunc(Xt,timespace[i])*dWt
  }

  if (ans==1)
  {
  hist(Xt, freq = FALSE, col = rainbow(1, start = runif(1,0.55,0.85), end = runif
      (1,0.65,0.85),alpha = transparency), border = NA, breaks = 50, xlim =c(0.5,5.5),
       ylim=c(0,5), main = NA,add=F)
  return(list(Xt=Xt,time = t))
  }
  else
  {
  hist(Xt, freq = FALSE, col = rainbow(1, start = runif(1,0.55,0.85), end = runif
      (1,0.65,0.85),alpha = transparency), border = NA, breaks = 50, xlim =c(0.5,5.5),
       ylim=c(0,5), main = NA,xlab = NA,ylab=NA,add=F)
  return(list(Xt=Xt,time = t))
  }

}

```

```
78
79    plot = histfunc(Xs=Xs,s,t,delta_t,numbsims,ans)
80
81  }
82
83
84  EM_gen_func = function(a,b,randsims)
85  {
86
87  sigma_vec         = rnorm(randsims,a,b)
88  transparency_vec = sort(as.vector(runif(randsims,0.15,0.45)),decreasing = T)
89
90  CIR_EM_ME(s,t,Xs,alpha,beta,sigma=sigma_vec[1],delta_t,startingstate,endstate,numbsims,
        transparency=0.3,F,1)
91
92     for (i in 2:randsims)
93      {
94      par(new=T)
95      sigma         = sigma_vec[i]
96      transparency = transparency_vec[i]
97
98      CIR_EM_ME(s,t,Xs,alpha,beta,sigma=sigma_vec[i],delta_t,startingstate,endstate,
            numbsims,transparency=0.5,F,0)
99
100     }
101
102 }
103
104 par(mfrow=c(3,3),ps=9,cex.lab=1,cex.axis=1,mar=c(3.5,3.5,3.5,2.5), mgp=c(2.5, 1.5, 0),
        las=1)
105
106 sim_plot_vec = c(1,2,3,4,10,25)
107
108 randsims = sim_plot_vec[1]
109 EM_gen_func(a,b,randsims)
110 title(main=bquote(lambda == .(randsims)))
111 box()
112
113 for (n in 2:length(sim_plot_vec))
114 {
115   randsims = sim_plot_vec[n]
116   EM_gen_func(a,b,randsims)
117   title(main=bquote(lambda == .(randsims)))
```

```
  box()
}




 CIR_EM_ME_AVE = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims
     )
  {

    mufunc = function(Xt,t)
    {
      return(alpha*(beta - Xt))
    }

    sigfunc = function(Xt,t)
    {
      return(sigma*sqrt(Xt))
    }

    Xt                = rep(Xs,numbsims)
    timespace         = seq(s,t,delta_t)

    for(i in 1:length(timespace))
    {
      if (i==1)
      {
        Xt = Xs
      }

      dWt   = sqrt(delta_t)*rnorm(numbsims)
      Xt    = Xt + mufunc(Xt, timespace[i])*delta_t + sigfunc(Xt,timespace[i])*dWt
    }
   return(Xt)
   }




ave_EM = function(a,b,randsimsactual)
 {
   sigma_vec   = rnorm(randsimsactual,a,b)
```

```
   Yt           = matrix(0,length(CIR_EM_ME_AVE(s,t,Xs,alpha,beta,sigma=0.5,delta_t,
       startingstate,endstate,numbsims)),1)
   for (a in 1:randsimsactual)
   {
     sigma = sigma_vec[a]
     Xt = CIR_EM_ME_AVE(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
     Yt = Yt + Xt
   }

Xtave = (1/randsimsactual)*Yt

hist(Xtave, freq = FALSE, col = "navyblue",border = "orchid",main=NA,xlab = "Xt",ylab="
    Density",add=F)

 }


ave_EM(a,b,randsimsactual=10)
title(main="Average: 10 simulations")
box()
ave_EM(a,b,randsimsactual=100)
title(main="Average: 100 simulations")
box()
ave_EM(a,b,randsimsactual=1000)
title(main="Average: 1000 simulations")
box()
```

Algorithm 15 Theoretical and empirical moments of the mixed-effects CIR process.

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)dt + sigma(Xt,t)dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt
#sigma ~ N(a,b^2)

rm(list=ls(all=TRUE))

#Seed:
set.seed(7)

simulate = function(numbsims = 10000)
{
  #Parameters

```

```
15
16    s              = 0
17    t              = 5
18    Xs             = 2.75
19    alpha          = 0.8
20    beta           = 3
21    delta_t        = 0.01    #step length
22    startingstate  = 0
23    endstate       = 5
24    timespace      = seq(s,t,delta_t)
25    statespace     = seq(startingstate,endstate,delta_t)
26    N              = length(timespace)
27    X              = rep(Xs, numbsims)
28
29    a              = 0.25 #sigma ~ N(a,b^2)
30    b              = 0.15
31
32    moments = matrix(0,4,N)
33    moments[,1] = Xs^{1:4}
34    sigma_gen  = rnorm(numbsims,a,b)
35
36    for(i in 2:N)
37    {
38      dWt = sqrt(X)*rnorm(numbsims,0,sqrt(delta_t))
39      X = X + alpha*(beta-X)*delta_t + sigma_gen*dWt
40      moments[,i]  = c(mean(X), mean(X^2), mean(X^3), mean(X^4))
41    }
42    rtrn = list(X= X, moments = moments,time = timespace)
43
44    return(rtrn)
45  }
46
47  res = simulate()
48  hist(res$X, freq = F,breaks = 30, col = 'grey75', border = 'white')
49
50
51  par(mfrow=c(2,2),ps=10,cex.lab=1,cex.axis=1,mar=c(3.5,3.5,3.5,2.5), mgp=c(2.5, 1, 0), las
        =1)
52
53
54  #Parameters
55  s              = 0
56  t              = 5
```

```
Xs             = 2.75
alpha          = 0.8
beta           = 3
delta_t        = 0.01   #step length
startingstate  = 0
endstate       = 5
numbsims       = 10000
timespace      = seq(s,t,delta_t)
statespace     = seq(startingstate,endstate,delta_t)
N              = length(timespace)
X              = rep(Xs, numbsims)

a              = 0.25 #sigma ~ N(a,b^2)
b              = 0.15

#m1t emperical and theoretical mixed-effects Model:

plot(res$moments[1,]~res$time, xlab='t',ylab='m1(t)',type = 'p',lwd = 0.5, col = "
    lightblue1")

CIR_moment1 = function(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
{

  momentfunc1 = function(Xs,s,t,delta_t,numbsims)
  {

    Xt = rep(Xs,numbsims)
    timespace = seq(s,t,delta_t)

    b11 = alpha*beta
    g1  = (1 - exp(-alpha*timespace))

    y1 = (1/alpha)*g1

    theomoment1 = Xs*exp(-alpha*timespace) + b11*y1
    lines(theomoment1~timespace,col="navyblue",lwd = 3)

  }

  m = momentfunc1(Xs,s,t,delta_t,numbsims)
}

M1_plot = CIR_moment1(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
```

```
labels = c("Theoretical", "Emperical")
legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("navyblue","
    lightblue1"), bty = 'n',inset =-0.025)


#m2t emperical and theoretical mixed-effects Model:

plot(res$moments[2,]~res$time, xlab='t',ylab='m2(t)',type = 'p',lwd = 0.5, col = "
    lightblue1")


CIR_moment2 = function(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
{

  momentfunc2 = function(Xs,s,t,delta_t,numbsims)
  {

    Xt = rep(Xs,numbsims)
    timespace = seq(s,t,delta_t)

    #b12 = 2*(alpha*beta)^2 + (alpha*beta)*(b^2+a^2)
    b22 = 2*alpha*beta + (b^2+a^2)
    b11 = alpha*beta

    h1 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^1
    h2 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2

    q1 = Xs*(1/alpha)*h1
    q2 = b11*(1/(2*alpha^2))*h2

    theomoment2 = (Xs^2)*exp(-2*alpha*(timespace)) + (b22)*(q1+q2)
    lines(theomoment2~timespace,col="navyblue",lwd = 3)

  }

  m = momentfunc2(Xs,s,t,delta_t,numbsims)
}


M2_plot = CIR_moment2(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
labels = c("Theoretical", "Emperical")
legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("navyblue","
    lightblue1"), bty = 'n')
```

```
139
140
141 #m3t emperical and theoretical mixed-effects Model:
142
143 plot(res$moments[3,]~res$time, xlab='t',ylab='m3(t)',type = 'p',lwd = 0.5, col = "
        lightblue1",inset =-0.025)
144
145
146 CIR_moment3 = function(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
147 {
148
149   momentfunc3 = function(Xs,s,t,delta_t,numbsims)
150   {
151
152     Xt = rep(Xs,numbsims)
153     timespace = seq(s,t,delta_t)
154
155     b33 = 3*alpha*beta + 3*(b^2+a^2)
156     b22 = 2*alpha*beta + (b^2+a^2)
157     b11 = alpha*beta
158
159     d1 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)
160     d2 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2
161     d3 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)^3
162
163     p1  = ((Xs^2)/alpha)*d1
164     p2  = (((Xs)*(b22))/(2*alpha^2))*d2
165     p3  = ((beta)/(6*alpha^2))*b22*d3
166
167     theomoment3 = (Xs^3)*exp(-3*alpha*(timespace)) + b33*(p1+p2+p3)
168     lines(theomoment3~timespace,col="navyblue",lwd = 3)
169
170   }
171
172   m = momentfunc3(Xs,s,t,delta_t,numbsims)
173 }
174
175
176 M3_plot = CIR_moment3(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
177 labels = c("Theoretical", "Emperical")
178 legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("navyblue","
        lightblue1"), bty = 'n')
179
```

```
#m4t emperical and theoretical mixed-effects Model:

plot(res$moments[4,]~res$time, xlab='t',ylab='m4(t)',type = 'p',lwd = 0.5, col = "
    lightblue1",inset =-0.025)


CIR_moment4 = function(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
{

  momentfunc4 = function(Xs,s,t,delta_t,numbsims)
  {

    Xt = rep(Xs,numbsims)
    timespace = seq(s,t,delta_t)

    b44 = 4*alpha*beta + 6*(b^2+a^2)
    b33 = 3*alpha*beta + 3*(b^2+a^2)
    b22 = 2*alpha*beta + (b^2+a^2)
    b11 = alpha*beta

    f1 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)
    f2 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^2
    f3 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^3
    f4 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^4

    w1 = (Xs^3)*(1/alpha)*f1
    w2 = b33*(Xs^2)*(1/(2*alpha^2))*f2
    w3 = Xs*b33*b22*(1/(6*alpha^3))*f3
    w4 = b33*b22*b11*(1/(24*alpha^4))*f4

    theomoment4 = (Xs^4)*exp(-4*alpha*(timespace)) + b44*(w1 + w2 + w3 + w4)

    lines(theomoment4~timespace,col="navyblue",lwd = 3)

  }

  m = momentfunc4(Xs,s,t,delta_t,numbsims)
}


M4_plot = CIR_moment4(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
labels = c("Theoretical", "Emperical")
legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("navyblue","
```

```
    lightblue1"), bty = 'n',inset =-0.025)
```

Algorithm 16 Theoretical and empirical moments of the mixed-effects CIR process.

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)dt + sigma(Xt,t)dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt
#sigma ~ N(a,b^2)

rm(list=ls(all=TRUE))

#Seed:
set.seed(7)

simulate = function(numbsims = 10000)
{
  #Parameters


  s              = 0
  t              = 5
  Xs             = 2.75
  alpha          = 0.8
  beta           = 3
  delta_t        = 0.01    #step length
  startingstate  = 0
  endstate       = 5
  timespace      = seq(s,t,delta_t)
  statespace     = seq(startingstate,endstate,delta_t)
  N              = length(timespace)
  X              = rep(Xs, numbsims)

  a              = 0.25 #sigma ~ N(a,b^2)
  b              = 0.15

  cumulants = matrix(0,4,N)
  cumulants[,1] = Xs^{1:4}
  sigma_gen  = rnorm(numbsims,a,b)

  for(i in 2:N)
  {
    dWt = sqrt(X)*rnorm(numbsims,0,sqrt(delta_t))
    X = X + alpha*(beta-X)*delta_t + sigma_gen*dWt
```

```
    ke1 = mean(X)
    ke2 = mean(X^2) - (mean(X))^2
    ke3 = mean(X^3) - 3*mean(X)*(mean(X^2)) + 2*mean(X)^3
    ke4 =-6*(mean(X)^4) + 12*(mean(X)^2)*(mean(X^2)) - 3*(mean(X^2)^2) - 4*mean(X)*mean(X
        ^3) + mean(X^4)

    cumulants[,i]  = c(ke1, ke2, ke3, ke4)
  }
  rtrn = list(X= X, cumulants = cumulants,time = timespace)

  return(rtrn)
}

res = simulate()
hist(res$X, freq = F,breaks = 30, col = 'grey75', border = 'white')


par(mfrow=c(2,2),ps=10,cex.lab=1,cex.axis=1,mar=c(4.5,4.5,4.5,2.5), mgp=c(3.2, 1, 0), las
    =1)


#Parameters
s              = 0
t              = 5
Xs             = 2.75
alpha          = 0.8
beta           = 3
delta_t        = 0.01   #step length
startingstate  = 0
endstate       = 5
numbsims       = 10000
timespace      = seq(s,t,delta_t)
statespace     = seq(startingstate,endstate,delta_t)
N              = length(timespace)
X              = rep(Xs, numbsims)

a              = 0.25 #sigma ~ N(a,b^2)
b              = 0.15

#K1t emperical and theoretical mixed-effects Model:

plot(res$cumulants[1,]~res$time, xlab='t',ylab='K1(t)',type = 'p',lwd = 0.5, col = "
    lightblue1")
```

```
80
81  CIR_cumulant1 = function(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
82  {
83
84    cumulantfunc1 = function(Xs,s,t,delta_t,numbsims)
85    {
86
87      Xt = rep(Xs,numbsims)
88      timespace = seq(s,t,delta_t)
89
90      b11 = alpha*beta
91      g1  = (1 - exp(-alpha*timespace))
92
93      y1 = (1/alpha)*g1
94
95      theomoment1 = Xs*exp(-alpha*timespace) + b11*y1
96
97      #theoretical Cumulant
98      theocumulant1 = theomoment1
99
100     lines(theocumulant1~timespace,col="navyblue",lwd = 3)
101
102   }
103
104   c = cumulantfunc1(Xs,s,t,delta_t,numbsims)
105 }
106
107 K1_plot = CIR_cumulant1(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
108 labels = c("Theoretical", "Emperical")
109 legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("navyblue","
        lightblue1"), bty = 'n',inset =-0.025)
110
111
112 #K2t emperical and theoretical mixed-effects Model:
113
114 plot(res$cumulants[2,]~res$time, xlab='t',ylab='K2(t)',type = 'p',lwd = 0.5, col = "
        lightblue1",ylim=c(0,0.3))
115
116
117 CIR_cumulant2 = function(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
118 {
119
120   cumulantfunc2 = function(Xs,s,t,delta_t,numbsims)
```

```
  {

    Xt = rep(Xs,numbsims)
    timespace = seq(s,t,delta_t)


    b11 = alpha*beta
    g1  = (1 - exp(-alpha*timespace))

    y1 = (1/alpha)*g1

    theomoment1 = Xs*exp(-alpha*timespace) + b11*y1

    b22 = 2*alpha*beta + (b^2+a^2)
    b11 = alpha*beta

    h1 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^1
    h2 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2

    q1 = Xs*(1/alpha)*h1
    q2 = b11*(1/(2*alpha^2))*h2

    theomoment2 = (Xs^2)*exp(-2*alpha*(timespace)) + (b22)*(q1+q2)

    #theoretical Cumulants
    theocumulant2 = theomoment2-(theomoment1)^2

    lines(theocumulant2~timespace,col="navyblue",lwd = 3)

  }

  c = cumulantfunc2(Xs,s,t,delta_t,numbsims)
}


K2_plot = CIR_cumulant2(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
labels = c("Theoretical", "Emperical")
legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("navyblue","
    lightblue1"), bty = 'n',inset =-0.025)


#K3t emperical and theoretical mixed-effects Model:

```

```
plot(res$cumulants[3,]~res$time, xlab='t',ylab='K3(t)',type = 'p',lwd = 0.5, col = "
    lightblue1",ylim=c(0,0.03))


CIR_cumulant3 = function(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
{

  cumulantfunc3 = function(Xs,s,t,delta_t,numbsims)
  {

    Xt = rep(Xs,numbsims)
    timespace = seq(s,t,delta_t)

    b11 = alpha*beta
    g1  = (1 - exp(-alpha*timespace))

    y1 = (1/alpha)*g1

    theomoment1 = Xs*exp(-alpha*timespace) + b11*y1

    b22 = 2*alpha*beta + (b^2+a^2)
    b11 = alpha*beta

    h1 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^1
    h2 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2

    q1 = Xs*(1/alpha)*h1
    q2 = b11*(1/(2*alpha^2))*h2

    theomoment2 = (Xs^2)*exp(-2*alpha*(timespace)) + (b22)*(q1+q2)


    b33 = 3*alpha*beta + 3*(b^2+a^2)
    b22 = 2*alpha*beta + (b^2+a^2)
    b11 = alpha*beta

    d1 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)
    d2 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2
    d3 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)^3

    p1  = ((Xs^2)/alpha)*d1
    p2  = (((Xs)*(b22))/(2*alpha^2))*d2
    p3  = ((beta)/(6*alpha^2))*b22*d3
```

```
205
206     theomoment3 = (Xs^3)*exp(-3*alpha*(timespace)) + b33*(p1+p2+p3)
207
208     #theoretical Cumulants
209     theocumulant1 = theomoment1
210     theocumulant2 = theomoment2-(theomoment1)^2
211     theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
212
213     lines(theocumulant3~timespace,col="navyblue",lwd = 3)
214
215   }
216
217   c = cumulantfunc3(Xs,s,t,delta_t,numbsims)
218 }
219
220
221 K3_plot = CIR_cumulant3(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
222 labels = c("Theoretical", "Emperical")
223 legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("navyblue","
        lightblue1"), bty = 'n',inset =-0.025)
224
225 #K4t emperical and theoretical mixed-effects Model:
226
227 plot(res$cumulants[4,]~res$time, xlab='t',ylab='K4(t)',type = 'p',lwd = 0.5, col = "
        lightblue1",ylim = c(0,0.003))
228
229
230 CIR_cumulant4 = function(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
231 {
232
233   cumulantfunc4 = function(Xs,s,t,delta_t,numbsims)
234   {
235
236     Xt = rep(Xs,numbsims)
237     timespace = seq(s,t,delta_t)
238
239     b11 = alpha*beta
240     g1  = (1 - exp(-alpha*timespace))
241
242     y1 = (1/alpha)*g1
243
244     theomoment1 = Xs*exp(-alpha*timespace) + b11*y1
245
```

```
    b22 = 2*alpha*beta + (b^2+a^2)
    b11 = alpha*beta

    h1 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^1
    h2 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2

    q1 = Xs*(1/alpha)*h1
    q2 = b11*(1/(2*alpha^2))*h2

    theomoment2 = (Xs^2)*exp(-2*alpha*(timespace)) + (b22)*(q1+q2)



    b33 = 3*alpha*beta + 3*(b^2+a^2)
    b22 = 2*alpha*beta + (b^2+a^2)
    b11 = alpha*beta

    d1 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)
    d2 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2
    d3 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)^3

    p1  = ((Xs^2)/alpha)*d1
    p2  = (((Xs)*(b22))/(2*alpha^2))*d2
    p3  = ((beta)/(6*alpha^2))*b22*d3

    theomoment3 = (Xs^3)*exp(-3*alpha*(timespace)) + b33*(p1+p2+p3)



    b44 = 4*alpha*beta + 6*(b^2+a^2)
    b33 = 3*alpha*beta + 3*(b^2+a^2)
    b22 = 2*alpha*beta + (b^2+a^2)
    b11 = alpha*beta

    f1 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)
    f2 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^2
    f3 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^3
    f4 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^4

    w1 = (Xs^3)*(1/alpha)*f1
    w2 = b33*(Xs^2)*(1/(2*alpha^2))*f2
    w3 = Xs*b33*b22*(1/(6*alpha^3))*f3
    w4 = b33*b22*b11*(1/(24*alpha^4))*f4

    theomoment4 = (Xs^4)*exp(-4*alpha*(timespace)) + b44*(w1 + w2 + w3 + w4)
```

```
289
290     #theoretical Cumulants
291     theocumulant1 = theomoment1
292     theocumulant2 = theomoment2-(theomoment1)^2
293     theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
294     theocumulant4 = -6*(theomoment1^4) + 12*(theomoment1^2)*(theomoment2) - 3*(
            theomoment2^2) - 4*theomoment1*theomoment3 + theomoment4
295
296
297     lines(theocumulant4~timespace,col="navyblue",lwd = 3)
298
299   }
300
301   c = cumulantfunc4(Xs,s,t,delta_t,numbsims)
302 }
303
304
305 K4_plot = CIR_cumulant4(a,b,s,t,Xs,alpha,beta,delta_t,startingstate,endstate,numbsims)
306 labels = c("Theoretical", "Emperical")
307 legend("bottomright", title = NA,labels,lty = c(1,3), lwd = c(2,3) ,col=c("navyblue","
        lightblue1"), bty = 'n',inset =-0.025)
```

Algorithm 17 Moment truncated saddlepoint transition density approximation of the mixed-effects CIR process.

```
1  #CIR Diffusion Process Analysis
2  #General: dXt = mu(Xt,t)dt + sigma(Xt,t)dWt
3  #dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt
4  #sigma ~ N(a,b^2)
5
6  rm(list=ls(all=TRUE))
7
8  #Seed:
9  set.seed(7)
10
11 #Parameters
12 s               = 0
13 t               = 5
14 Xs              = 2.75
15 alpha           = 0.8
16 beta            = 3
17 delta_t         = 0.1    #step length
18 startingstate   = 0
```

```
endstate        = 5
timespace       = seq(s,t,delta_t)
statespace      = seq(startingstate,endstate,delta_t)
a               = 0.25
b               = 0.15
randsimsactual = 100

par(ps=10,cex.lab=1,cex.axis=1,mar=c(3.5,3.5,3.5,2.5), mgp=c(2.5, 1, 0), las=1)

#saddlepoint Approx

timespace = t-s

b11 = alpha*beta
g1  = (1 - exp(-alpha*timespace))

y1 = (1/alpha)*g1

theomoment1 = Xs*exp(-alpha*timespace) + b11*y1

b22 = 2*alpha*beta + (b^2+a^2)
b11 = alpha*beta

h1 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^1
h2 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2

q1 = Xs*(1/alpha)*h1
q2 = b11*(1/(2*alpha^2))*h2

theomoment2 = (Xs^2)*exp(-2*alpha*(timespace)) + (b22)*(q1+q2)


b33 = 3*alpha*beta + 3*(b^2+a^2)
b22 = 2*alpha*beta + (b^2+a^2)
b11 = alpha*beta

d1 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)
d2 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2
d3 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)^3

p1  = ((Xs^2)/alpha)*d1
p2  = (((Xs)*(b22))/(2*alpha^2))*d2
p3  = ((beta)/(6*alpha^2))*b22*d3
```

```
62
63 theomoment3 = (Xs^3)*exp(-3*alpha*(timespace)) + b33*(p1+p2+p3)
64
65
66 b44 = 4*alpha*beta + 6*(b^2+a^2)
67 b33 = 3*alpha*beta + 3*(b^2+a^2)
68 b22 = 2*alpha*beta + (b^2+a^2)
69 b11 = alpha*beta
70
71 f1 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)
72 f2 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^2
73 f3 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^3
74 f4 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^4
75
76 w1 = (Xs^3)*(1/alpha)*f1
77 w2 = b33*(Xs^2)*(1/(2*alpha^2))*f2
78 w3 = Xs*b33*b22*(1/(6*alpha^3))*f3
79 w4 = b33*b22*b11*(1/(24*alpha^4))*f4
80
81 theomoment4 = (Xs^4)*exp(-4*alpha*(timespace)) + b44*(w1 + w2 + w3 + w4)
82
83 #theoretical Cumulants
84 theocumulant1 = theomoment1
85 theocumulant2 = theomoment2-(theomoment1)^2
86 theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
87 theocumulant4 = -6*(theomoment1^4) + 12*(theomoment1^2)*(theomoment2) - 3*(theomoment2^2)
       - 4*theomoment1*theomoment3 + theomoment4
88
89 X = statespace
90
91 s = (1/theocumulant3)*(sqrt(theocumulant2^2 - 2*theocumulant3*(theocumulant1-X)) -
       theocumulant2)
92 Ksapprox = theocumulant1*s + theocumulant2*((1/2)*s^2) + theocumulant3*((1/6)*s^3) +
       theocumulant4*((1/24)*s^4)
93 Ks2approx = theocumulant2 + theocumulant3*s + 0.5*theocumulant4*s^2
94
95 saddle_pt_approx = exp(Ksapprox-s*X)*sqrt(1/(2*pi*Ks2approx))
96 #print(saddle_pt_approx)
97
98 plot(saddle_pt_approx~X,col = "orchid4",lwd = 3, type="l", ylab = "Density", xlab='Xt')
99 labels = c("saddlepoint approx")
100 legend("topleft", title = NA,labels,lty = 1 ,lwd = 3, col=c("orchid4"), bty = 'n')
```

Algorithm 18 Moment-truncated saddlepoint transition density approximation for the mixed-effects and scalar CIR diffusion process.

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)dt + sigma(Xt,t)dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt
#sigma ~ N(a,b^2)

rm(list=ls(all=TRUE))

#Seed:
set.seed(7)

#Parameters
s               = 0
t               = 5
Xs              = 2.75
alpha           = 0.8
beta            = 3
delta_t         = 0.1   #step length
startingstate   = 0
endstate        = 5
timespace       = seq(s,t,delta_t)
statespace      = seq(startingstate,endstate,delta_t)
a               = 0.25
b               = 0.15
randsimsactual = 100

par(ps=10,cex.lab=1,cex.axis=1,mar=c(3.5,3.5,3.5,2.5), mgp=c(2.5, 1, 0), las=1)

#mixed-effects Model, sigma random

#saddlepoint Approx

timespace = t-s

b11 = alpha*beta
g1  = (1 - exp(-alpha*timespace))

y1 = (1/alpha)*g1

theomoment1 = Xs*exp(-alpha*timespace) + b11*y1

```

```
b22 = 2*alpha*beta + (b^2+a^2)
b11 = alpha*beta

h1 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^1
h2 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2

q1 = Xs*(1/alpha)*h1
q2 = b11*(1/(2*alpha^2))*h2

theomoment2 = (Xs^2)*exp(-2*alpha*(timespace)) + (b22)*(q1+q2)


b33 = 3*alpha*beta + 3*(b^2+a^2)
b22 = 2*alpha*beta + (b^2+a^2)
b11 = alpha*beta

d1 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)
d2 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2
d3 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)^3

p1  = ((Xs^2)/alpha)*d1
p2  = (((Xs)*(b22))/(2*alpha^2))*d2
p3  = ((beta)/(6*alpha^2))*b22*d3

theomoment3 = (Xs^3)*exp(-3*alpha*(timespace)) + b33*(p1+p2+p3)


b44 = 4*alpha*beta + 6*(b^2+a^2)
b33 = 3*alpha*beta + 3*(b^2+a^2)
b22 = 2*alpha*beta + (b^2+a^2)
b11 = alpha*beta

f1 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)
f2 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^2
f3 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^3
f4 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^4

w1 = (Xs^3)*(1/alpha)*f1
w2 = b33*(Xs^2)*(1/(2*alpha^2))*f2
w3 = Xs*b33*b22*(1/(6*alpha^3))*f3
w4 = b33*b22*b11*(1/(24*alpha^4))*f4

theomoment4 = (Xs^4)*exp(-4*alpha*(timespace)) + b44*(w1 + w2 + w3 + w4)
```

```
84
85  #theoretical Cumulants
86  theocumulant1 = theomoment1
87  theocumulant2 = theomoment2-(theomoment1)^2
88  theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
89  theocumulant4 = -6*(theomoment1^4) + 12*(theomoment1^2)*(theomoment2) - 3*(theomoment2^2)
        - 4*theomoment1*theomoment3 + theomoment4
90
91  X = statespace
92
93  s = (1/theocumulant3)*(sqrt(theocumulant2^2 - 2*theocumulant3*(theocumulant1-X)) -
        theocumulant2)
94  Ksapprox = theocumulant1*s + theocumulant2*((1/2)*s^2) + theocumulant3*((1/6)*s^3) +
        theocumulant4*((1/24)*s^4)
95  Ks2approx = theocumulant2 + theocumulant3*s + 0.5*theocumulant4*s^2
96
97  saddle_pt_approx = exp(Ksapprox-s*X)*sqrt(1/(2*pi*Ks2approx))
98  #print(saddle_pt_approx)
99
100 plot(saddle_pt_approx~X,col = "orchid4",lwd = 3, type="l", ylab = "Density", xlab='Xt',
        ylim = c(0,1.2))
101
102
103 #Original Model sigma fixed
104 sigma = 0.25
105
106 #saddlepoint Approx
107
108 #theoretical Moment
109 del = Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs)+beta*(beta + ((sigma^2)/(2*alpha)
        ))+2*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
110 gamma = alpha*(Xs^2 + (beta + ((sigma^2)/(2*alpha)))*(beta-2*Xs))+3*alpha*beta*(beta + ((
        sigma^2)/(2*alpha)))+4*alpha*(beta + ((sigma^2)/(2*alpha)))*(Xs-beta)
111 kappa = 2*(alpha^2)*beta*(beta + ((sigma^2)/(2*alpha)))
112 A = kappa/(6*alpha^3)
113 C = -4*((1/(4*alpha^2))*(gamma-9*A*alpha^2)-(1/(2*alpha))*(del-3*alpha*A))
114 B = (1/(2*alpha))*(del-3*alpha*A-alpha*C)
115 D = -A-B-C
116
117 gamma_star = Xs^3 + 3*(alpha*beta + sigma^2)*(A + B + C + D)
118 lambda_star = 3*alpha*Xs^3 + 3*(alpha*beta + sigma^2)*(6*alpha*A + 5*alpha*B + 4*alpha*C
        + 3*alpha*D)
119 omega_star = 2*(alpha^2)*Xs^3 + 3*(alpha*beta + sigma^2)*(11*(alpha^2)*A + 6*(alpha^2)*B
```

```
    + 3*(alpha^2)*C + 2*(alpha^2)*D)
nu_star = 3*(alpha*beta + sigma^2)*(6*A*alpha^3)

E = nu_star/(24*alpha^4)
I = (-1/(6*alpha^3))*(((omega_star-(13*nu_star/(12*alpha)))-12*(alpha^2)*(gamma_star-(nu_
    star/(24*alpha^3))))-4*alpha*((lambda_star-(3*nu_star/(8*alpha^2)))-7*alpha*((gamma_
    star-(nu_star/(24*alpha^3))))))
H = (1/(2*alpha^2))*(((lambda_star-(3*nu_star/(8*alpha^2)))-7*alpha*((gamma_star-(nu_star
    /(24*alpha^3))))) - 6*(alpha^2)*I)
G = (-1/alpha)*((gamma_star-(nu_star/(24*alpha^3)))+ 2*alpha*H + 3*alpha*I)
FF = -E - G - H - I

theomoment1 = Xs*exp(-alpha*(t-s)) + beta*(1 - exp(-alpha*(t-s)))
theomoment2 = (Xs^2)*exp(-2*alpha*(t-s)) + (beta + (sigma^2)/(2*alpha))*(beta + 2*(Xs-
    beta)*exp(-alpha*(t-s)) + (beta - 2*Xs)*exp(-2*alpha*(t-s)))
theomoment3 = (Xs^3)*exp(-3*alpha*(t-s)) + (3*alpha*beta+3*sigma^2)*(A + B*exp(-alpha*(t-
    s)) + C*exp(-2*alpha*(t-s)) + D*exp(-3*alpha*(t-s)))
theomoment4 = (Xs^4)*exp(-4*alpha*(t-s)) + (4*alpha*beta + 6*sigma^2)*(E + FF*exp(-1*
    alpha*(t-s)) + G*exp(-2*alpha*(t-s)) + H*exp(-3*alpha*(t-s)) + I*exp(-4*alpha*(t-s)))

#theoretical Cumulant
theocumulant1 = theomoment1
theocumulant2 = theomoment2-(theomoment1)^2
theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
theocumulant4 = -6*(theomoment1^4) + 12*(theomoment1^2)*(theomoment2) - 3*(theomoment2^2)
     - 4*theomoment1*theomoment3 + theomoment4

X = statespace

s = (1/theocumulant3)*(sqrt(theocumulant2^2 - 2*theocumulant3*(theocumulant1-X)) -
    theocumulant2)
Ksapprox = theocumulant1*s + theocumulant2*((1/2)*s^2) + theocumulant3*((1/6)*s^3) +
    theocumulant4*((1/24)*s^4)
Ks2approx = theocumulant2 + theocumulant3*s + 0.5*theocumulant4*s^2

saddle_pt_approx = exp(Ksapprox-s*X)*sqrt(1/(2*pi*Ks2approx))
#print(saddle_pt_approx)

lines(saddle_pt_approx~statespace,lty = 31,col = "dodgerblue1", lwd = 3)
labels = c("Original Model", "mixed-effects model")
legend("topleft", title = NA,labels,lty = c(1,1), lwd = c(3,3) ,col=c("orchid4","
    dodgerblue1"), bty = 'n')
```

Algorithm 19 Maximum likelihood estimation of the parameters of the mixed-effects CIR process's saddlepoint transition density approximation, based on the S&P500VIX dataset.

```
rm(list = ls(all = TRUE))

set.seed(7)

library(readxl)
SP500VIX <- read_excel("D:/ResearchUSB/Report_Draft_Final/Code/R_code/final_draft_code/
    original_model/SP500VIX.xlsx",
                       col_types = c("date", "numeric"))

#saddlepoint mle
X = as.matrix(na.omit(SP500VIX$VIX),nrow(na.omit(SP500VIX$VIX)),1) #1Jan2012-30Dec2016
nn=nrow(SP500VIX)
nn

#plot(SP500VIX, type = "l", ylab="VIX")

s               = 1/250
Xs              = mean(X)
Tt              = 5
numberparsim    = 1000
numberpar       = 4
sims            = 1
alpha_min       = 20
alpha_max       = 24
beta_min        = 13
beta_max        = 17
a_min           = 2
a_max           = 8
b_min           = 0.5
b_max           = 5
simulations     = 10
dt              = s

max_theta_matrix = matrix(0,simulations,numberpar)


for (m in 1:simulations)
{

  main_function = function(numberparsim,numberpar,s,t,Xs,alpha_min,alpha_max,beta_min,
```

```
    beta_max,a_min,a_max,b_min,b_max)
{
  m = numberpar + 1

  theta_matrix = matrix(0,numberparsim,m)

  for (i in 1:numberparsim)
  {
    theta_matrix[i,1] = runif(1,alpha_min,alpha_max)
    theta_matrix[i,2] = runif(1,beta_min,beta_max)
    theta_matrix[i,3] = runif(1,a_min,a_max)
    theta_matrix[i,4] = runif(1,b_min,b_max)


    alpha = theta_matrix[i,1]
    beta  = theta_matrix[i,2]
    a     = theta_matrix[i,3]
    b     = theta_matrix[i,4]


    #saddlepoint Approx

    timespace = dt

    b11 = alpha*beta
    g1  = (1 - exp(-alpha*timespace))

    y1 = (1/alpha)*g1

    theomoment1 = Xs*exp(-alpha*timespace) + b11*y1

    b22 = 2*alpha*beta + (b^2+a^2)
    b11 = alpha*beta

    h1 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^1
    h2 = (exp(-2*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2

    q1 = Xs*(1/alpha)*h1
    q2 = b11*(1/(2*alpha^2))*h2

    theomoment2 = (Xs^2)*exp(-2*alpha*(timespace)) + (b22)*(q1+q2)


```

```
82      b33 = 3*alpha*beta + 3*(b^2+a^2)
83      b22 = 2*alpha*beta + (b^2+a^2)
84      b11 = alpha*beta
85
86      d1 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)
87      d2 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)^2
88      d3 = (exp(-3*alpha*(timespace)))*(exp(alpha*(timespace))-1)^3
89
90      p1  = ((Xs^2)/alpha)*d1
91      p2  = (((Xs)*(b22))/(2*alpha^2))*d2
92      p3  = ((beta)/(6*alpha^2))*b22*d3
93
94      theomoment3 = (Xs^3)*exp(-3*alpha*(timespace)) + b33*(p1+p2+p3)
95
96
97      b44 = 4*alpha*beta + 6*(b^2+a^2)
98      b33 = 3*alpha*beta + 3*(b^2+a^2)
99      b22 = 2*alpha*beta + (b^2+a^2)
100     b11 = alpha*beta
101
102     f1 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)
103     f2 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^2
104     f3 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^3
105     f4 = exp(-4*alpha*(timespace))*(exp(alpha*(timespace))-1)^4
106
107     w1 = (Xs^3)*(1/alpha)*f1
108     w2 = b33*(Xs^2)*(1/(2*alpha^2))*f2
109     w3 = Xs*b33*b22*(1/(6*alpha^3))*f3
110     w4 = b33*b22*b11*(1/(24*alpha^4))*f4
111
112     theomoment4 = (Xs^4)*exp(-4*alpha*(timespace)) + b44*(w1 + w2 + w3 + w4)
113
114     #theoretical Cumulants
115     theocumulant1 = theomoment1
116     theocumulant2 = theomoment2-(theomoment1)^2
117     theocumulant3 = theomoment3 - 3*theomoment1*theomoment2 + 2*theomoment1^3
118     theocumulant4 = -6*(theomoment1^4) + 12*(theomoment1^2)*(theomoment2) - 3*(
            theomoment2^2) - 4*theomoment1*theomoment3 + theomoment4
119
120
121     n = nrow(X)
122     cumul_sum_vec = matrix(0,n,1)
123
```

```
      r = (1/theocumulant3)*(sqrt(theocumulant2^2 - 2*theocumulant3*(theocumulant1-X[1]))
           - theocumulant2)
      Krapprox = theocumulant1*r + theocumulant2*((1/2)*r^2) + theocumulant3*((1/6)*r^3)
          + theocumulant4*((1/24)*r^4)
      Kr2approx = theocumulant2 + theocumulant3*r + 0.5*theocumulant4*r^2
      saddle_pt_approx = exp(Krapprox-r*X[1])*sqrt(1/(2*pi*Kr2approx))
      sum1 = log(saddle_pt_approx)

      cumul_sum_vec[1] = sum1

      for (l in 2:n)
      {
        r = (1/theocumulant3)*(sqrt(theocumulant2^2 - 2*theocumulant3*(theocumulant1-X[l
            ])) - theocumulant2)
        Krapprox = theocumulant1*r + theocumulant2*((1/2)*r^2) + theocumulant3*((1/6)*r
            ^3) + theocumulant4*((1/24)*r^4)
        Kr2approx = theocumulant2 + theocumulant3*r + 0.5*theocumulant4*r^2
        saddle_pt_approx = exp(Krapprox-r*X[l])*sqrt(1/(2*pi*Kr2approx))
        log_saddle = log(saddle_pt_approx)
        sum_log_saddle = cumul_sum_vec[l-1] + log_saddle
        cumul_sum_vec[l] = sum_log_saddle
      }

      theta_matrix[i,5] = cumul_sum_vec[n]

    }

    return(theta_matrix)

}

  call_main_function = main_function(numberparsim,numberpar,s,t,Xs,alpha_min,alpha_max,
      beta_min,beta_max,a_min,a_max,b_min,b_max)


  for (i in 1:numberparsim)
  {
    if ((call_main_function[i,5]=="NaN"))
    {call_main_function[i,5] = -10000000000000000000000000000}
  }

  maximum = max(call_main_function[,5])

```

```
  for (j in 1:numberparsim)
  {
    if (call_main_function[j,5] == maximum)
    {
      max_theta = call_main_function[j,1:4]
      max_alpha = max_theta[1]
      max_beta  = max_theta[2]
      max_a     = max_theta[3]
      max_b     = max_theta[4]
    }
  }

  max_theta_matrix[m,1] = max_alpha
  max_theta_matrix[m,2] = max_beta
  max_theta_matrix[m,3] = max_a
  max_theta_matrix[m,4] = max_b

}

print(max_theta_matrix)

one = matrix(1,1,simulations)
theta_mles = (1/simulations)*one%*%(max_theta_matrix)
names(theta_mles) = c("alpha","beta","a","b")

theta_mles_fin  = as.data.frame(rbind(names(theta_mles),theta_mles),1,4)

print(theta_mles_fin)

saddle_mlesa = as.matrix(theta_mles_fin,4,1)
```

Algorithm 20 Forward simulated trajectories and 250-day prediction based on the maximum likelihood estimates for the mixed-effects CIR process.

```
#CIR Diffusion Process Analysis
#General: dXt = mu(Xt,t)*dt + sigma(Xt,t)*dWt
#dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt
rm(list = ls(all = TRUE))

library(readxl)
set.seed(7)

VIX_5year_inday0punt004steps <- read_excel("D:/ResearchUSB/Report_Draft_Final/Code/R_code
```

```
    /VIX_5year_inday0punt004steps.xlsx",
                                          col_types = c("numeric", "numeric"))
#View(VIX_5year_inday0punt004steps)

X = as.matrix(na.omit(VIX_5year_inday0punt004steps$VIX),nrow(na.omit(VIX_5year_
    inday0punt004steps$VIX)),1) #1Jan2012-30Dec2016


#Parameters
s              = 1/250
tobs           = 5
t              = tobs+1
Xs             = mean(X)
#MLE's:
alpha          = 22.1485165404156
beta           = 14.6996205216274
a              = 4.96140836593695
b              = 1.95301413626876

delta_t        = s   #step length
startingstate  = min(X)
endstate       = max(X)
timespace      = seq(s,t,delta_t)
statespace     = seq(startingstate,endstate,delta_t)
timespaceobs   = seq(s,t,s)

#Simulating the trajectory
func = function(randsims, a,b)
{
  sigma_vec          = rnorm(randsims,a,b)
  trajectory_matrix  = matrix(0,nrow = length(timespace), ncol = randsims)
  one                = matrix(1,nrow = randsims, 1)

  for (r in 1:randsims)
  {
    sigma      = sigma_vec[r]

    CIR_trajectory = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate)
    {

      timespace                = seq(s,t,delta_t)
      datamatrix               = matrix(0,nrow = length(timespace), ncol = 1)
      Z1                       = rnorm(1,mean = 0, sd = sqrt(delta_t))
```

```
51        Xt                        = Xs + alpha*(beta-Xs)*delta_t + sigma*sqrt(Xs)*Z1
52        datamatrix[1]             = Xt
53        trajectory_matrix[,1]     = datamatrix[1]
54
55        for(i in 2:length(timespace))
56        {
57          dWt             = rnorm(1,mean = 0, sd = sqrt(delta_t))
58          Xtplus1         = Xt  + alpha*(beta-Xt)*delta_t + sigma*sqrt(Xt)*dWt
59          Xt              = Xtplus1
60          datamatrix[i] = Xtplus1
61        }
62        return(datamatrix)
63      }
64
65      trajectory_matrix[,r] = CIR_trajectory(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,
            endstate)
66
67    }
68
69    #print(trajectory_matrix)
70
71    plot(trajectory_matrix[,1]~timespace,type ='l', col = rainbow(1, start = runif
          (1,0.55,0.85), end = runif(1,0.65,0.75),alpha = 0.15), xlab="years",ylab = "VIX",
          ylim=c(-5,40),xlim=c(0,6))
72    text(1,0, "FORECAST:", col = "midnightblue", cex = 1.1, adj = c(-9.5,-21))
73    abline(v = 5, col = "midnightblue", lty = 3, lwd = 3)
74
75    color = rainbow(randsims-1, start = .5, end = .7)
76    for (plotnumb in 2:randsims)
77    {
78      lines(trajectory_matrix[,plotnumb]~timespace,col = rainbow(1, start = runif
            (1,0.55,0.85), end = runif(1,0.65,0.85),alpha = 0.15), type = 'l')
79    }
80
81    mean_trajectory = (1/randsims)*trajectory_matrix %*% one
82    #print(mean_trajectory)
83
84    par(new=T)
85    plot(VIX_5year_inday0punt004steps,ylab = NA, xlab=NA,type = "l", col="violetred",ylim =
           c(-5,40),xlim=c(0,6))
86    lines(mean_trajectory~timespace, col = "midnightblue",  type = 'l', lwd = 1)
87
88  }
```

```
89
90
91
92  par(mfrow=c(4,1),ps=10,cex.lab=1,cex.axis=1,mar=c(2.5,2.5,2.5,2.5), mgp=c(1.5, 0.75, 0),
        las=1)
93
94
95  func(10,a,b)
96  labels = c("10 Simulations", "Average", "VIX data")
97  legend("bottomleft", title = NA, labels,lty = c(1,1,1), lwd = c(5,2,2) ,col=c(rainbow(1,
        start = 0.7, end = 0.75 ,alpha = 0.25),"midnightblue","violetred"), bty = 'n',horiz=T
        )
98
99  func(25,a,b)
100 labels = c("25 Simulations", "Average","VIX data")
101 legend("bottomleft", title = NA, labels,lty = c(1,1,1), lwd = c(5,2,2) ,col=c(rainbow(1,
        start = 0.7, end = 0.75 ,alpha = 0.35),"midnightblue","violetred"), bty = 'n',horiz=T
        )
102
103 func(100,a,b)
104 labels = c("100 Simulations", "Average","VIX data")
105 legend("bottomleft", title = NA, labels,lty = c(1,1,1), lwd = c(5,2,2) ,col=c(rainbow(1,
        start = 0.7, end = 0.75 ,alpha = 0.5),"midnightblue","violetred"), bty = 'n',horiz=T)
106
107 func(250,a,b)
108 labels = c("250 Simulations", "Average","VIX data")
109 legend("bottomleft", title = NA, labels,lty = c(1,1,1), lwd = c(5,2,2) ,col=c(rainbow(1,
        start = 0.7, end = 0.75 ,alpha = 0.6),"midnightblue","violetred"), bty = 'n',horiz =
        T)
```

Algorithm 21 State frequency diagrams based on 1 and 10 simulations.

```
1   #CIR Diffusion Process Analysis
2   #General: dXt = mu(Xt,t)dt + sigma(Xt,t)dWt
3   #dXt = alpha*(beta-Xt)*dt + sigma*sqrt(Xt)*dWt
4   #sigma ~ N(a,b^2)
5
6   library(RColorBrewer)
7   library(hexbin)
8
9
10  rm(list=ls(all=TRUE))
11
```

```
#Seed:
set.seed(7)

#Parameters

s              = 0
t              = 5
Xs             = 2.75
alpha          = 0.8
beta           = 3
delta_t        = 0.01    #step length
startingstate  = 0
endstate       = 5
numbsims       = 1000
timespace      = seq(s,t,delta_t)
statespace     = seq(startingstate,endstate,delta_t)
N              = length(timespace)
X              = rep(Xs, numbsims)

a              = 0.25 #sigma ~ N(a,b^2)
b              = 0.15
randsims       = 1


rf <- colorRampPalette(rev(brewer.pal(11,'Spectral')))
r <- rf(32)


heatplot = function(s,t,Xs,alpha,beta,sigma,delta_t,startingstate,endstate,numbsims)
{

  mufunc = function(Xt,t)
  {
    return(alpha*(beta - Xt))
  }

  sigfunc = function(Xt,t)
  {
    return(sigma*sqrt(Xt))
  }


  Xt         = rep(Xs,numbsims)
```

```
  timespace = seq(s,t,delta_t)

  heatmat    = matrix(0,length(timespace),1)
  heatmat[1] = Xs

  for(i in 2:length(timespace))
  {
    dWt  = sqrt(delta_t)*rnorm(numbsims)
    Xt   = Xt + mufunc(Xt, timespace[i])*delta_t + sigfunc(Xt,timespace[i])*dWt
    heatmat[i] = Xt
  }

  return(heatmat)
}

sigma_vec  = rnorm(randsims,a,b)

Xt1 = as.vector(heatplot(s,t,Xs,alpha,beta,sigma=sigma_vec[1],delta_t,startingstate,
    endstate,numbsims))

X = cbind(timespace,Xt1)
X = as.data.frame(X)

for (r in 2:randsims)
{
  X = as.matrix(X,randsims,nxol(X))
  sigma = sigma_vec[r]
  Xtr = as.vector(heatplot(s,t,Xs,alpha,beta,sigma=sigma_vec[r],delta_t,startingstate,
      endstate,numbsims))
  timeXtr = cbind(timespace,Xtr)
  times_Xts = as.data.frame(X)
  X = rbind(X,times_Xts)

}

my_frame = data.frame(X)

names(my_frame) = c("t","Xt")

hexbinplot(Xt~t, data=my_frame, colramp=rf,type = c("g"), ybnds = "data",xbins=15)
```

B3 SAS algorithm

Algorithm 22 SAS code for the S&P500VIX data AR(1)-fitted model, 150 year forecast.

```
data SP;
set work.SP500VIX;
Zt = VIX;
t=_n_;
dZt = dif(Zt);
run;


goptions reset=all i=join;
axis1 label=('VIX');
axis2 label=('days');
legend1 label=('PLOTS:') value=('VIX' 'VIX first difference');
symbol1 color=blue width=2;
symbol2 color=red width=1.5;
title 'Timeplot of VIX and VIX first difference';
proc gplot data=SP;
        plot (Zt dZt)*t / overlay legend=legend1 vaxis=axis1 haxis=axis2;
run;

title 'AR(1) - fitted';
proc arima data=SP out=reg1;
        identify var=Zt scan esacf minic p=(0:3) q=(0:3) stationarity=(adf=(0,1,2,3,4,5))
            ;
        estimate p=1 method=ml plot;
        forecast lead=250;
run;

data reg2;
set reg1;
t = _n_;
run;


goptions reset=all i=join;
axis1 label=('Residual');
axis2 label=('t');
symbol1 color=aqua width=2;
title 'Timeplot of residuals vs time(t)';
proc gplot data=reg2;
```

```
        plot residual*t / vaxis=axis1 haxis=axis2;
run;

goptions reset=all i=join;
axis1 label=('Residual');
axis2 label=('prediction');
symbol2 color=green width=1.5;
title 'Timeplot of residual vs prediction';
proc gplot data=reg2;
        plot residual*forecast /  vaxis=axis1 haxis=axis2;
run;

proc corr data=reg2;
        var forecast Zt;
run;

proc univariate data=reg2;
        histogram residual / normal (mu = est sigma = est color=blue w=2);
        qqplot residual / normal (mu = est sigma = est color=blue w=2) square;
        probplot residual / normal (mu = est sigma = est color=blue w=2) square;
run;



goptions reset=all i=join;
axis1 order=5 to 30 by 5 label=('VIX');
axis2 order=1000 to 1400 by 50 label=('days');
legend1 label=('Plots:') value=('VIX data' 'forecast' 'l95' 'u95');
symbol1 color=purple width=1;
symbol2 color=blue width=1;
symbol3 color=pink line=2 width=1;
symbol4 color=pink line=2 width=1;
title 'Partially shown VIX trajectory and 250 day forecast';
proc gplot data=reg2;
        plot (Zt forecast l95 u95)*t / overlay legend=legend1 vaxis=axis1 haxis=axis2
            href=1258;

run;
```

Exploring the MDL principle of model selection

Paul van Tonder 14101328

STK795 Research Report

Submitted in partial fulfillment of the degree BCom(Hons) Statistics

Supervisor: Dr. FHJ Kanfer

Department of Statistics, University of Pretoria

30 October 2017

Abstract

Often we face situations where we have to decide which model *summarize* our data best for a specific purpose. Model selection has thus become an important task in a statistical analysis and it can be done in various ways. Using Bayesian information criterion (BIC), Akaike information criterion (AIC) or the Minimum Description Length (MDL) principle, etc. Where the MDL principle measures the amount of variation in information and will be the main focus of this report, along with how Huffman coding can be used to measure the information captured through the regression model. The MDL principle uses the complexity of each description when comparing competing models. From the theory of algorithmic complexity by Kolmogorov, this approach started. Further developed in the literature on information theory, and have been developing in statistics recently.

Declaration

I, *Paul Isak van Tonder,* declare that this essay, submitted in partial fulfillment of the degree *BCom(Hons) Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_ _

Paul Isak van Tonder

_ _

Dr FHJ Kanfer

_ _

Date

Acknowledgments

I would like to give my gratitude to the contributors towards my research and degree.

To my supervisor, Dr Kanfer, I give thanks for his time, supervision and constructive guidance during my research.

I give thanks towards my financial contributors.

I acknowledge the financial support from the Center for AI Research, Meraka Institute, CSIR and STATOMET for the duration of my honors degree. Any findings and conclusions in this report is not the liability of the financial contributors, but that of the author(s).

My gratitude towards my family for their motivation and emotional support.

Contents

List of Figures

List of Tables

1 Introduction

The problem often arises as to which model *summarizes* the data best. The Minimum Description Length (MDL) principle is proposed to assist in selecting the better model. Stine [9] stated: "The description length is the length of this code for the data plus the length of a description of the model itself." A statistical model can be described as a code, for compressing data into a sequence of bits [9]. The MDL principle chooses the model with the smallest sequence of bits, balancing complexity and fit [9]. The length of code needed to describe the data is the fit, and the length of code needed to describe the model structure is the complexity. Therefor complexity and fit must be combined for a model selection criterion to produce an aggregate figure of merit for selection of the better model [1]. The MDL of the data will be calculated based on the concept of Huffman coding introduced by David Huffman in 1952. It is important to note that two types of compression exist. Lossless compression being compressed data that would be able to reproduce the original data exactly without losing any data, whereas with lossy compression one would not be able to reproduce the original data exactly. Huffman coding is a well known method for lossless data compression with variable-length codes. Where code of variable lengths is used to represent the data based on their frequency of occurrence. That is, we assign shorter codes to the more frequently occurring symbols and longer codes to the symbols that occur infrequently. It will be compared with fixed length codes such as ASCII (American Standard Code for Information Interchange) which requires a fixed amount of 8 bits to represent each character in a document. The expected lengths of these two types of code are then calculated and compared to the entropy. Entropy was introduced by Claude Shannon in 1948, as the equation that provides a way to estimate the average minimum number of bits needed to encode a symbol, based on the symbols frequency of occurrence. Shannon entropy provides a lower bound that can be achieved for data compression. The average length of Huffman coding may not always equal entropy, but comes close and satisfy the following equation,

$$H(Y) \leq average\ length\ of\ Huffman\ code \leq H(Y) + 1, \tag{1}$$

where $H(Y)$ represents entropy.

Having the equation of entropy, the equations of complexity and fit are then derived on that concept. Noting the similarity between the MDL principle and alternative model selection methods as the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). The equations of complexity and fit are then applied to different models. The sum of complexity and fit for each model will result in the description lengths of each model. Finally the model with the minimum description length (smallest sequence of bits) can be selected, as the model that will represent the data best. Before presenting a practical application, it is of importance to discuss some background theory and consider the derivations

of the equations to be used.

2 Theoretical discussion

2.1 Concept of Huffman coding

David Huffman went to Ohio State University, but the unusual part of his BS Electrical Engineering degree was that he received it in 1944, at the age of 18 [8]. Huffman served in the US Navy where after he finished his MS degree at Ohio in 1949 and finished his PhD at MIT in 1953 [8]. The famous Huffman code developed in 1952 in a final term paper which Huffman wrote at MIT, where Robert Fano (his professor) gave a question on shortest variable-length coding, given the probability of occurrence. In [8] Salomon mentioned: "It should be noted that in the late 1940s, Fano himself (and independently, also Claude Shannon) had developed a similar, but suboptimal, algorithm known today as the Shannon–Fano method." There was a difference in the code of Huffman and in the code of Shannon-Fano. In a sense that the Huffman code tree is build from the bottom upwards, whereas the Shannon-Fano code tree was constructed from the top downwards. Huffman coding is known to at least equal the efficiency of Shannon-Fano coding and is generally seen as optimal.

To explain how Huffman coding works, in Figure 1 it is seen that given an example set of data symbols and their probabilities or, their frequencies of occurrence, a set of variable-length code-words are allocated to the symbols, see [8]. Given the probabilities as $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{8}$ for symbols $y = \{y_1, y_2, y_3, y_4\}$ respectively. From the frequency of occurrence for each symbol, shorter code-words are allocated to symbols that occur more frequently and longer code-words for lower frequency symbols. The code-words, binary code, consist out of a sequence of bits, 0's and 1's, used in an unambiguous manner and should be prefix-free. A code is said to be prefix-free if no code-word is the prefix, the first part, of another code-word. To see how to allocate this binary code to the probabilities, the Huffman tree is used. The Huffman tree was constructed in Figure 1 by arranging the probabilities of each symbol in ascending or descending order. Then grouping the smallest two probabilities by connecting them with branches and adding the two probabilities together. The process is then repeated, each time connecting the smallest two branch probabilities together. When grouping the final two branches, the probabilities should equal to 1. In doing so, a binary code of 0's and 1's are allocated to the branches on both sides of the joined probabilities. Allocating the binary codes in a systematic way, such that the 0 are always assigned to the branch on the same side of the joined probabilities. On completion, allocate the binary codes to the symbols. In Figure 1 it was done by reading off the binary code from top to bottom. The order of the code-word is important, because if the sequence of binary code is read off backwards, the resulting code will not be prefix-free. This variable-length binary codes were presented in Table 1 column *Code V*.

Note that in Table 1, a fixed length code-words example are given in column *Code F*, where equal length code-words were allocated to the symbols, not determined by using Huffman's method, but an arbitrary method.

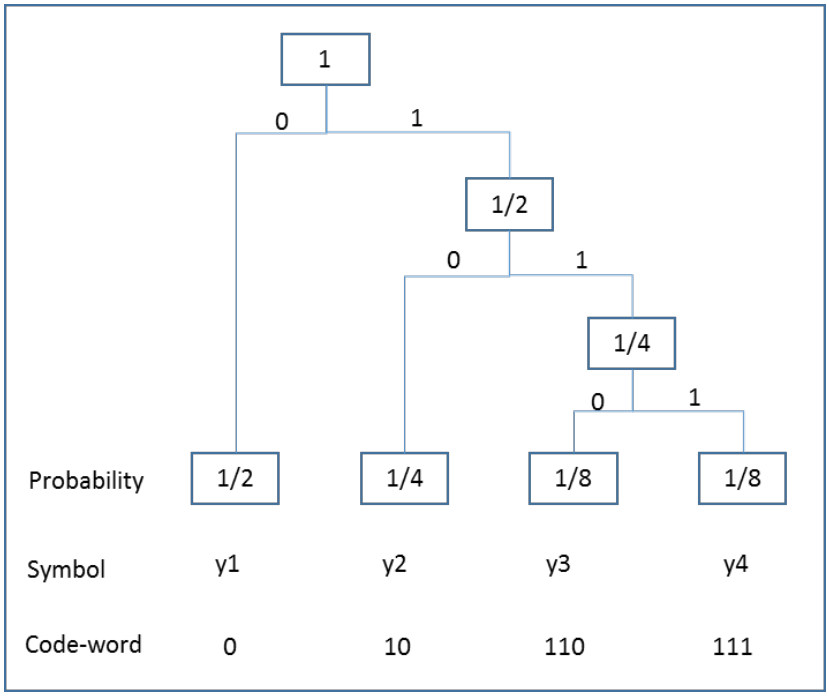


Figure 1: Huffman tree for creating variable length code-words

The code length are the amount of bits needed to encode a symbol y, and is represented by $l(y)$. Let $p(y)$ be the probability of occurrence for the symbol y.

The theorem of Claude Shannon [4] shows that the optimal code length of a symbol y is related to the probability $p(y)$ of the symbol y,

$$l(y) = -log_2 p(y). \tag{2}$$

Using this equation, from the example in Figure 1, the example code lengths of the variable length code-words can be determined and denoted by $l_V(y_i)$ for symbol y_i, $i = 1, 2, 3, 4$. Let $p(y_i)$ be the probability of occurrence for the symbol y_i, $i = 1, 2, 3, 4$, then

$$l_V(y_i) = -log_2 p(y_i) \tag{3}$$

where V indicate the optimal length encoder. Note that in Table 1, the example symbol probabilities comply with equation (3), and the Huffman coding is thus optimal, with equality in equation (1). In the cases where equation (3) indicate a fractional bit count, the upper bound in equation (1) will be relevant, making Huffman coding almost optimal, see Figure 2. The fixed code, *Code F*, with a bit count of 2

for each symbol, is not optimal. The fixed lengths and variable lengths are given in columns $l_F(y_i)$ and $l_V(y_i)$ respectively in Table 1.

i	Symbol y_i	$p(y_i)$	Code F (Fixed)	$l_F(y_i)$	Code V (Variable)	$l_V(y_i)$
1	y_1	0.5	00	2	0	1
2	y_2	0.25	01	2	10	2
3	y_3	0.125	10	2	110	3
4	y_4	0.125	11	2	111	3

Table 1: Code length comparisons

From Hansen and Yu in [4] it is stated that the correspondence between the variable code length and probability can also be reversed, and satisfy the Kraft inequality,

$$\sum_{y \in \Theta} 2^{-l(y)} \leq 1 \tag{4}$$

where symbol y for $y \in \Theta$, the domain with positive mass.

The Kraft inequality proof is given by Cover and Thomas [2].

This inequality states that the sum of the probabilities should be smaller or equal to 1, with equality if the coding scheme is efficient, or optimal. Thus, if we know the code length, using equation (4), we can calculate the probability of occurrence for symbol y as

$$p(y) = 2^{-l(y)} \tag{5}$$

for all y in the symbol set. That is, $2^{-l(y)}$ behaves like a probability function, it is non-negative and sums to unity. There is thus a one-to-one relationship between optimal coding schemes and probability functions. The Kraft inequality relates the notation of *information* and uncertainty explained by a probability function.

The Kraft inequality can be illustrated on the example symbol probabilities. From Table 1, for instance, if the length of the fourth symbol y_i, $i = 4$, is known as 3. The probability of occurrence can be determined as

$$\begin{aligned} p(y_i) &= 2^{-l_V(y_i)} \\ p(y_4) &= 2^{-l_V(y_4)} \\ &= 2^{-(3)} \\ p(y_4) &= 0.125. \end{aligned} \tag{6}$$

2.2 Entropy

Consider a random variable Y with probability distribution $p(y)$ for $y \in \Theta$, the domain with a positive mass. The minimum length of bits to code y, $y \in \Theta$, is given by equation (2) as $-log_2 p(y)$. The expected minimum length to code Y is thus given by

$$\begin{aligned} E(l(Y)) &= \sum_{y \in \Theta} -p(y) log_2 p(y) \\ &= H(Y). \end{aligned} \tag{7}$$

The expected minimum, [4], is know as Shannon entropy or just as entropy. In [9] it is stated that: "On average, the length $l(y)$ of any lossless code is at least the entropy [2]." Since Huffman coding is lossless, it achieves the entropy lower bound and is seen as an optimal code. At most Huffman coding will be 1 unit above entropy.

In [9] it is seen that Shannon's theorem states, that

$$E(l(Y)) \geq H(Y). \tag{8}$$

It follows that $H(Y) \leq E(l(Y)) \leq H(Y) + 1$ with

$$E(l(Y)) = \sum_{y \in \Theta} p(y) l(y). \tag{9}$$

From the example in Table 1, knowing the probability of occurrence $p(y_i)$ for y_i, $i = 1, 2, 3, 4$. With V the optimal length encoder, knowing the code lengths $l_V(y_i)$, the expected minimum length for *Code* V can be denoted and determined as

$$\begin{aligned} E(l_V(y_i)) &= \sum_{i=1}^{4} p(y_i) l_V(y_i) \\ &= (0.5)(1) + (0.25)(2) + (0.125)(3) + (0.125)(3) \\ &= 1.75. \end{aligned} \tag{10}$$

The entropy $H(y_i)$ for the probability distribution given in Table 1 is calculated as

$$
\begin{aligned}
H(y_i) &= \sum_{i=1}^{4} -p(y_i) log_2 p(y_i) \\
&= -0.5 log_2 0.5 - 0.25 log_2 0.25 - 0.125 log_2 0.125 - 0.125 log_2 0.125 \\
&= 1.75.
\end{aligned} \tag{11}
$$

The expected minimum length of *Code F* can be calculated as

$$
\begin{aligned}
E(l_F(y_i)) &= \sum_{i=1}^{4} p(y_i) l_F(y_i) \\
&= (0.5)(2) + (0.25)(2) + (0.125)(2) + (0.125)(2) \\
&= 2
\end{aligned} \tag{12}
$$

which is larger than $H(y_i)$, thus not an optimal code.

In the example above, it is seen that the expected length of *Code V* is shorter than those of *Code F*, and that it equals the entropy. Making *Code V* an optimal code, also showing that Huffman coding can reach the entropy lower bound.

i	Symbol y_i	$p(y_i)$	Code F (Fixed)	$l_F(y_i)$	Code V (Variable)	$l_V(y_i)$
1	y_1	0.5	00	2	0	1
2	y_2	0.25	01	2	10	2
3	y_3	0.1875	10	2	110	$2.415 \approx 3$
4	y_4	0.0625	11	2	111	4

Table 2: Code length comparisons using fractional variable code-word lengths

Table 2 presents a slight different probability distribution as compared to Table 1. The table also include minimum length $-log_2 p(y_i)$, which indicate fractional code-word lengths. Rounding the code lengths to the next integer, code-words as in Figure 1 have been selected. It is shown in Figure 2 that the fractional code-word lengths had resulted in no changes to the structure of the Huffman tree. However, the number of bits, for example, needed to represent the symbols are $l_V(y_3) = 2.415 \approx 3$ (rounded) and $l_V(y_4) = 4$.

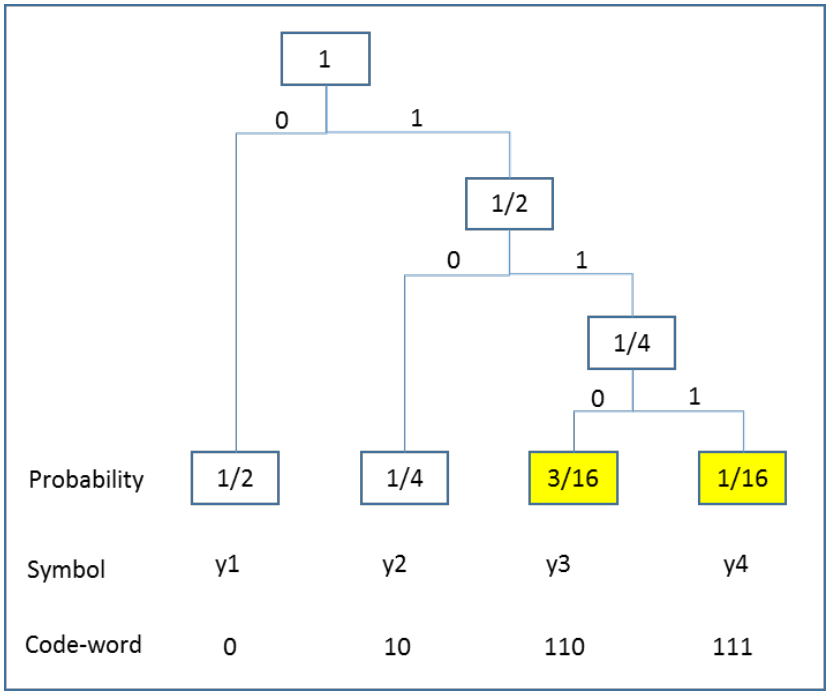


Figure 2: Huffman tree for creating variable length code-words using fractional code-word lengths

The expected length of *Code V* was then determined as

$$\begin{aligned} E(l_V(y_i)) &= \sum_{i=1}^{4} p(y_i) l_V(y_i) \\ &= (0.5)(1) + (0.25)(2) + (0.1875)(3) + (0.0625)(4) \\ &= 1.8125 \end{aligned} \tag{13}$$

and the corresponding entropy can be calculated as

$$\begin{aligned} H(y_i) &= \sum_{i=1}^{4} -p(y_i) log_2 p(y_i) \\ &= -0.5 log_2 0.5 - 0.25 log_2 0.25 - 0.1875 log_2 0.1875 - 0.0625 log_2 0.0625 \\ &= 1.70281. \end{aligned} \tag{14}$$

Thus the expected length did not reach entropy, but is within one unit,

$$H(Y) \leq average\ length\ of\ Huffman\ code \leq H(Y) + 1. \tag{15}$$

2.3 Model selection

Model selection is to determine and select the model that best *summarize* the data. Thus coding the data and the model in terms of code lengths. The code length measures the model coding effectiveness. A model associate a probability to the observed data. That probability, in turn, suggest an optimal code length to representing the data [9]. The ideal is a minimum number of bits (shortest code) to *summarize* the data and the model structure used. The *best* model produces the shortest code. What the *best* model and coding have in common, is that they both maximize the likelihood of the observed data. Thus, the best code for a random variable $Y \sim p$, has an idealized length of $l(y) = -log_2 p(y)$ [9]. The length of this code for representing a sample $Y_1, Y_2, ..., Y_n \sim p(y)$ is the negative log-likelihood [9],

$$\begin{aligned} l(y_1, ..., y_n) &= -log_2 L(y_1, ..., y_n) \\ &= -log_2 p(y_1)p(y_2)...p(y_n). \end{aligned} \tag{16}$$

The expected length of any other code will be larger. If coding just maximize the likelihood, over fit needs to be avoided. It is overcome by the code identifying the model used, as to which method was used to encode the data. The code length will therefor be represented by two components, one to identify the data when encoded with the chosen model and a component to identify the model structure itself. Thus description length, DL, is defined as the combination of code length required to represent the data plus the code length required to describe the model structure [9]. In order to determine the DL of a model, based on the concept of Huffman coding, consider the following. Let y denote n response observations $y1$, $y2$, ..., y_n. Let M_θ denote the parametric model indexed by parameter vector θ and let $L(y \mid M_\theta)$ denote the likelihood given the model. Let $\hat{\theta}(y)$ be the maximum likelihood estimator, since maximizing the likelihood, obtain the shortest code for the data. Let $l(M_{\hat{\theta}(y)})$ denote the number of bits required to represent the fitted model. The DL is then given by

$$D(y;\, M_{\hat{\theta}(y)}) = l(M_{\hat{\theta}(y)}) - log_2 L(y \mid M_{\hat{\theta}(y)}). \tag{17}$$

In the above equation, the first term of the equation represents complexity, the code length required to identify the model structure. The second term represents fit, the length of code required to identify the data. Before calculating the DL's, the equation above needs to be considered. In a regression model, consider how to encode a parameter, for example, the slope. The best data summary through a model is obtained by using the maximum likelihood estimator (MLE) of the parameter. The coding for the model has to identify $\hat{\theta}(y)$, because the MLE depends on y. Least square slopes are often integers values. Rissanen showed in [7] that, rather than identifying $\hat{\theta}(y)$ exactly, we only need to calculate how many

standard errors (SE) it lies from 0. Deriving $\tilde{\theta}(y)$ from $\hat{\theta}(y)$ as

$$\begin{aligned}\tilde{\theta}(y) &= SE(\hat{\theta}(y)) \left\langle \frac{\hat{\theta}(y)}{SE(\hat{\theta}(y))} \right\rangle \\ &= SE(\hat{\theta}(y))\, z_{\tilde{\theta}(y)},\end{aligned} \tag{18}$$

where $\langle x \rangle$ denotes the integers closest to x. Therefor there is no need to encode $\hat{\theta}(y)$, but rather encode, for example, $z_{\tilde{\theta}} = 4$, meaning the estimate lies four SE's above zero. The likelihood and code's ability to *summarize* data, are hardly effected by this rounding. In practice the rounded estimates are used to calculate DL. From equation (17) and (18), the DL is derived as

$$D(y;\, M_{\hat{\theta}(y)}) \approx l(M_{\tilde{\theta}(y)}) - log_2 L(y \mid M_{\hat{\theta}(y)}). \tag{19}$$

Encoding integer z scores will be the last step in the process of determining the MDL. Before continuing, consider the following equation,

$$AIC(\hat{\theta}(y)) = dim(\hat{\theta}(y)) - log_2 L(y \mid M_{\hat{\theta}(y)}). \tag{20}$$

It shows that alternative model selection methods, such as the AIC, selects the model with a minimum penalized likelihood. Where $dim(\hat{\theta}(y))$ is used to denote the parameter dimension. Comparing equation (20) with (19), it is shown that the code length $l(M_{\tilde{\theta}(y)})$ used in equation (19) to identify the model, fill the role of a penalty factor in equation (20). To determine the amount of bits needed to present the model, the rounded z scores need to be encoded. Since the rounded z scores have integer values, assuming they are each bounded in size by $|\tilde{z}| < B/2$. Given B, for $|\tilde{z}|$ is identified by its binary expansion $1 + log_2 B$ and setting the bound $B = \sqrt{n}$ [7]. Then the MDL criteria result to

$$D(y;\, M_{\hat{\theta}(y)}) = dim(\theta)(1 + \frac{1}{2} log_2 n) - log_2 L(y \mid M_{\hat{\theta}(y)}). \tag{21}$$

For large n, the first term plays the role of the BIC penalty. From the work of Elias [3], Rissanen [6], a universal code to represent integers were proposed. It assigns probabilities that decrease as $\tilde{z}$ moves away from zero, rather than dividing the code evenly over a bounded area. Where an integer, for $j \neq 0$, the ideal universal code length is approximated by

$$l_u(j) = 2 + log_2^+ |j| + 2 log_2^+ log_2^+ |j|\,, \tag{22}$$

with $log_2^+(x)$ as the positive log function. Where $log_2^+(x) = 0$ for $|x| \leq 1$. A collection of rounded z scores are denoted by [9],

$$\tilde{z}_j = \left\langle \frac{\hat{\theta}_j}{SE(\hat{\theta}_j)} \right\rangle, j = 1, ..., dim(\theta), \tag{23}$$

Then the DL becomes

$$D(y; M_{\hat{\theta}(y)}) = \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) - log_2 L(y \mid M_{\hat{\theta}(y)}). \tag{24}$$

Assuming the structure of the model is known in advance and not determined from the data, and the predictors in the model does not need to be identified by the code. Referring back to equation (19), under the assumption of normality and using the rounded t statistics, complexity can be calculated as

$$l(M_{\tilde{\theta}(y)}) = \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j), \tag{25}$$

and the fit can be calculated as

$$-log_2 L(y \mid M_{\hat{\theta}(y)}) = \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right), \tag{26}$$

where $ESS(\hat{\theta})$ is the error sum of squares for the fitted model.

The DL for the regression model is then given by

$$D(theoretical\, regression) = \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) + \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right). \tag{27}$$

The model with the lowest DL, balancing complexity and fit, is then the MDL.

3 Application

A data sample was taken from lake Laengelmavesi near Tampere in Finland [5]. The sample has 157 observations of randomly selected fish. The sample has seven variables, including species, three lengths, height, width and weight. The Weight of the fish was the dependent variable from the given data. Weight will be known as (Y) and was given in grams. The estimated Weight will be determined as $\hat{Y}$. The independent variables will be Species, Length1, Length2, Length3, Height% and Width%. Length1 (L_1) indicate a measurement from the nose to the beginning of the tail in centimeters. Length2 (L_2) indicate a measurement from the nose to the notch of the tail in centimeters. Length3 (L_3) was measured from the nose to the end of the tail in centimeters. The maximal Height% (H) and Width% (W) was measured as a percentage of Length3. Species will be represented by dummy variables, 1 if the selected specie occur, 0 otherwise. The different species are illustrated in Table 3.

Remark 1. Images in Table 3 are free to use or share, even commercially.

Species	Image	Species	Image
Bream		Roach	
Parkki		Smelt	
Perch		Whitefish	
Pike			

Table 3: Fish species

We want to determine which model will describe our data best for a given purpose.

For Model 1, the model was selected as

$$Y = B_0 + B_1L_1 + B_2L_2 + B_3L_3 + B_4H + B_5W + B_6D_1 + B_7D_2 + B_8D_3 + B_9D_4 + B_{10}D_5 + B_{11}D_6.$$

The model is simple and can be seen as a full model since it includes all the independent variables. This model was selected because it contains all the predictor variables, but the high number of variables may lead to a high complexity value.

The fit of Model 1 was determined as

$$\begin{aligned} -log_2L(y \mid M_{\hat{\theta}(y)}) &= \frac{n}{2}log_2\left(\frac{ESS(\hat{\theta})}{n}\right) \\ &= \frac{157}{2}log_2\left(\frac{1241215.8}{n}\right) \\ &= 10.121661 \end{aligned}$$

and the complexity was determined as

$$\begin{aligned} l(M_{\tilde{\theta}(y)}) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) \\ &= l_u(15) + l_u(2) + l_u(2) + l_u(1) + l_u(0) + l_u(0) + l_u(1) + l_u(2) + l_u(1) + l_u(5) + l_u(0) + l_u(8) \\ &= 9.8389323 + 3 + 3 + 2 + 2 + 2 + 2 + 3 + 2 + 6.7525747 + 2 + 8.169925 \\ &= 45.761432. \end{aligned}$$

The DL for regression Model 1 was then determined as

$$\begin{aligned} D(theoretical\, regression) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) + \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right) \\ &= 45.761432 + 10.121661 \\ &= 55.883093. \end{aligned}$$

When looking at the t-values and their respective p-values, most of them are significantly larger than 0.05. The respective null hypotheses $B_k = 0$ are accepted for $k = 2, 3, 4, 5, 6, 7, 8, 10$. The relatively high value of $R^2 = 0.938199$ with few significant t statistics is the one indicator of multicollinearity. Several high values in the correlation matrix suggest the same. Since there exist high correlations between the three length measurements, only one length measurement can be used. Knowing that a fish is three dimensional and since volume has a linear relationship with Weight, calculating the volume of a fish could be a suitable approximation of Weight. In determining the volume, we should remember that Height% and Width% was measured as a maximal percentage of Length3. The Height can thus be approximated by $Height = Height\% * Length3/100$ and the Width can be approximated by $Width = Width\% * Length3/100$. Thus approximating Volume as

$$Volume1 \propto \frac{Length3^3 * Height\% * Width\%}{100^2}.$$

Then Model 2 was selected as

$$Y = B_0 + B_1 Volume1.$$

The fit of Model 2 was determined as

$$\begin{aligned} -log_2 L(y \mid M_{\hat{\theta}(y)}) &= \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right) \\ &= \frac{157}{2} log_2 \left(\frac{2218926.9}{n} \right) \\ &= 10.540715 \end{aligned}$$

and the complexity was determined as

$$\begin{aligned} l(M_{\tilde{\theta}(y)}) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) \\ &= l_u(13) + l_u(35) \\ &= 9.4758331 + 11.846797 \\ &= 21.322631. \end{aligned}$$

The DL for regression Model 2 was then determined as

$$\begin{aligned} D(theoretical\, regression) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) + \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right) \\ &= 21.322631 + 10.540715 \\ &= 31.863346. \end{aligned}$$

It is shown that Model 2 perform significantly better that Model 1 when comparing the two DL's, due to a significantly lower complexity value in Model 2, even though there was not a large difference between the fit of the two models. In Figure 3 and 4 below, a graphical presentation of Model 1 and 2 was given. The model fitted to the data was used to determine the predicted Weights, $\hat{Y}$, and was given on the x-axis. Furthermore, the observed Weights, Y, which was given in the data set was plotted on the y-axis. The ideal model would then be plotted as a positive linear 45 degree line from the origin. A graphical presentation thus allows the reader to see graphically as to which model best *summarize* the data.

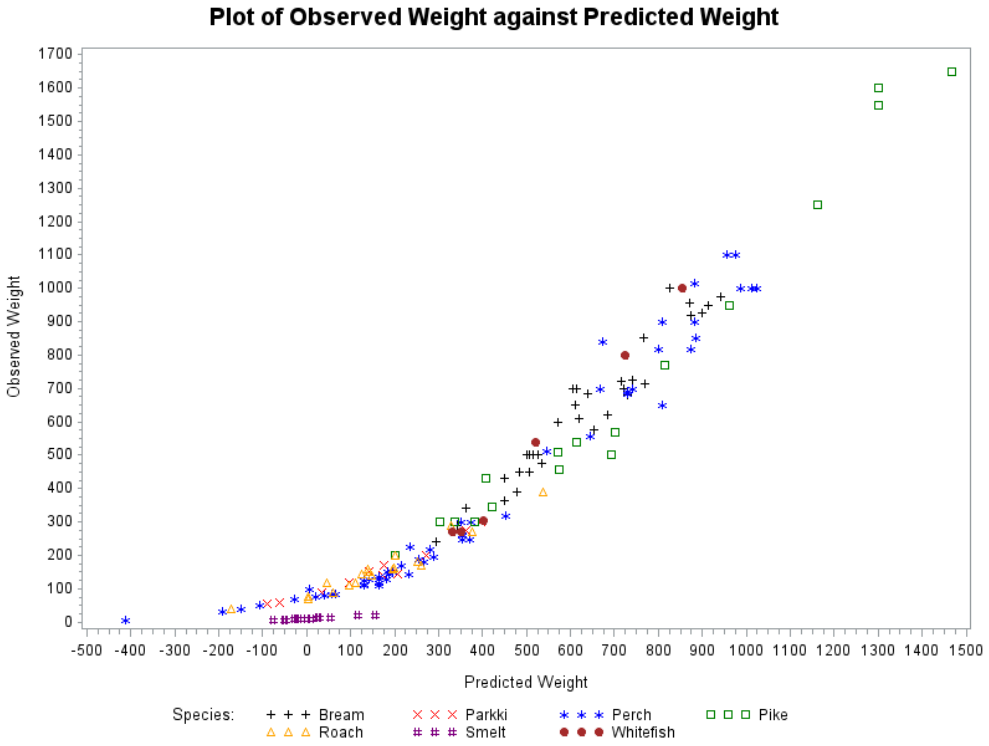


Figure 3: Model 1 fitted to the data

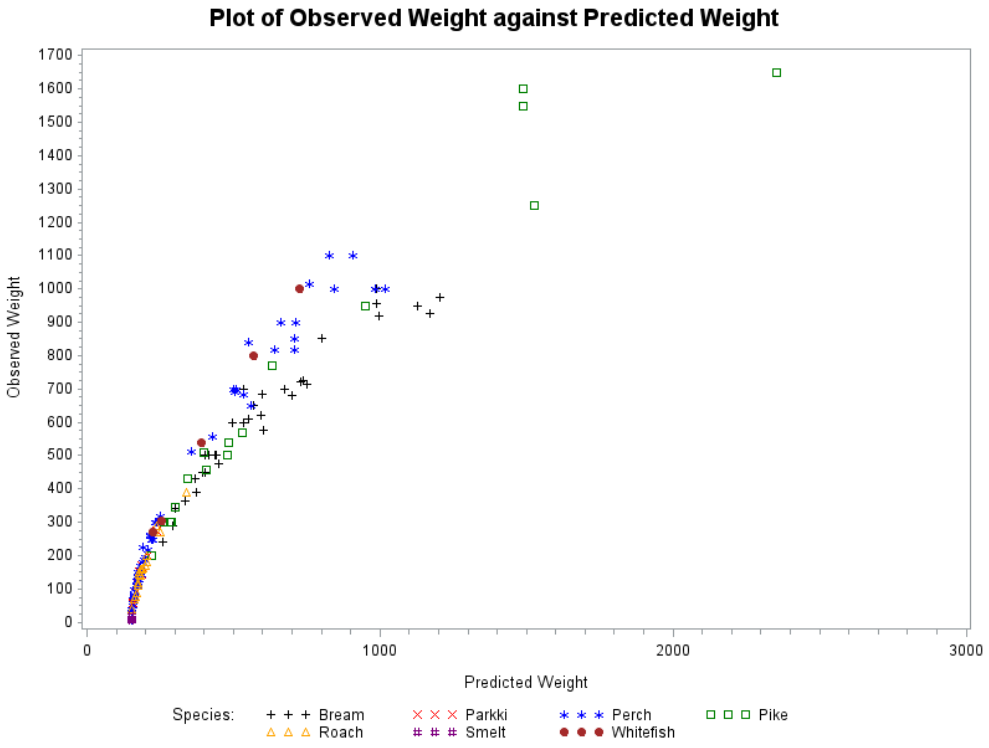


Figure 4: Model 2 fitted to the data

Suppose for different species, there are different linear relationships, since all seven species have diversified shapes. A different model can be used for each individual specie. But due to the lack of

observations per specie, indicated in Table 4, it won‘t be an accurate representation of the data.

Species	Frequency
Bream	34
Parkki	11
Perch	56
Pike	17
Roach	19
Smelt	14
Whitefish	6

Table 4: Frequencies of fish species in the sample

For Perches it might be worthwhile to determine its own model since the observation number is large enough. Should the modeler have insufficient knowledge about the different fish species or as in the instance of not having enough observation per specie, one could also use the next model.

It is known that a fish are not plane and unidimensional, as seen in Model 1. A cubic relationship is thus explored. Model 3 was then selected as

$$\sqrt[3]{Y} = B_0 + B_1 L3 + B_2 H + B_3 W.$$

The data tends to be non-linear, thus $\frac{1}{3}$ seems to be a reasonable power of Y, since a fish are three dimensional. Then the fit of Model 3 was determined as

$$\begin{aligned} -log_2 L(y \mid M_{\hat{\theta}(y)}) &= \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right) \\ &= \frac{157}{2} log_2 \left(\frac{7.9960594}{n} \right) \\ &= 1.4996446 \end{aligned}$$

and the complexity was determined as

$$\begin{aligned} l(M_{\tilde{\theta}(y)}) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) \\ &= l_u(1) + l_u(31) + l_u(14) + l_u(22) \\ &= 2 + 11.571498 + 9.6649331 + 10.773151 \\ &= 34.009583. \end{aligned}$$

The DL for regression Model 3 was then determined as

$$\begin{aligned} D(theoretical\,regression) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) + \frac{n}{2} log_2\left(\frac{ESS(\hat{\theta})}{n}\right) \\ &= 34.009583 + 1.4996446 \\ &= 35.509227. \end{aligned}$$

According to the DL's, Model 2 is preferred over Model 3, because Model 2 had a lower DL of 31.863346 compared to Model 3 with a DL of 35.509227. Even though Model 2 might have a better DL than Model 3, it is observed in Table 6 that Model 3 have better R^2 and R_a^2 values than those of Model 2. Furthermore, note that from the first three models explored, Model 3 had a significantly lower fit of 1.5, rounded, compared to Model 1 and 2 with rounded values of 10.1 and 10.5 respectively. In Figure 5 it was clearly shown that Model 3 had a better fit compared to the first two models. Yet it might suggest over fit.

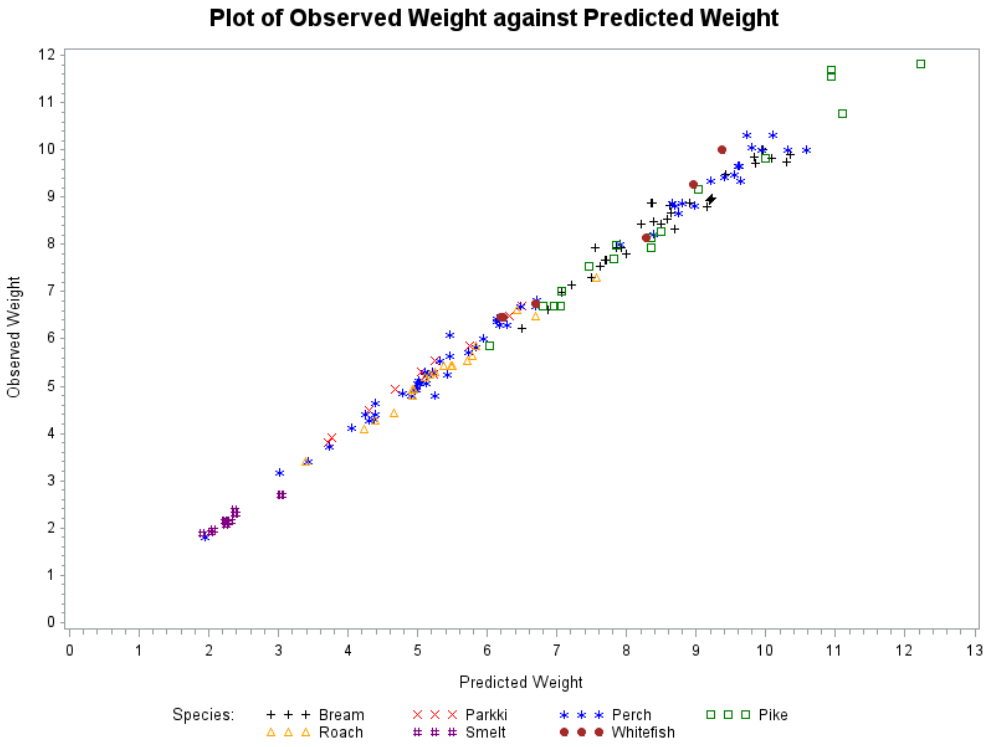


Figure 5: Model 3 fitted to the data

In the next model to be tested, the length needed to calculate the volume will be changed. Knowing that volume consists out of three dimensions. Almost as in Model 2, where Length3 was used for all three the dimensions, where Height% and Width% were maximal percentages of Length3. What if the length used for the volume approximation were now changed from Length3 to Length2. Since the measurement from the nose to the end of the tail may not be an accurate representation in determining the weight,

because of a lack of weight in the tail. Yet, the length in the tail should not be disregarded. Therefor using Length 2, which is the measurement from the nose to the notch of the tail. Then Model 4 was given by

$$Y = B_0 + B_1 Volume2$$

where Volume is now determined as

$$Volume2 \propto \frac{Length2 * (Length3 * Height\%) * (Length3 * Width\%)}{100^2}.$$

The fit of Model 4 was determined as

$$\begin{aligned} -log_2 L(y \mid M_{\hat{\theta}(y)}) &= \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right) \\ &= \frac{157}{2} log_2 \left(\frac{2209633.2}{n} \right) \\ &= 10.537688 \end{aligned}$$

and the complexity was determined as

$$\begin{aligned} l(M_{\tilde{\theta}(y)}) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) \\ &= l_u(13) + l_u(35) \\ &= 9.4758331 + 11.846797 \\ &= 21.322631. \end{aligned}$$

The DL for regression Model 4 was then determined as

$$\begin{aligned} D(theoretical\, regression) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) + \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right) \\ &= 21.322631 + 10.537688 \\ &= 31.860318. \end{aligned}$$

The DL of Model 4 is very close the DL of Model 2, both would round to 32 bits. Indicating that correlation exist between the two lengths, since it had no significant effect on either the fit nor the complexity, such that a difference in DL only occur from the third decimal. Noting that their R^2 and R_a^2 have no significant difference up to the third decimal as well.

Up to this point, Model 2 and 4 are the best models with regard to their, almost identical, DL's. Would the different lengths used, to approximate the volume, still present the same DL if Species were to be considered in the two models? And will the addition of Species improve the models further, or would it increase the complexity of the model excessively? The following two models were explored. Model 5, where Species was added to Model 2, given by

$$Y = B_0 + B_1 Volume1 + B_2 D_1 + B_3 D_2 + B_4 D_3 + B_5 D_4 + B_6 D_5 + B_7 D_6.$$

Model 6, where Species was added to Model 4, given by

$$Y = B_0 + B_1 Volume2 + B_2 D_1 + B_3 D_2 + B_4 D_3 + B_5 D_4 + B_6 D_5 + B_7 D_6.$$

The fit of Model 5 was determined as

$$\begin{aligned} -log_2 L(y \mid M_{\hat{\theta}(y)}) &= \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right) \\ &= \frac{157}{2} log_2 \left(\frac{1690713.3}{n} \right) \\ &= 10.3446 \end{aligned}$$

and the complexity was determined as

$$\begin{aligned} l(M_{\tilde{\theta}(y)}) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) \\ &= l_u(12) + l_u(31) + l_u(2) + l_u(1) + l_u(2) + l_u(0) + l_u(2) + l_u(5) \\ &= 9.2688786 + 11.571498 + 3 + 2 + 3 + 2 + 3 + 6.7525747 \\ &= 40.592952. \end{aligned}$$

The DL for regression Model 5 was then determined as

$$\begin{aligned} D(theoretical\, regression) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) + \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right) \\ &= 40.592952 + 10.3446 \\ &= 50.937552. \end{aligned}$$

It was shown in Figure 6 that the fit of Model 5 are similar to the fit of Model 2, but not identical. From the calculation of complexity for Model 5, it is shown that adding the Species variable to Model 2, did indeed increase the complexity value. Thus confirming that complexity increase as the independent

variables increase. This was expected. From the definition of complexity, it was known that, complexity is the amount of length needed to identify the model structure.

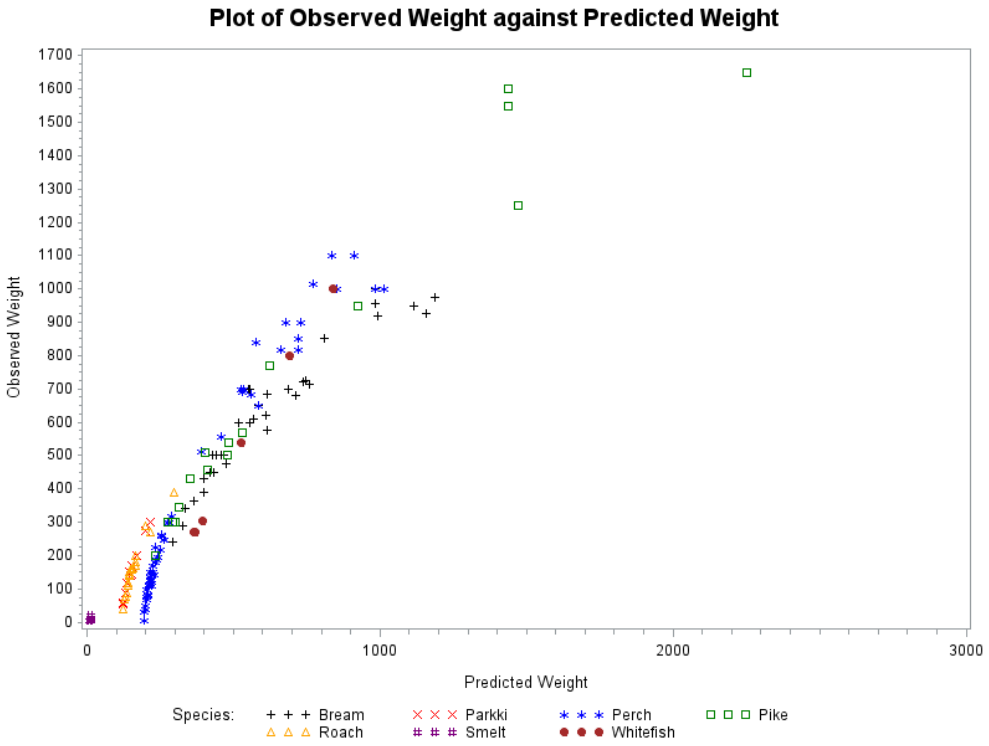


Figure 6: Model 5 fitted to the data

For Model 6 the fit was determined as

$$\begin{aligned} -log_2 L(y \mid M_{\hat{\theta}(y)}) &= \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right) \\ &= \frac{157}{2} log_2 \left(\frac{1634569.9}{n} \right) \\ &= 10.32024 \end{aligned}$$

and the complexity was determined as

$$\begin{aligned} l(M_{\tilde{\theta}(y)}) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) \\ &= l_u(13) + l_u(32) + l_u(3) + l_u(1) + l_u(2) + l_u(0) + l_u(2) + l_u(6) \\ &= 9.4758331 + 11.643856 + 4.9138599 + 2 + 3 + 2 + 3 + 7.3252492 \\ &= 43.358798. \end{aligned}$$

The DL for regression Model 6 was then determined as

$$\begin{aligned} D(theoretical\,regression) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) + \frac{n}{2} log_2\left(\frac{ESS(\hat{\theta})}{n}\right) \\ &= 43.358798 + 10.32024 \\ &= 53.679038. \end{aligned}$$

It is observed that with Species in the model, the DL's of Model 5 and 6 does in fact differ. Model 5 and 6 have similar fit values, but different complexity values. Therefore the graphical representations are expected to be very similar as well. Yet neither one of the two have been observed to be optimal. If Model 3 is used as before, it is expected that when adding Species to the model, the complexity would increase as well. Yet it is the fit result that is achieved that is interesting. For Model 7 given by

$$\sqrt[3]{Y} = B_0 + B_1 L3 + B_2 H + B_3 W + B_4 D_1 + B_5 D_2 + B_6 D_3 + B_7 D_4 + B_8 D_5 + B_9 D_6$$

The fit of Model 7 was determined as

$$\begin{aligned} -log_2 L(y \mid M_{\hat{\theta}(y)}) &= \frac{n}{2} log_2\left(\frac{ESS(\hat{\theta})}{n}\right) \\ &= \frac{157}{2} log_2\left(\frac{5.53748}{n}\right) \\ &= 1.2346148 \end{aligned}$$

and the complexity was determined as

$$\begin{aligned} l(M_{\tilde{\theta}(y)}) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) \\ &= l_u(2) + l_u(18) + l_u(5) + l_u(7) + l_u(2) + l_u(2) + l_u(4) + l_u(3) + l_u(1) + l_u(1) \\ &= 3 + 10.289968 + 6.7525747 + 7.7857779 + 3 + 3 + 6 + 4.9138599 + 2 + 2 \\ &= 48.74218. \end{aligned}$$

The DL for regression Model 7 was then determined as

$$\begin{aligned} D(theoretical\,regression) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) + \frac{n}{2} log_2\left(\frac{ESS(\hat{\theta})}{n}\right) \\ &= 48.74218 + 1.2346148 \\ &= 49.976795. \end{aligned}$$

Looking back at Model 3, with a DL of 35.509227. Model 3 *summarized* the data fairly well, recalling a fit of 1.4996446. After adding the species variable to Model 3, Model 7 is observed with an even smaller fit of 1.2346148. In comparison with all models tested, Model 7 was observed with the smallest fit value. Figure 7 reflect how well Model 7 is fitted to the data. Should the Modeler deem fit to be of most importance, irrelevant of the high complexity. Then Model 7 should be chosen to *summarize* the data. Disregarding the fact that Model 7 have a high DL value compared to previous models tested.

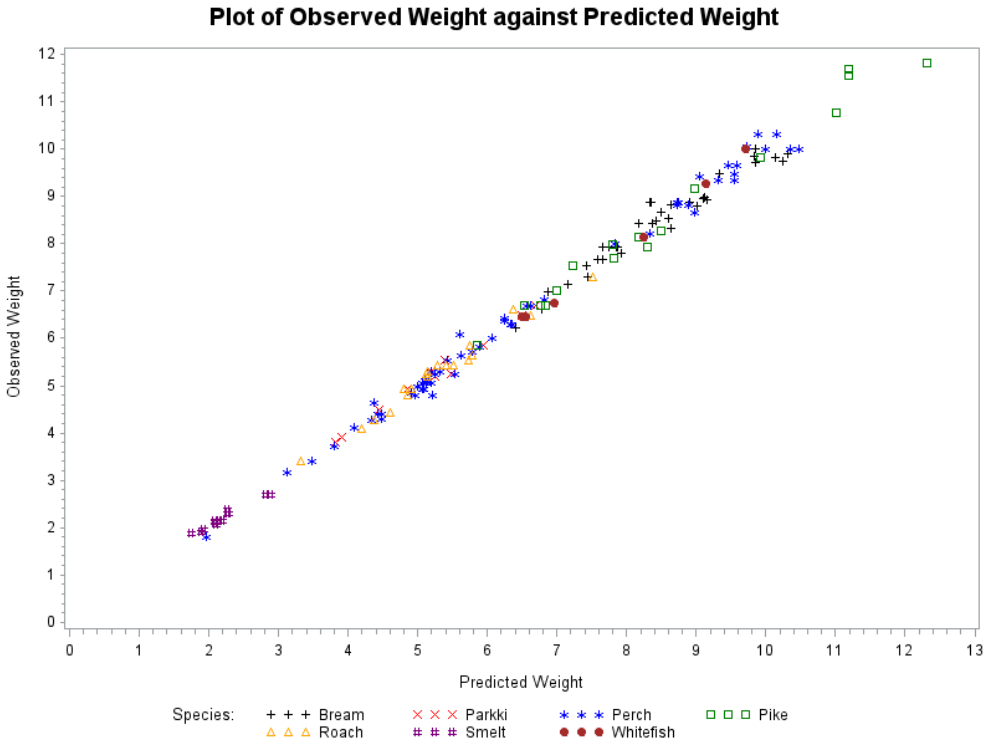


Figure 7: Model 7 fitted to the data

For the last model tested, it came to thought that Model 3 was tested using Width% and Height% as maximal percentages of Length 3. Therefore it would be interesting to see how the model would fare if the actual Width and Height measures were used, where $Height$ was approximated by $Height\% * Length3/100$ and the $Width$ was approximated by $Width\% * Length3/100$. Model 8 was then given by

$$\sqrt[3]{Y} = B_0 + B_1 L3 + B_2 H(L3/100) + B_3 W(L3/100).$$

The fit of Model 8 was determined as

$$\begin{aligned} -log_2 L(y \mid M_{\hat{\theta}(y)}) &= \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right) \\ &= \frac{157}{2} log_2 \left(\frac{37.212465}{n} \right) \\ &= 2.608857 \end{aligned}$$

and the complexity was determined as

$$\begin{aligned} l(M_{\tilde{\theta}(y)}) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) \\ &= l_u(10) + l_u(7) + l_u(8) + l_u(6) \\ &= 8.7859698 + 7.7857779 + 8.169925 + 7.3252492 \\ &= 32.066922. \end{aligned}$$

The DL for regression Model 8 was then determined as

$$\begin{aligned} D(theoretical\, regression) &= \sum_{j=1}^{dim(\theta)} l_u(\tilde{z}_j) + \frac{n}{2} log_2 \left(\frac{ESS(\hat{\theta})}{n} \right) \\ &= 32.066922 + 2.608857 \\ &= 34.675779. \end{aligned}$$

It has been observed that Model 8 do fare better than Model 3. Not only because Model 8 have a lower DL than Model 3, with values determined as 34.675779 and 35.509227 respectively. When considering the fact that the modeler should balance complexity and fit. Model 8 seems more balanced with a fit of 2.608857, and complexity of 32.066922, compared to Model 3 with a fit of 1.4996446, and complexity of 34.009583. Figure 8 illustrate how Model 8 was fitted to the data.

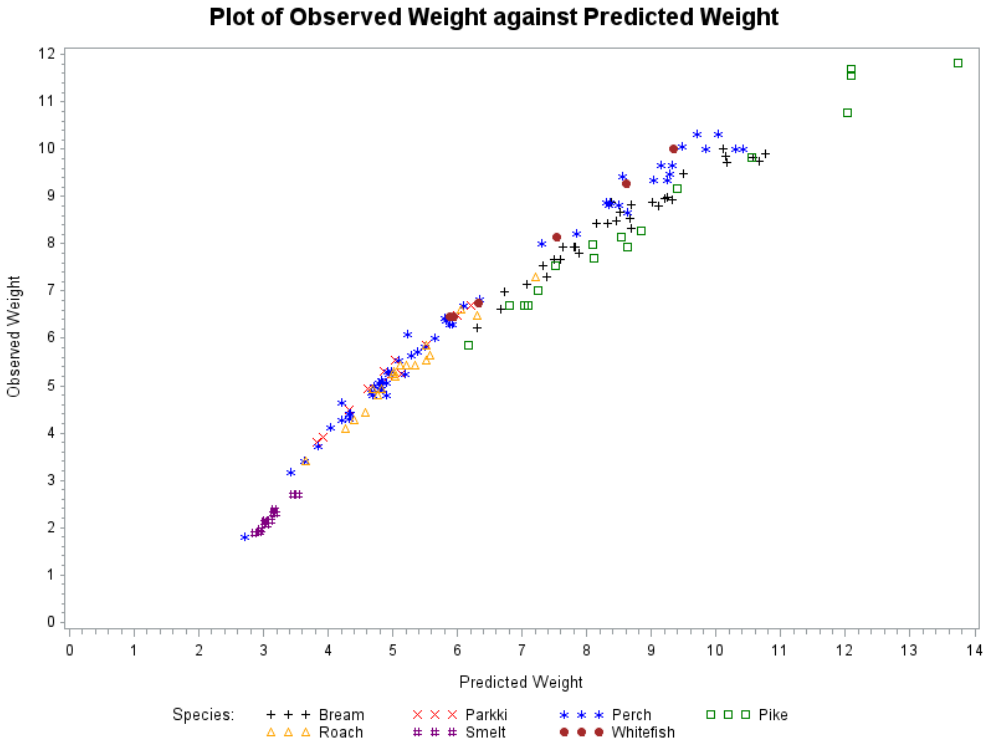


Figure 8: Model 8 fitted to the data

Comparing all eight models explored, Model 2 and 4 are both rounded to a DL of 32. Thus Model 2 and 4 can be seen as the MDL models. Stating that when a fish species are unknown, but knowing that a fish are three dimensional beings, Model 2 and Model 4 are the best models to *summarize* the data balancing complexity and fit. Should the modeler feel that the fit in Model 2 and 4 was to vague. Then Model 8 would be a sufficient alternative model to use. At the expense of 3 more bits for the rounded DL, Model 8 had a significant better fit. Thus, even though Model 8 was not the MDL, in accordance with the modelers preference for balancing complexity and fit, Model 8 could be an acceptable model to *summarize* the data.

In Table 5 a summary of all the models explored was given.

Model 1	$\mathrm{Y}=\mathrm{B}_0 + B_1L_1 + B_2L_2 + B_3L_3 + B_4H + B_5W + B_6D_1 + B_7D_2 + B_8D_3 + B_9D_4 + B_{10}D_5 + B_{11}D_6$
Model 2	$\mathrm{Y}=\mathrm{B}_0 + B_1Volume1$
Model 3	$\sqrt[3]{Y} = B_0 + B_1L3 + B_2H + B_3W$
Model 4	$\mathrm{Y}=\mathrm{B}_0 + B_1Volume2$
Model 5	$Y = B_0 + B_1Volume1 + B_2D_1 + B_3D_2 + B_4D_3 + B_5D_4 + B_6D_5 + B_7D_6$
Model 6	$Y = B_0 + B_1Volume2 + B_2D_1 + B_3D_2 + B_4D_3 + B_5D_4 + B_6D_5 + B_7D_6$
Model 7	$\sqrt[3]{Y} = B_0 + B_1L3 + B_2H + B_3W + B_4D_1 + B_5D_2 + B_6D_3 + B_7D_4 + B_8D_5 + B_9D_6$
Model 8	$\sqrt[3]{Y} = B_0 + B_1L3 + B_2H(L3/100) + B_3W(L3/100)$

Table 5: Summary of models used

In Table 6 the SAS results were *summarized* for each of the models used in Table 5. The ESS will denote the error sum of squares, R^2 will denote the coefficient of determination and R^2_a will be the

adjusted R square. The complexity, fit and DL are also shown in Table 6.

	ESS	R^2	R_a^2	Fit	Complexity	Description length
Model 1	1241215.8	0.938199	0.9335106	10.121661	45.761432	55.883093
Model 2	2218926.9	0.889518	0.8888053	10.540715	21.322631	31.863346
Model 3	7.9960594	0.9912744	0.9911033	1.4996446	34.009583	35.509227
Model 4	2209633.2	0.8899808	0.889271	10.537688	21.322631	31.860318
Model 5	1690713.3	0.9158182	0.9118633	10.3446	40.592952	50.937552
Model 6	1634569.9	0.9186136	0.9147901	10.32024	43.358798	53.679038
Model 7	5.53748	0.9939573	0.9935874	1.2346148	48.74218	49.976795
Model 8	37.212465	0.9593925	0.9585963	2.608857	32.066922	34.675779

Table 6: Summary of results per model

4 Conclusion

It can be concluded that MDL is an effective model selection criteria. For selecting the model that best summarize the data, for a specific purpose. Using MDL, it was noted that complexity and fit was the key elements of the description length. As the results indicated on the data sample taken from lake Laengelmavesi in Finland. Relatively equal description lengths can be found, with complexity and fit differently distributed. It is the task of the modeler to assign the weights of importance to these two elements when selecting a significant model. This trade off leads to an important property of the MDL principle. The trade off provides a natural safeguard against over fit. It is important to realize that the MDL principle did not state how to select the suggested models. This could be seen as one of the shortfalls of the principle. In practice, the building of models are done on human judgment or past experience, with prior knowledge of similar models. Noting that the eight suggested models used in this report was selected on human judgment. Better models may be available, yet the models used was for illustration of how the MDL principle can be used to select the best model from those suggested. The MDL principle allows for comparison of models from different types, since a general criterion is used. In comparison with the BIC and AIC, which only depend on the number of parameters used in the model. Noting that both AIC and BIC fare well when many predictors are related to the response. So which model fits best? It depends on how the modeler approach the analysis. Even though models can be imperfect, MDL can still provide reliable estimates of all the properties captured by the model. If theory allows to express something as a probability distribution, then MDL presents a framework in which its merits can be judged.

References

[1] Peter G Bryant and Olga I Cordero-Brana. Model selection using the minimum description length principle. *The American Statistician*, 54(4):257–268, 2000.

[2] Thomas M Cover and Joy A Thomas. *Elements of information theory*. New York: John Wiley, 1991.

[3] Peter Elias. Universal codeword sets and representations of the integers. *IEEE Transactions on Information Theory*, 21(2):194–203, 1975.

[4] Mark H Hansen and Bin Yu. Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96(454):746–774, 2001.

[5] J Puranen. Fish catch data set. *Journal of Statistics Education Data Archive*, 1917.

[6] Jorma Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, pages 416–431, 1983.

[7] Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry*, volume 15. World scientific, 1998.

[8] David Salomon. *A Concise Introduction to Data Compression*. Springer Science & Business Media, 2007.

[9] Robert A Stine. Model selection using information theory and the MDL principle. *Sociological Methods & Research*, 33(2):230–260, 2004.

Appendix

SAS code for the data sample taken from lake Laengelmavesi near Tampere in Finland.

```
quit;
dm 'log;clear';
dm 'odsresult;clear';

PROC IMPORT OUT= WORK.fish
            DATAFILE= "C:\PAUL\#Tuks\Honours\STK 795\sas coding\sas_data.xls"
            DBMS=EXCEL REPLACE;
     RANGE="fish";
     GETNAMES=YES;
     MIXED=NO;
     SCANTEXT=YES;
     USEDATE=YES;
     SCANTIME=YES;
RUN;

proc print data = work.fish;
run;

title "The Species Variable";
proc freq data=work.fish;
tables Species;
run;

proc iml;
print 'Model 1';
use work.fish; read all var{Weight} into Y;
use work.fish; read all var{Length1} into Length1;
use work.fish; read all var{Length2} into Length2;
use work.fish; read all var{Length3} into Length3;
use work.fish; read all var{Height} into Height;
use work.fish; read all var{Width} into Width;
use work.fish; read all var{Species} into Species;
```

```
print 'Weight=Y';
print 'Length1=L1,  Length2=L2,  Length3=L3,  Height=H,  Width=W';
print 'Species: Bream=D1, Parkki=D2, Perch=D3, Pike=D4, Roach=D5, Smelt=D6, Whitefish=D7';
print 'Y= b0 +b1L1 +b2L2 +b3L3 +b4H +b5W +b6D1 +b7D2 +b8D3 +b9D4 +b10D5 +b11D6';

n=157;
Species2=Species;
Species = designf( Species );
*print Species2;
X=J(n,1,1)||Length1||Length2||Length3||Height||Width||Species;
*print X;
k=ncol(x);
bh=inv(X'*X)*X'*Y;
H=X*inv(X'*X)*X';
yh=H*Y;
res=y-x*bh;
ESS=res'*res;
TSS=ssq(y-y[:]);
RSS=TSS-ESS;
df_ess=n-ncol(x);
df_rss=ncol(x)-1;
df_tss=n-1;
MSE=ESS/df_ess;
MSR=RSS/df_rss;
R2=RSS/TSS;
aR2=1 - ((1-R2)*((n-1)/(n-k)));
Cov_b=MSE*inv(X'*X);
se_b=sqrt(vecdiag(Cov_b));
t_values=bh/se_b;
t_int= abs(round(t_values));
p_t= 2*(1-probt(abs(t_values),df_ess));
print n k ESS , R2 aR2 , bh se_b p_t t_values t_int;
```

```
Corr=corr(x);
nm={"int" "Length1" "Length2" "Length3" "Height" "Width" "D1" "D2" "D3" "D4" "D5" "D6"};
print Corr[colname=nm rowname=nm];

Fit=(n/2)*(log2(ESS)/n);
print Fit;

do i=1 to nrow(t_int);
if t_int[i,] <=1 then L_var=2;
else L_var=2+log2(t_int[i,])+2*log2(log2(t_int[i,]));
*print L_var;

L_var_comb = L_var_comb // L_var;

end;

print L_var_comb;

Complexity=L_var_comb[+];
print Complexity;

Description_length=Fit+Complexity;
print Description_length;

create yhat from yh[colname="yh"];
append from yh;
quit;

data work.fish;
merge work.fish yhat;
run;

proc sort data =work.fish;
by Weight;
```

```
run;

goptions reset=all;
axis1 label=(angle=90 'Observed Weight');
axis2 label=('Predicted Weight');
legend1 label=('Species:')
value=('Bream' 'Parkki' 'Perch' 'Pike' 'Roach' 'Smelt' 'Whitefish');
symbol1 color=black value=plus;
symbol2 color=red value=x;
symbol3 color=blue value=star;
symbol4 color=green value=square;
symbol5 color=orange value=triangle;
symbol6 color=purple value=hash;
symbol7 color=brown value=dot;
title1 'Plot of Observed Weight against Predicted Weight';

proc gplot data=fish;
plot Weight *yh =Species / legend=legend1 vaxis=axis1 haxis=axis2;
run;

proc iml;
print 'Model 2';
use work.fish; read all var{Weight} into Y;
use work.fish; read all var{Length1} into Length1;
use work.fish; read all var{Length2} into Length2;
use work.fish; read all var{Length3} into Length3;
use work.fish; read all var{Height} into Height;
use work.fish; read all var{Width} into Width;
use work.fish; read all var{Species} into Species;

print 'Weight=Y';
print 'Length3=L3,  Height=H,  Width=W';
print 'Volume1= (L3^3 *H *W) /100^2';
print 'Y= b0 +b1Volume1';
```

```
n=157;
Species2= Species;
Species = designf( Species );
*print Species;
volume1=((Length3##3)#Height#Width)/(100##2);
*print volume1;
X=J(n,1,1)||volume1;
*print X;
k=ncol(x);
bh=inv(X‘*X)*X‘*Y;
H=X*inv(X‘*X)*X‘;
yh=H*Y;
res=y-x*bh;
ESS=res‘*res;
TSS=ssq(y-y[:]);
RSS=TSS-ESS;
df_ess=n-ncol(x);
df_rss=ncol(x)-1;
df_tss=n-1;
MSE=ESS/df_ess;
MSR=RSS/df_rss;
R2=RSS/TSS;
aR2=1 - ((1-R2)*((n-1)/(n-k)));
Cov_b=MSE*inv(X‘*X);
se_b=sqrt(vecdiag(Cov_b));
t_values=bh/se_b;
t_int= abs(round(t_values));
p_t= 2*(1-probt(abs(t_values),df_ess));
print n k ESS , R2 aR2 , bh se_b p_t t_values t_int;

Corr=corr(x);
nm={"int" "Volume1"};
print Corr[colname=nm rowname=nm];
```

```
Fit=(n/2)*(log2(ESS)/n);
print Fit;

do i=1 to nrow(t_int);
if t_int[i,] <=1 then L_var=2;
else L_var=2+log2(t_int[i,])+2*log2(log2(t_int[i,]));
*print L_var;

L_var_comb = L_var_comb // L_var;

end;

print L_var_comb;

Complexity=L_var_comb[+];
print Complexity;

Description_length=Fit+Complexity;
print Description_length;

create yhat from yh[colname="yh"];
append from yh;
quit;

data work.fish;
merge work.fish yhat;
run;

proc sort data =work.fish;
by Weight;
run;

goptions reset=all;
```

```
axis1 label=(angle=90 'Observed Weight');
axis2 label=('Predicted Weight');
legend1 label=('Species:')
value=('Bream' 'Parkki' 'Perch' 'Pike' 'Roach' 'Smelt' 'Whitefish');
symbol1 color=black value=plus;
symbol2 color=red value=x;
symbol3 color=blue value=star;
symbol4 color=green value=square;
symbol5 color=orange value=triangle;
symbol6 color=purple value=hash;
symbol7 color=brown value=dot;
title1 'Plot of Observed Weight against Predicted Weight';

proc gplot data=fish;
plot Weight *yh =Species / legend=legend1 vaxis=axis1 haxis=axis2 ;
run;

proc iml;
print 'Model 3';
use work.fish; read all var{Weight} into Y;
use work.fish; read all var{Length1} into Length1;
use work.fish; read all var{Length2} into Length2;
use work.fish; read all var{Length3} into Length3;
use work.fish; read all var{Height} into Height;
use work.fish; read all var{Width} into Width;
use work.fish; read all var{Species} into Species;

print 'Weight=Y';
print 'Length3=L3,  Height=H,  Width=W';
print 'Y^1/3= b0 +b1L3 +b2H +b3W';

n=157;
Species2= Species;
Species = designf( Species );
```

```
*print Species;
X=J(n,1,1)||Length3||Height||Width;
*print X;
k=ncol(x);
Y=Y##(1/3);
ynew=Y;
*print y;
bh=inv(X‘*X)*X‘*Y;
H=X*inv(X‘*X)*X‘;
yh=H*Y;
res=y-x*bh;
ESS=res‘*res;
TSS=ssq(y-y[:]);
RSS=TSS-ESS;
df_ess=n-ncol(x);
df_rss=ncol(x)-1;
df_tss=n-1;
MSE=ESS/df_ess;
MSR=RSS/df_rss;
R2=RSS/TSS;
aR2=1 - ((1-R2)*((n-1)/(n-k)));
Cov_b=MSE*inv(X‘*X);
se_b=sqrt(vecdiag(Cov_b));
t_values=bh/se_b;
t_int= abs(round(t_values));
p_t= 2*(1-probt(abs(t_values),df_ess));
print n k ESS , R2 aR2 , bh se_b p_t t_values t_int;

Corr=corr(x);
nm={"int" "Length3" "Height" "Width"};
print Corr[colname=nm rowname=nm];

Fit=(n/2)*(log2(ESS)/n);
print Fit;
```

```
do i=1 to nrow(t_int);
if t_int[i,] <=1 then L_var=2;
else L_var=2+log2(t_int[i,])+2*log2(log2(t_int[i,]));
*print L_var;

L_var_comb = L_var_comb // L_var;

end;

print L_var_comb;

Complexity=L_var_comb[+];
print Complexity;

Description_length=Fit+Complexity;
print Description_length;

create yhat from yh[colname="yh"];
append from yh;
create yn from ynew[colname="ynew"];
append from ynew;
quit;

data work.fish;
merge work.fish yhat yn;
run;

proc sort data =work.fish;
by ynew;
run;

goptions reset=all;
axis1 label=(angle=90 'Observed Weight');
```

```
axis2 label=('Predicted Weight');
legend1 label=('Species:')
value=('Bream' 'Parkki' 'Perch' 'Pike' 'Roach' 'Smelt' 'Whitefish');
symbol1 color=black value=plus;
symbol2 color=red value=x;
symbol3 color=blue value=star;
symbol4 color=green value=square;
symbol5 color=orange value=triangle;
symbol6 color=purple value=hash;
symbol7 color=brown value=dot;
title1 'Plot of Observed Weight against Predicted Weight';

proc gplot data=fish;
plot ynew *yh =Species / vzero hzero legend=legend1 vaxis=axis1 haxis=axis2;
run;

proc iml;
print 'Model 4';
use work.fish; read all var{Weight} into Y;
use work.fish; read all var{Length1} into Length1;
use work.fish; read all var{Length2} into Length2;
use work.fish; read all var{Length3} into Length3;
use work.fish; read all var{Height} into Height;
use work.fish; read all var{Width} into Width;
use work.fish; read all var{Species} into Species;

print 'Weight=Y';
print 'Length2=L2, Length3=L3,  Height=H,  Width=W';
print 'Volume2= (L2 *(H*L3) *(W*L3)) /100^2';
print 'Y= b0 +b1Volume2';

n=157;
Species2= Species;
Species = designf( Species );
```

```
*print Species;
volume2=(Length2#(Height#Length3)#(Width#Length3))/(100##2);
*print volume2;
X=J(n,1,1)||volume2;
*print X;
k=ncol(x);
bh=inv(X‘*X)*X‘*Y;
H=X*inv(X‘*X)*X‘;
yh=H*Y;
res=y-x*bh;
ESS=res‘*res;
TSS=ssq(y-y[:]);
RSS=TSS-ESS;
df_ess=n-ncol(x);
df_rss=ncol(x)-1;
df_tss=n-1;
MSE=ESS/df_ess;
MSR=RSS/df_rss;
R2=RSS/TSS;
aR2=1 - ((1-R2)*((n-1)/(n-k)));
Cov_b=MSE*inv(X‘*X);
se_b=sqrt(vecdiag(Cov_b));
t_values=bh/se_b;
t_int= abs(round(t_values));
p_t= 2*(1-probt(abs(t_values),df_ess));
print n k ESS , R2 aR2 , bh se_b p_t t_values t_int;

Corr=corr(x);
nm={"int" "Volume2"};
print Corr[colname=nm rowname=nm];

Fit=(n/2)*(log2(ESS)/n);
print Fit;
```

```
do i=1 to nrow(t_int);
if t_int[i,] <=1 then L_var=2;
else L_var=2+log2(t_int[i,])+2*log2(log2(t_int[i,]));
*print L_var;

L_var_comb = L_var_comb // L_var;

end;

print L_var_comb;

Complexity=L_var_comb[+];
print Complexity;

Description_length=Fit+Complexity;
print Description_length;

create yhat from yh[colname="yh"];
append from yh;
quit;

data work.fish;
merge work.fish yhat;
run;

proc sort data =work.fish;
by Weight;
run;

goptions reset=all;
axis1 label=(angle=90 'Observed Weight');
axis2 label=('Predicted Weight');
legend1 label=('Species:')
value=('Bream' 'Parkki' 'Perch' 'Pike' 'Roach' 'Smelt' 'Whitefish');
```

```
symbol1 color=black value=plus;
symbol2 color=red value=x;
symbol3 color=blue value=star;
symbol4 color=green value=square;
symbol5 color=orange value=triangle;
symbol6 color=purple value=hash;
symbol7 color=brown value=dot;
title1 'Plot of Observed Weight against Predicted Weight';

proc gplot data=fish;
plot Weight *yh =Species / legend=legend1 vaxis=axis1 haxis=axis2;
run;

proc iml;
print 'Model 5';
use work.fish; read all var{Weight} into Y;
use work.fish; read all var{Length1} into Length1;
use work.fish; read all var{Length2} into Length2;
use work.fish; read all var{Length3} into Length3;
use work.fish; read all var{Height} into Height;
use work.fish; read all var{Width} into Width;
use work.fish; read all var{Species} into Species;

print 'Weight=Y';
print 'Length3=L3,  Height=H,  Width=W';
print 'Species: Bream=D1, Parkki=D2, Perch=D3, Pike=D4, Roach=D5, Smelt=D6, Whitefish=D7';
print 'Volume1= (L3^3 *H *W) /100^2';
print 'Y= b0 +b1Volume1 +b2D1 +b3D2 +b4D3 +b5D4 +b6D5 +b7D6';

n=157;
Species2= Species;
Species = designf( Species );
*print Species;
volume1=((Length3##3)#Height#Width)/(100##2);
```

```
*print volume1;
X=J(n,1,1)||volume1||Species;
*print X;
k=ncol(x);
bh=inv(X‘*X)*X‘*Y;
H=X*inv(X‘*X)*X‘;
yh=H*Y;
res=y-x*bh;
ESS=res‘*res;
TSS=ssq(y-y[:]);
RSS=TSS-ESS;
df_ess=n-ncol(x);
df_rss=ncol(x)-1;
df_tss=n-1;
MSE=ESS/df_ess;
MSR=RSS/df_rss;
R2=RSS/TSS;
aR2=1 - ((1-R2)*((n-1)/(n-k)));
Cov_b=MSE*inv(X‘*X);
se_b=sqrt(vecdiag(Cov_b));
t_values=bh/se_b;
t_int= abs(round(t_values));
p_t= 2*(1-probt(abs(t_values),df_ess));
print n k ESS , R2 aR2 , bh se_b p_t t_values t_int;

Corr=corr(x);
nm={"int" "Volume1" "D1" "D2" "D3" "D4" "D5" "D6"};
print Corr[colname=nm rowname=nm];

Fit=(n/2)*(log2(ESS)/n);
print Fit;

do i=1 to nrow(t_int);
if t_int[i,] <=1 then L_var=2;
```

```
else L_var=2+log2(t_int[i,])+2*log2(log2(t_int[i,]));
*print L_var;

L_var_comb = L_var_comb // L_var;

end;

print L_var_comb;

Complexity=L_var_comb[+];
print Complexity;

Description_length=Fit+Complexity;
print Description_length;

create yhat from yh[colname="yh"];
append from yh;
quit;

data work.fish;
merge work.fish yhat;
run;

proc sort data =work.fish;
by Weight;
run;

goptions reset=all;
axis1 label=(angle=90 'Observed Weight');
axis2 label=('Predicted Weight');
legend1 label=('Species:')
value=('Bream' 'Parkki' 'Perch' 'Pike' 'Roach' 'Smelt' 'Whitefish');
symbol1 color=black value=plus;
symbol2 color=red value=x;
```

```
symbol3 color=blue value=star;
symbol4 color=green value=square;
symbol5 color=orange value=triangle;
symbol6 color=purple value=hash;
symbol7 color=brown value=dot;
title1 'Plot of Observed Weight against Predicted Weight';

proc gplot data=fish;
plot Weight *yh =Species / legend=legend1 vaxis=axis1 haxis=axis2;
run;

proc iml;
print 'Model 6';
use work.fish; read all var{Weight} into Y;
use work.fish; read all var{Length1} into Length1;
use work.fish; read all var{Length2} into Length2;
use work.fish; read all var{Length3} into Length3;
use work.fish; read all var{Height} into Height;
use work.fish; read all var{Width} into Width;
use work.fish; read all var{Species} into Species;

print 'Weight=Y';
print 'Length2=L2, Length3=L3,  Height=H,  Width=W';
print 'Species: Bream=D1, Parkki=D2, Perch=D3, Pike=D4, Roach=D5, Smelt=D6, Whitefish=D7';
print 'Volume2= (L2 *(H*L3) *(W*L3)) /100^2';
print 'Y= b0 +b1Volume2 +b2D1 +b3D2 +b4D3 +b5D4 +b6D5 +b7D6';

n=157;
Species2= Species;
Species = designf( Species );
*print Species;
volume2=(Length2#(Height#Length3)#(Width#Length3))/(100##2);
*print volume2;
X=J(n,1,1)||volume2||Species;
```

```
*print X;
k=ncol(x);
bh=inv(X‘*X)*X‘*Y;
H=X*inv(X‘*X)*X‘;
yh=H*Y;
res=y-x*bh;
ESS=res‘*res;
TSS=ssq(y-y[:]);
RSS=TSS-ESS;
df_ess=n-ncol(x);
df_rss=ncol(x)-1;
df_tss=n-1;
MSE=ESS/df_ess;
MSR=RSS/df_rss;
R2=RSS/TSS;
aR2=1 - ((1-R2)*((n-1)/(n-k)));
Cov_b=MSE*inv(X‘*X);
se_b=sqrt(vecdiag(Cov_b));
t_values=bh/se_b;
t_int= abs(round(t_values));
p_t= 2*(1-probt(abs(t_values),df_ess));
print n k ESS , R2 aR2 , bh se_b p_t t_values t_int;

Corr=corr(x);
nm={"int" "Volume2" "D1" "D2" "D3" "D4" "D5" "D6"};
print Corr[colname=nm rowname=nm];

Fit=(n/2)*(log2(ESS)/n);
print Fit;

do i=1 to nrow(t_int);
if t_int[i,] <=1 then L_var=2;
else L_var=2+log2(t_int[i,])+2*log2(log2(t_int[i,]));
*print L_var;
```

```
L_var_comb = L_var_comb // L_var;

end;

print L_var_comb;

Complexity=L_var_comb[+];
print Complexity;

Description_length=Fit+Complexity;
print Description_length;

create yhat from yh[colname="yh"];
append from yh;
quit;

data work.fish;
merge work.fish yhat;
run;

proc sort data =work.fish;
by Weight;
run;

goptions reset=all;
axis1 label=(angle=90 'Observed Weight');
axis2 label=('Predicted Weight');
legend1 label=('Species:')
value=('Bream' 'Parkki' 'Perch' 'Pike' 'Roach' 'Smelt' 'Whitefish');
symbol1 color=black value=plus;
symbol2 color=red value=x;
symbol3 color=blue value=star;
symbol4 color=green value=square;
```

```
symbol5 color=orange value=triangle;
symbol6 color=purple value=hash;
symbol7 color=brown value=dot;
title1 'Plot of Observed Weight against Predicted Weight';

proc gplot data=fish;
plot Weight *yh =Species / legend=legend1 vaxis=axis1 haxis=axis2;
run;

proc iml;
print 'Model 7';
use work.fish; read all var{Weight} into Y;
use work.fish; read all var{Length1} into Length1;
use work.fish; read all var{Length2} into Length2;
use work.fish; read all var{Length3} into Length3;
use work.fish; read all var{Height} into Height;
use work.fish; read all var{Width} into Width;
use work.fish; read all var{Species} into Species;

print 'Weight=Y';
print 'Length3=L3,  Height=H,  Width=W';
print 'Species: Bream=D1, Parkki=D2, Perch=D3, Pike=D4, Roach=D5, Smelt=D6, Whitefish=D7';
print 'Y^1/3= b0 +b1L3 +b2H +b3W +b4D1 +b5D2 +b6D3 +b7D4 +b8D5 +b9D6';

n=157;
Species2 =Species;
Species = designf( Species );
*print Species;
X=J(n,1,1)||Length3||Height||Width||Species;
*print X;
k=ncol(x);
Y=Y##(1/3);
ynew=y;
*print y;
```

```
bh=inv(X‘*X)*X‘*Y;
H=X*inv(X‘*X)*X‘;
yh=H*Y;
res=y-x*bh;
ESS=res‘*res;
TSS=ssq(y-y[:]);
RSS=TSS-ESS;
df_ess=n-ncol(x);
df_rss=ncol(x)-1;
df_tss=n-1;
MSE=ESS/df_ess;
MSR=RSS/df_rss;
R2=RSS/TSS;
aR2=1 - ((1-R2)*((n-1)/(n-k)));
Cov_b=MSE*inv(X‘*X);
se_b=sqrt(vecdiag(Cov_b));
t_values=bh/se_b;
t_int= abs(round(t_values));
p_t= 2*(1-probt(abs(t_values),df_ess));
print n k ESS , R2 aR2 , bh se_b p_t t_values t_int;

Corr=corr(x);
nm={"int" "Length3" "Height" "Width" "D1" "D2" "D3" "D4" "D5" "D6"};
print Corr[colname=nm rowname=nm];

Fit=(n/2)*(log2(ESS)/n);
print Fit;

do i=1 to nrow(t_int);
if t_int[i,] <=1 then L_var=2;
else L_var=2+log2(t_int[i,])+2*log2(log2(t_int[i,]));
*print L_var;

L_var_comb = L_var_comb // L_var;
```

```
end;

print L_var_comb;

Complexity=L_var_comb[+];
print Complexity;

Description_length=Fit+Complexity;
print Description_length;

create yhat from yh[colname="yh"];
append from yh;
create yn from ynew[colname="ynew"];
append from ynew;
quit;

data work.fish;
merge work.fish yhat yn;
run;

proc sort data =work.fish;
by ynew;
run;

goptions reset=all;
axis1 label=(angle=90 'Observed Weight');
axis2 label=('Predicted Weight');
legend1 label=('Species:')
value=('Bream' 'Parkki' 'Perch' 'Pike' 'Roach' 'Smelt' 'Whitefish');
symbol1 color=black value=plus;
symbol2 color=red value=x;
symbol3 color=blue value=star;
symbol4 color=green value=square;
```

```
symbol5 color=orange value=triangle;
symbol6 color=purple value=hash;
symbol7 color=brown value=dot;
title1 'Plot of Observed Weight against Predicted Weight';

proc gplot data=fish;
plot ynew *yh =Species / vzero hzero legend=legend1 vaxis=axis1 haxis=axis2;
run;

proc iml;
print 'Model 8';
use work.fish; read all var{Weight} into Y;
use work.fish; read all var{Length1} into Length1;
use work.fish; read all var{Length2} into Length2;
use work.fish; read all var{Length3} into Length3;
use work.fish; read all var{Height} into Height;
use work.fish; read all var{Width} into Width;
use work.fish; read all var{Species} into Species;

print 'Weight=Y';
print 'Length3=L3,  Height=H,  Width=W';
print 'Y^1/3= b0 +b1L3 +b2H(L3/100) +b3W(L3/100)';

n=157;
Species2= Species;
Species = designf( Species );
*print Species;
X=J(n,1,1)||Length3||Height#(Length3/100)||Width#(Length3/100);
*print X;
k=ncol(x);
Y=Y##(1/3);
ynew=y;
*print y;
bh=inv(X'*X)*X'*Y;
```

```
H=X*inv(X‘*X)*X‘;
yh=H*Y;
res=y-x*bh;
ESS=res‘*res;
TSS=ssq(y-y[:]);
RSS=TSS-ESS;
df_ess=n-ncol(x);
df_rss=ncol(x)-1;
df_tss=n-1;
MSE=ESS/df_ess;
MSR=RSS/df_rss;
R2=RSS/TSS;
aR2=1 - ((1-R2)*((n-1)/(n-k)));
Cov_b=MSE*inv(X‘*X);
se_b=sqrt(vecdiag(Cov_b));
t_values=bh/se_b;
t_int= abs(round(t_values));
p_t= 2*(1-probt(abs(t_values),df_ess));
print n k ESS , R2 aR2 , bh se_b p_t t_values t_int;

Corr=corr(x);
nm={"int" "Length3" "H(L3/100)" "w(L3/100)"};
print Corr[colname=nm rowname=nm];

Fit=(n/2)*(log2(ESS)/n);
print Fit;

do i=1 to nrow(t_int);
if t_int[i,] <=1 then L_var=2;
else L_var=2+log2(t_int[i,])+2*log2(log2(t_int[i,]));
*print L_var;

L_var_comb = L_var_comb // L_var;
```

```
end;

print L_var_comb;

Complexity=L_var_comb[+];
print Complexity;

Description_length=Fit+Complexity;
print Description_length;

create yhat from yh[colname="yh"];
append from yh;
create yn from ynew[colname="ynew"];
append from ynew;
quit;

data work.fish;
merge work.fish yhat yn;
run;

proc sort data =work.fish;
by ynew;
run;

goptions reset=all;
axis1 label=(angle=90 'Observed Weight');
axis2 label=('Predicted Weight');
legend1 label=('Species:')
value=('Bream' 'Parkki' 'Perch' 'Pike' 'Roach' 'Smelt' 'Whitefish');
symbol1 color=black value=plus;
symbol2 color=red value=x;
symbol3 color=blue value=star;
symbol4 color=green value=square;
symbol5 color=orange value=triangle;
```

```
symbol6 color=purple value=hash;
symbol7 color=brown value=dot;
title1 'Plot of Observed Weight against Predicted Weight';

proc gplot data=fish;
plot ynew *yh =Species / vzero hzero legend=legend1 vaxis=axis1 haxis=axis2;
run;
```

A comparison between the Kumaraswamy and beta generators

Matthias Wagener 13042450

WST795 Research Report

Submitted in partial fulfilment of the degree BCom(Hons) Mathematical Statistics

Supervisor: Mrs S. L. Makgai, Co-supervisors: Dr I.J.H. Visagie and Prof A. Bekker

Department of Statistics, University of Pretoria

29 September 2017

30th October 2017

Abstract

In this study, we introduce the Kumaraswamy generalised normal distribution (KGN). Applied to a real-life data set, the KGN distribution outperforms its competing beta generalised normal distribution (BGN) making it a useful contribution to the Kumaraswamy generated family of distributions. To familiarise the reader, the beta generator is discussed alongside the Kumaraswamy generator as a comparison study between the two generators by use of various applications of the beta Weibull, Kumaraswamy Weibull, BGN and KGN distributions to real-life data sets. Explicit expansions for mathematical properties are derived such as the PDF, CDF, moments, moment generating function and hazard function. The methods of estimation include maximum likelihood estimation (MLE). The evaluation of goodness-of-fit is done by Akaike information criterion (AIC), Bayesian information criterion (BIC) and Consistent Akaike information criterion (CAIC).

An excerpt of this report titled, "On the Kumaraswamy-generalised normal distribution" is under revision for the peer-reviewed SASA 2017 conference proceedings.

Declaration

I, *Matthias Wagener,* declare that this essay, submitted in partial fulfilment of the degree *BCom(Hons) Mathematical Statistics,* at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.

_____

Matthias Wagener

_____

Seitebaleng L. Makgai

_____

Dr Jaco I.H. Visagie

_____

Prof Andriette Bekker

_____

Date

Acknowledgements

I acknowledge the financial support from the Center for AI Research, Meraka Institute, CSIR.

Contents

List of Figures

List of Tables

1 Introduction

In a scientific field, certain univariate models are known or developed to fit particular data. As such, every distribution is suited best to the data for which it was created.

It may happen, though, that the most appropriate distribution for a certain application does not yield a good fit. Makgai et al. [17] studied this problem under peak-over-threshold data. In their paper, they found that the best suited Pareto distribution model (the baseline distribution) would not give an adequate fit for a certain river flooding data set. The generating method addresses certain shortfalls of the baseline distribution, such as allowing for heavier tail weights [29].

Generated families of distributions usually have complex expressions and therefore have been made more feasible by the computational and analytical facilities available in modern programming software [29]. The beta-generator approach was pioneered by Eugene et al. [8] while deriving the beta normal distribution. This distribution is intended for modelling symmetric heavy-tailed distributions as well as skewed and bimodal distributions.

Jones [12] proved that the beta type I generated family has its origins in order statistics. To illustrate the intuition behind the effect of the shape parameters offered in a beta generated family, we consider the origin of the beta-generator approach.

Let random variable H with probability density function (PDF) $h(x)$ be the $\alpha' th$ order statistic from a random sample of size β from some baseline distribution G [1]:

$$h(x) = \frac{1}{B(\alpha,\beta)} G(X)^{\alpha-1}(1-G(x))^{\beta-\alpha} g(x). \tag{1}$$

The set of PDFs for the order statistics $\alpha = 1, ..., 30$ of random sample size $\beta = 30$ of $G \sim Normal(0, 1)$ are given in Figure 1. The baseline distribution is skewed positively to negatively with increasing values of α relative to β. Note that the PDF (1) implies $X = G^{-1}(F)$ with $F \sim Beta(\alpha, \beta + 1 - \alpha)$, see (69). By letting $\alpha > 0$ and $\beta > 0$ vary as real-valued numbers we generalise the equation (1) to yield the beta type I generated family.

Here we show the Kumaraswamy generated family has similar origins in order statistics to the beta type I generated family. For a set of β independent random samples of size α from an $Uni(0, 1)$ distribution. Taking the maxima of the samples, the minimum value of the maxima will follow a Kumaraswamy distribution with parameters α and β [13]. The beta type I distribution is related to the Kumaraswamy in the cases of $(\alpha, 1)$ and $(1, \beta)$ where their PDFs are equal as can be seen in equations (3) and (5).

Generalising the PDF of the Kumaraswamy for real-valued numbers of $\alpha > 0$ and $\beta > 0$ we have the motivation for the Kumaraswamy generated family.

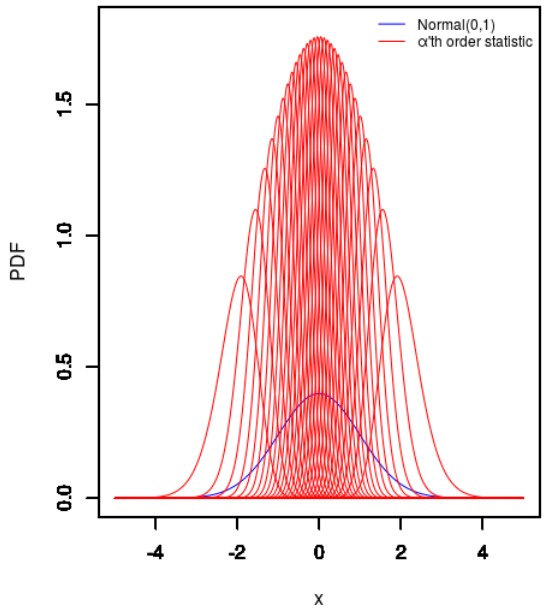


Figure 1: Illustrating the effect of increasing the value of the shape parameter α in the beta type I generated family. The PDFs for the order statistics $\alpha = 1, 2, \ldots, 30$ (from left to right) are from a random sample size $\beta = 30$ from a $Normal(0, 1)$ distribution, analogous to the baseline distribution.

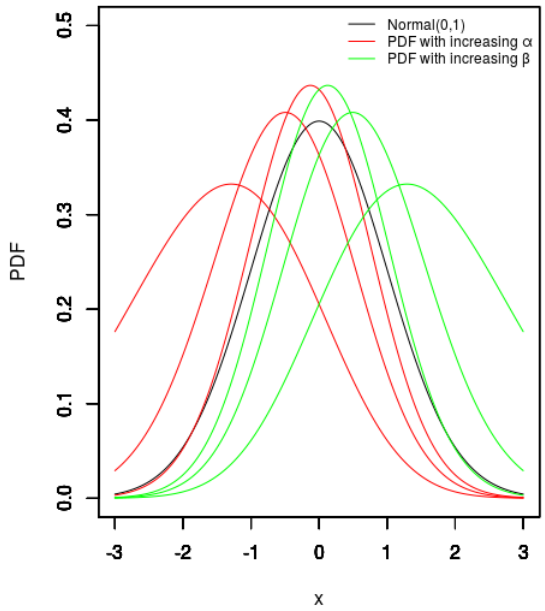


Figure 2: Illustrating the effect of increasing the value of transformation the shape parameters α and β in the Kumaraswamy generated family. The PDFs for increasing values of $\alpha = 0.3, 0.7, 1.1$ with $\beta = 1.3$ (from left to right). The density functions for increasing values of $\beta = 0.3, 0.7, 1.1$ with $\alpha = 1.3$ (from right to left).

Increasing values of α skew the baseline distribution negatively and vice versa for values of β as shown in Figure 2.

The inverse probability integral transformation gives another perspective on how the generator operates. The transformation is given by $X = G^{-1}(F)$ where $F \sim Uni(0, 1)$. By simply replacing F with any distribution, that has a support of $(0, 1)$, we have a generated distribution. In this case we will let $F \sim Beta(\alpha, \beta)$ or $F \sim Kumaraswamy(\alpha, \beta)$ and this relation then inserts the indexing parameters α and β to the baseline distribution as required.

Moving to motivation of the derivation of the Kumaraswamy generalised normal distribution. The shape of the BGN PDF has desirable properties such as high flexibility in location, dispersion, modality and skewness. Applying the BGN model to image pixels from pasture and ocean data, Cintra et al. [3] concluded that the BGN more adequately described the data than classical models for synthetic aperture radar data.

In 2004, Jones [12] recommended the use of symmetric baseline distributions for beta generators such as the generalised normal (GN) distribution. Also in 2009, Jones [13] proposed the Kumaraswamy as a beta-type generator with tractability advantages. This motivates the derivation of the Kumaraswamy generalised normal (KGN) distribution for the first time.

1.1 Methodology

A distribution is said to be generated if it has a cumulative distribution function (CDF) $H(x)$ and PDF $h(x)$ as defined by:

$$\begin{aligned} H(x) &= F(G(x)) \\ &= \int_0^{G(x)} f(w)dw \\ \therefore h(x) &= \frac{d}{dx}F(G(x)), \end{aligned} \tag{2}$$

where $F(\cdot)$ and $f(\cdot)$ are the CDF and PDF the generator distribution and $G(\cdot)$ the CDF the baseline distribution [17].

1.1.1 The beta type family

In this study, two members of the beta family will be investigated as generators, namely the beta type I and Kumaraswamy distributions.

The beta type I PDF (3) and CDF (4) are given below:

$$f(w) = \frac{1}{B(\alpha,\beta)} w^{\alpha-1}(1-w)^{\beta-1}, 0 \leq w \leq 1, \alpha > 0, \beta > 0 \tag{3}$$

and

$$F(w) = \frac{1}{B(\alpha,\beta)} \int_0^w t^{\alpha-1}(1-t)^{\beta-1} dt \tag{4}$$

where $B(\cdot,\cdot)$ is the beta function *(Appendix eq.*(61))[14].

The Kumaraswamy PDF (5) and CDF (6) are given below [15]:

$$f(w) = \alpha\beta w^{\alpha-1}(1-w^{\alpha})^{\beta-1}, 0 \leq w \leq 1, \alpha > 0, \beta > 0 \tag{5}$$

and

$$F(w) = 1 - (1-w^{\alpha})^{\beta}. \tag{6}$$

Remark: The PDFs (3) and (5) are equal for the cases of $\alpha = 1$ or $\beta = 1$.

1.1.2 Baseline distributions

The underlying baseline distributions are chosen as the Weibull and generalised normal distributions.

The Weibull distribution as defined by [14] has a PDF $g(x)$ and CDF $G(x)$ are given by:

$$g(x) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} \exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\} \text{ if } x \geq 0 \\ 0 \text{ if } x > 0 \end{cases} \tag{7}$$

and

$$G(x) = \begin{cases} 1 - \exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\} \text{ if } x \geq 0 \\ 0 \text{ if } x > 0 \end{cases} \tag{8}$$
$$0 < x < \infty, k > 0, \lambda > 0$$

The GN distribution as studied by Nadarajah [21], see also Subbotin [28], has a PDF $g(x)$ and CDF $G(x)$ are given by:

$$g(x) = \frac{s}{2\sigma\Gamma\left(1/s\right)} \exp\left\{-\left|\frac{x-\mu}{\sigma}\right|^{s}\right\} \tag{9}$$

and

$$G(x) = \begin{cases} \frac{s\int_{-\infty}^{x} \exp\left\{-\left(\frac{\mu-y}{\sigma}\right)\right\} dy}{2\sigma\Gamma\left(1/s\right)} \text{if } x \leq \mu \\ 1 - \frac{s\int_{x}^{\infty} \exp\left\{-\left(\frac{y-\mu}{\sigma}\right)^{s}\right\} dy}{2\sigma\Gamma\left(1/s\right)} \text{if } x > \mu \end{cases} \tag{10}$$

$$= \begin{cases} \frac{\Gamma\left(1/s, \left(\frac{\mu-x}{\sigma}\right)^{s}\right)}{2\Gamma\left(1/s\right)} \text{if } x \leq \mu \\ 1 - \frac{\Gamma\left(1/s, \left(\frac{x-\mu}{\sigma}\right)^{s}\right)}{2\Gamma\left(1/s\right)} \text{if } x > \mu \end{cases} \tag{11}$$

$$-\infty < x < \infty, -\infty < \mu < \infty, s > 0, \sigma > 0,$$

where the location, dispersion, and shape is given by parameters μ, σ, and s. Also note that $\Gamma(\cdot)$ and $\Gamma(\cdot,\cdot)$ are the complete and upper incomplete gamma functions *(Appendix eq.*(57) & (60)).

The GN PDF includes the Laplace distribution PDF for $s = 1$ and the normal distribution PDF for $s = 2$ as illustrated in Figure 3.

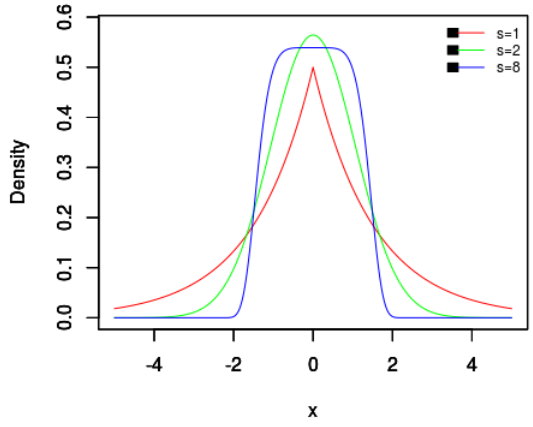


Figure 3: The PDF of a generalised normal distribution with parameters $\mu = 0$, $\sigma = 1$, and $s = \{1, 2, 8\}$. Note that the GN PDF resembles the Laplace PDF for $s = 1$ the and the normal PDF for $s = 2$.

1.2 Literature review

The generator and beta-generator approach has led to an extensive body of research literature. We note the Marshall-Olkin family of generated distributions by Marshall and Olkin [18]; the beta generated distributions by Eugene et al. [8]; gamma generated distributions by Zografos and Balakrishnan [32]; Kumaraswamy generated distributions by Cordeiro and Castro [7]; modified beta generated distributions by Nadarajah et al. [24]; finally, the beta Burr type X distribution by Merovci et al. [19]. For a detailed list of Kumaraswamy and beta generated distributions see the Appendix Tables 7, 8 & 9.

1.3 Report structure

The study is organised as follows.

In Sections 2 and 3, the beta type I and Kumaraswamy distributions will be considered as generators. Using the beta-generator approach, each generator will be coupled with the Weibull distribution as baseline. Properties such as the PDF, CDF, moments, and hazard functions of the beta type I Weibull (BW) and Kumaraswamy Weibull (KW) are revisited and studied.

In Sections 4 and 5, the beta type I and Kumaraswamy distributions will be coupled with the GN distribution. This will be the first coupling of the Kumaraswamy and generalised normal distribution. Properties such as the PDF, CDF, moments, and hazard functions of the beta type I generalised normal (BGN) and Kumaraswamy generalised normal (KGN) are derived and studied.

In Section 6, an application of the two generator distributions is done to compare and evaluate performance of each. In the application, the distributions are fitted using MLE and the performance is measured by AIC, BIC and CAIC.

In Section 7, the study ends with some final concluding remarks and recommendations for future research.

2 The beta type I Weibull Distribution

The beta type I Weibull (BW) distribution was introduced by Lee et al. [16] using the beta generator approach. The derivations of the PDF, CDF, hazard function, moments, moment generating function and likelihood function are included in this Section.

2.1 Introduction

The PDF $h(x)$ and CDF $H(x)$ is obtained using definition (2), baseline CDF (8) and generator CDF (4) [5]:

$$
\begin{aligned}
h(x) &= \frac{1}{B(\alpha,\beta)} G(x)^{\alpha-1}(1-G(x))^{\beta-1} g(x) \qquad (12)\\
&= \frac{1}{B(\alpha,\beta)} \left[1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right]^{\alpha-1} \left[1-\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right)\right]^{\beta-1}\\
&\quad \cdot\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} \exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\\
&= \frac{1}{B(\alpha,\beta)} \frac{k}{\lambda^k} x^{k-1} \exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\} \left[1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right]^{\alpha-1}\\
&\quad \cdot\exp\left\{-(\beta-1)\left(\frac{x}{\lambda}\right)^k\right\}\\
&= \frac{1}{B(\alpha,\beta)} \frac{k}{\lambda^k} x^{k-1} \left[1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right]^{\alpha-1} \exp\left\{-\beta\left(\frac{x}{\lambda}\right)^k\right\} \qquad (13)
\end{aligned}
$$

and

$$
H(x) = \frac{1}{B(\alpha,\beta)} \int_0^{G(x)} t^{\alpha-1}(1-t)^{\beta-1} dt, 0 < x < \infty, k, \lambda, \alpha, \beta > 0. \qquad (14)
$$

This random variable will be denoted as $X \sim BW(k, \lambda, \alpha, \beta)$.

In Figure 4 the BW PDF is shown for various combinations of its parameters demonstrating the flexibility of the PDF shapes achievable. The Weibull baseline distribution, with $\alpha = 1$ and $\beta = 1$, is shown as reference (dashed).

2.2 Hazard function

The hazard rate function of the BW distribution is derived from the definition (*Appendix eq.*(56)), (13), (14)) and (*Appendix eq.*(65)) [5]:

$$
\begin{aligned}
\tau(x) &= \frac{h(x)}{1-H(x)}\\
&= \frac{\frac{1}{B(\alpha,\beta)} \frac{k}{\lambda^k} x^{k-1} \left[1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right]^{\alpha-1} \exp\left\{-\beta\left(\frac{x}{\lambda}\right)^k\right\}}{1-\frac{1}{B(\alpha,\beta)} \int_0^{1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}} t^{\alpha-1}(1-t)^{\beta-1} dt}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\frac{1}{B(\alpha,\beta)}\frac{k}{\lambda^k}x^{k-1}\left[1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right]^{\alpha-1}\exp\left\{-\beta\left(\frac{x}{\lambda}\right)^k\right\}}{1-I_{1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}}(\alpha,\beta)} \\
&= \frac{\frac{1}{B(\alpha,\beta)}\frac{k}{\lambda^k}x^{k-1}\left[1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right]^{\alpha-1}\exp\left\{-\beta\left(\frac{x}{\lambda}\right)^k\right\}}{1-\left(1-I_{\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}}(\beta,\alpha)\right)} \\
&= \frac{\frac{k}{\lambda^k}x^{k-1}\left[1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right]^{\alpha-1}\exp\left\{-\beta\left(\frac{x}{\lambda}\right)^k\right\}}{B(\alpha,\beta)I_{\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}}(\beta,\alpha)},
\end{aligned}
$$

where $I.(\cdot,\cdot)$ is the incomplete beta function ratio (*Appendix eq.*(64)).

In Figure 4 the BW hazard function is shown for various combinations of its parameters. The Weibull baseline distribution hazard function is shown as reference (dashed).

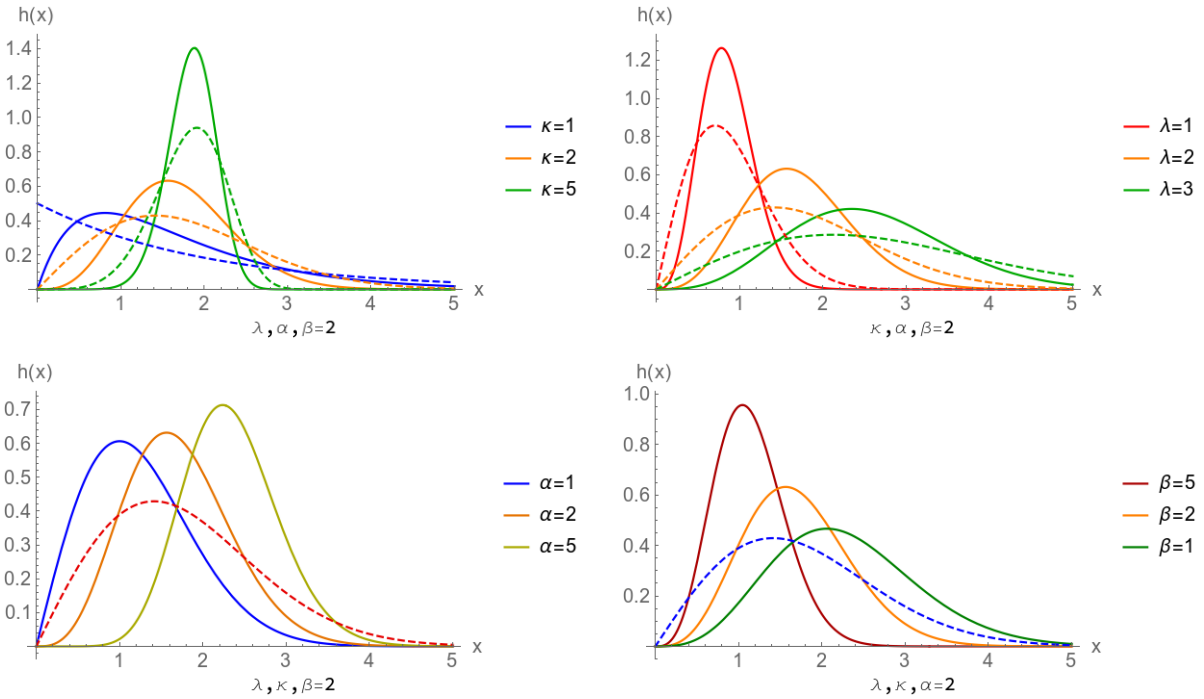


Figure 4: The effect of parameters $k, \lambda, \alpha, \beta$ on the BW PDF. The Weibull baseline distribution, without the influence of the α and β, is shown (dashed) for comparison.

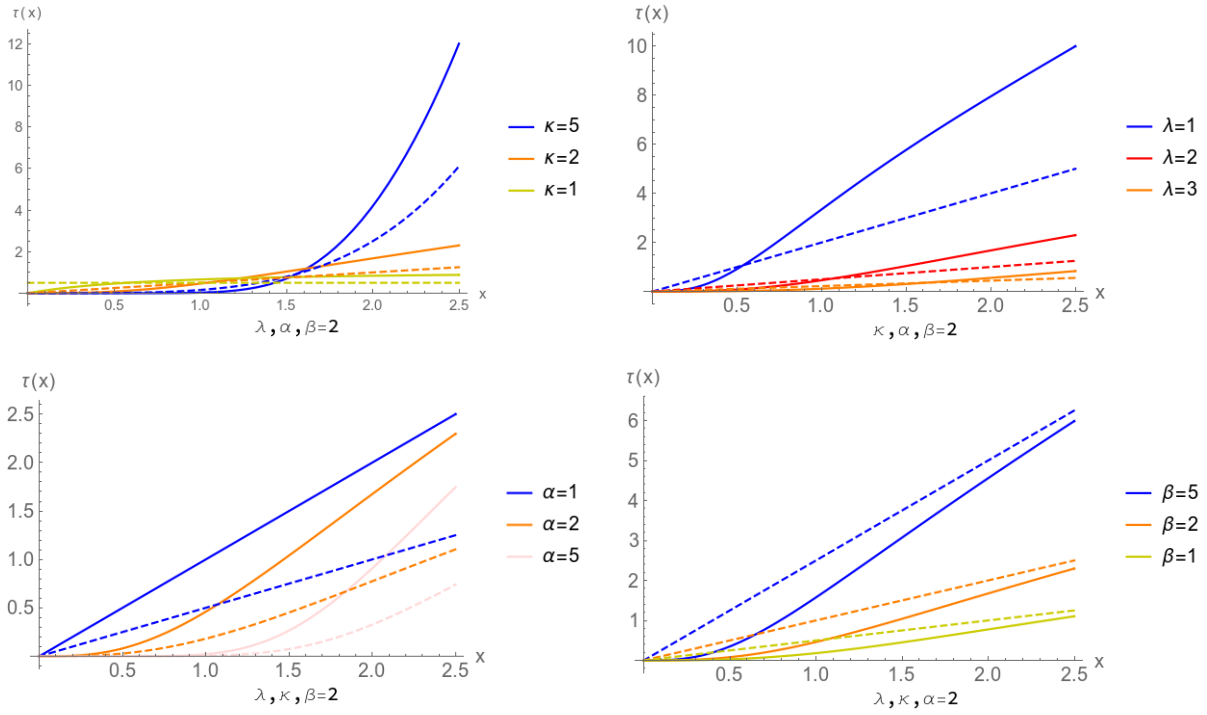


Figure 5: The effect of parameters $k, \lambda, \alpha, \beta$ on the BW hazard function (solid). The Weibull baseline distribution, without the influence of the α and β, is shown as reference (dashed).

2.3 Infinite expansion of the probability density function

The infinite expansion of the BW distribution CDF is used to derive an infinite expansion of the BW PDF.

Firstly, and important integral is expanded with (*Appendix eq.*(82)) and (*Appendix eq.*(68)) [5]:

$$\begin{aligned}
\int_0^x t^{\alpha-1}(1-t)^{\beta-1}dt &= \int_0^x (1-(1-t))^{\alpha-1}(1-t)^{\beta-1}dt \\
&= \int_0^x \sum_{i=0}^{\infty} \binom{\alpha-1}{i} (-(1-t))^i (1-t)^{\beta-1}dt \\
&= \sum_{i=0}^{\infty} \binom{\alpha-1}{i} (-1)^i \int_0^x (1-t)^{\beta+i-1}dt \\
&= \sum_{i=0}^{\infty} \frac{\binom{\alpha-1}{i}(-1)^i}{\beta+i} \left(1-(1-x)^{\beta+i}\right) \\
&= \sum_{i=0}^{\infty} \frac{\Gamma(\alpha)(-1)^i}{\Gamma(a-i)i!(\beta+i)} \left(1-(1-x)^{\beta+i}\right), 0<x<1, 0<\alpha, 0<\beta. \qquad (15)
\end{aligned}$$

Note that the upper bound of the sum is $\alpha - 1$ for integer values of α, see (*Appendix eq.*(83)).

Substituting (15) and (8) into (14), we have that:

$$
\begin{aligned}
H(x) &= \frac{1}{B(\alpha,\beta)}\sum_{i=0}^{\infty}\frac{\Gamma(a)\,(-1)^{i}}{\Gamma(a-i)i!(\beta+i)}(1-(1-G(x))^{\beta+i}) \\
&= \sum_{i=0}^{\infty} w_i(\alpha,\beta)\left(1-\exp\left\{-(\beta+i)\left(\frac{x}{\lambda}\right)^{k}\right\}\right) \\
&= \sum_{i=0}^{\infty} w_i(\alpha,\beta)\left(1-\exp\left\{-\left((\beta+i)^{1/k}\frac{x}{\lambda}\right)^{k}\right\}\right) \\
&= \sum_{i=0}^{\infty} w_i(\alpha,\beta)G_{\lambda_i,k}(x) \\
h(x) &= \frac{d}{dx}H(x) \\
&= \sum_{i=0}^{\infty} w_i(\alpha,\beta)g_{\lambda_i,k}(x), \qquad (16)
\end{aligned}
$$

where

$$
\begin{aligned}
w_i(\alpha,\beta) &= \frac{\Gamma(\alpha)\,(-1)^{i}}{\Gamma(\alpha-i)}\frac{1}{B(\alpha,\beta)(\beta+i)i!} \\
&= \frac{(1-\alpha)_i}{B(\alpha,\beta)(\beta+i)i!} \qquad (17)
\end{aligned}
$$

and $(\alpha)_i$ is the Pochhammer function (*Appendix eq.*(66) & (67)).

The CDF (14) is now given by an infinite weighted sum of Weibull CDFs $G_{\lambda_i,k}(x)$ with parameter $\lambda_i = \lambda/(\beta+i)^{1/k}$ (*see eq.*(8)) and weights $w_i(\alpha,\beta)$. The PDF (13) is similarly given by Weibull PDFs $g_{\lambda_i,k}(x)$ (*see eq.*(7)).

2.4 Moments

The r′th moment of the BW is derived using the infinite weighted sum of Weibull PDFs with parameters $\lambda_i = \lambda/(\beta+i)^{1/k}$ and k (16).

From the definition of a moment, (16) and (17), it follows that the moments of the BW is [5]:

$$
\begin{aligned}
E\left(X^{r}\right) &= \int_0^{\infty} x^{r}h(x)dx \\
&= \int_0^{\infty} x^{r}\sum_{i=0}^{\infty} w_i(\alpha,\beta)g_{\lambda_i,k}(x)dx \\
&= \sum_{i=0}^{\infty} w_i(\alpha,\beta)\int_0^{\infty} x^{r}g_{\lambda_i,k}(x)dx.
\end{aligned}
$$

From the moment $\mu'_{i,r} = \int_0^{\infty} x^{r}g_{\lambda_i,k}(x)dx = \Gamma\left(r/k+1\right)\lambda_i^{r}$ see (*Appendix eq.*(74)):

$$
\begin{aligned}
E\left(X^r\right) &= \sum_{i=0}^{\infty} w_i(\alpha,\beta)\int_0^{\infty} x^r g_{\lambda_i,k}(x)dx \\
&= \sum_{i=0}^{\infty} w_i(\alpha,\beta)\Gamma\left(\frac{r}{k}+1\right)\lambda_i^r \\
&= \Gamma\left(\frac{r}{k}+1\right)\sum_{i=0}^{\infty} w_i(\alpha,\beta)\lambda_i^r \\
&= \Gamma\left(\frac{r}{k}+1\right)\sum_{i=0}^{\infty} w_i(\alpha,\beta)\left(\frac{\lambda}{(\beta+i)^{1/k}}\right)^r \\
&= \Gamma\left(\frac{r}{k}+1\right)\lambda^r\sum_{i=0}^{\infty} w_i(\alpha,\beta)(\beta+i)^{-\frac{r}{k}} \\
&= \frac{\Gamma\left(\frac{r}{k}+1\right)}{B(\alpha,\beta)}\lambda^r\sum_{i=0}^{\infty}\frac{(1-\alpha)_i}{i!(\beta+i)^{1+r/k}}.
\end{aligned}
$$

Note that the upper bound of the sum is $\alpha-1$ for integer values of α, see (*Appendix eq.*(83)).

2.5 Moment generating function

From the definition of a moment generating function, the infinite expansion of the PDF (13), (*Appendix eq.*(82)), (*Appendix eq.*(68)) and (*Appendix eq.*(67)). The BW moment generating function is [5]:

$$
\begin{aligned}
M(t) &= E[\exp\{tX\}] \\
&= \int_0^{\infty}\exp\{tx\}\frac{1}{B(\alpha,\beta)}\frac{k}{\lambda^k}x^{k-1}\left[1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right]^{\alpha-1}\exp\left\{-\beta\left(\frac{x}{\lambda}\right)^k\right\}dx \\
&= \frac{k}{\lambda^k B(\alpha,\beta)}\int_0^{\infty} x^{k-1}\exp\left\{tx-\beta\left(\frac{x}{\lambda}\right)^k\right\}\left[1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right]^{\alpha-1}dx \\
&= \frac{k}{\lambda^k B(\alpha,\beta)}\int_0^{\infty} x^{k-1}\exp\left\{tx-\beta\left(\frac{x}{\lambda}\right)^k\right\}\sum_{i=0}^{\infty}\binom{\alpha-1}{i}\left[-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right]^i dx \\
&= \frac{k}{\lambda^k B(\alpha,\beta)}\sum_{i=0}^{\infty}\binom{\alpha-1}{i}(-1)^i\int_0^{\infty} x^{k-1}\exp\left\{tx-\beta\left(\frac{x}{\lambda}\right)^k\right\}\exp\left\{-i\left(\frac{x}{\lambda}\right)^k\right\}dx \\
&= \frac{k}{\lambda^k B(\alpha,\beta)}\sum_{i=0}^{\infty}\frac{\Gamma(a)(-1)^i}{\Gamma(a-i)i!}\int_0^{\infty} x^{k-1}\exp\left\{tx-(\beta+i)\left(\frac{x}{\lambda}\right)^k\right\}dx \\
&= \frac{k}{\lambda^k B(\alpha,\beta)}\sum_{i=0}^{\infty}\frac{(1-\alpha)_i}{i!}\int_0^{\infty} x^{k-1}\exp\left\{tx-\left(\frac{(i+\beta)^{1/k}x}{\lambda}\right)^k\right\}dx, \qquad (18)
\end{aligned}
$$

Note that the upper bound of the sum is $\alpha-1$ for integer values of α, see (*Appendix eq.*(83)).

Remark:

For an alternative expression of the moment generating function, consider the transformation $y=x^k$, which implies $x=y^{\frac{1}{k}}$ and $\frac{dx}{dy}=\frac{1}{k}y^{\frac{1}{k}-1}$ on the integral in (18):

$$\begin{aligned}\int_0^\infty x^{k-1}\exp\left\{tx-\left(\frac{(i+\beta)^{1/k}x}{\lambda}\right)^k\right\}dx &= \frac{1}{k}\int_0^\infty y^{\frac{1}{k}-1}y^{\frac{k-1}{k}}\exp\left\{tx-\left(\frac{(i+\beta)^{1/k}y^{\frac{1}{k}}}{\lambda}\right)^k\right\}dy\\ &= \frac{1}{k}\int_0^\infty \exp\left\{-\frac{(i+\beta)}{\lambda^k}y\right\}\exp\left\{ty^{\frac{1}{k}}\right\}dy\\ &= \frac{1}{k}E_Y\left[\exp\left\{ty^{\frac{1}{k}}\right\}\right],\end{aligned}$$

where $Y \sim Exp\left(\frac{i+\beta}{\lambda^k}\right)$ distributed, see (71). Therefore,

$$M(t) = \frac{1}{\lambda^k B(\alpha,\beta)}\sum_{i=0}^{\infty}\frac{(1-\alpha)_i}{i!}E_Y\left[\exp\left\{ty^{\frac{1}{k}}\right\}\right].$$

2.6 Likelihood function

The likelihood function follows from (13),

$$\begin{aligned}L(x;\lambda,\alpha,\beta) &= \prod_{i=1}^{n}h(x_i)\\ &= \prod_{i=1}^{n}\frac{1}{B(\alpha,\beta)}\frac{k}{\lambda^k}x_i^{k-1}\exp\left\{-\left(\frac{x_i}{\lambda}\right)^k\right\}\left[1-\exp\left\{-\left(\frac{x_i}{\lambda}\right)^k\right\}\right]^{\alpha-1}\\ &\quad\cdot\exp\left\{-(\beta-1)\left(\frac{x_i}{\lambda}\right)^k\right\}.\end{aligned} \tag{19}$$

Due to the complexity of the solutions to the maximum likelihood estimators (MLE) in (19) are not derived analytically. This function is however used to numerically obtain the MLE estimates in Section 6 for estimates of λ, α, β.

3 The Kumaraswamy Weibull Distribution

The Kumaraswamy Weibull (KW) distribution was introduced by Cordeiro et al. [6] using the beta generator approach. The derivations of the PDF, CDF, hazard function, moments, moment generating function and likelihood function are included in this Section.

3.1 Introduction

The PDF $h(x)$ and CDF $H(x)$ is obtained using definition (2), baseline CDF (8) and generator CDF (6) [6]:

$$
\begin{aligned}
h(x) &= \alpha\beta G(x)^{\alpha-1}(1-G(x)^{\alpha})^{\beta-1}g(x) && (20)\\
&= \alpha\beta\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\right)^{\alpha-1}\left(1-\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\right)^{\alpha}\right)^{\beta-1}\\
&\quad \cdot\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1}\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\\
&= \alpha\beta\frac{k}{\lambda^{k}}x^{k-1}\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\right)^{\alpha-1}\\
&\quad \cdot\left(1-\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\right)^{\alpha}\right)^{\beta-1} && (21)
\end{aligned}
$$

and

$$
\begin{aligned}
H(x) &= 1-(1-G(x)^{\alpha})^{\beta} && (22)\\
&= 1-\left(1-\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\right)^{\alpha}\right)^{\beta}, 0<x<\infty, k, \lambda, \alpha, \beta>0. && (23)
\end{aligned}
$$

This random variable will be denoted as $X \sim KW(k, \lambda, \alpha, \beta)$.

In Figure 6 the KW PDF (dashed) is compared to the BW PDF (shaded from axis). The KW PDF displays the same flexibility as the BW PDF but with lower and higher peaks in the PDF for small and large values of α.

3.2 Hazard function

The hazard rate function of the KW distribution is derived from the definition (*Appendix eq.*(56)), (21) and (23) [6]:

$$
\begin{aligned}
\tau(x) &= \frac{\alpha\beta\frac{k}{\lambda^{k}}x^{k-1}\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\right)^{\alpha-1}\left(1-\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\right)^{\alpha}\right)^{\beta-1}}{\left(1-\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\right)^{\alpha}\right)^{\beta}}\\
&= \frac{\alpha\beta\frac{k}{\lambda^{k}}x^{k-1}\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\right)^{\alpha-1}}{1-\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}\right)^{\alpha}}.
\end{aligned}
$$

In Figure 7 the KW hazard function (dashed) is compared to the BW hazard function (shaded from axis). The differences between the KW hazard function (dashed) and the BW hazard function (shaded from axis) is more pronouced for changes in α and β. This is due to the functional difference of the α parameter in the generators.

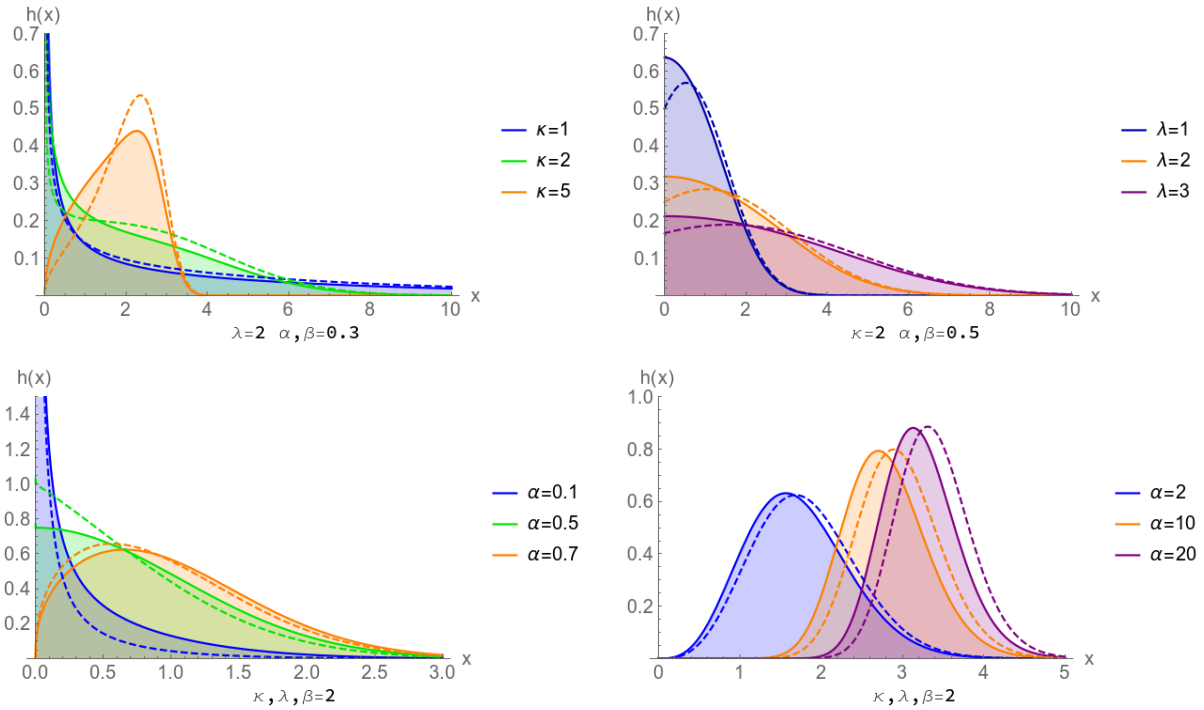


Figure 6: The effect of the parameters on KW PDF (dashed) compared to the BW PDF (shaded from the axis).

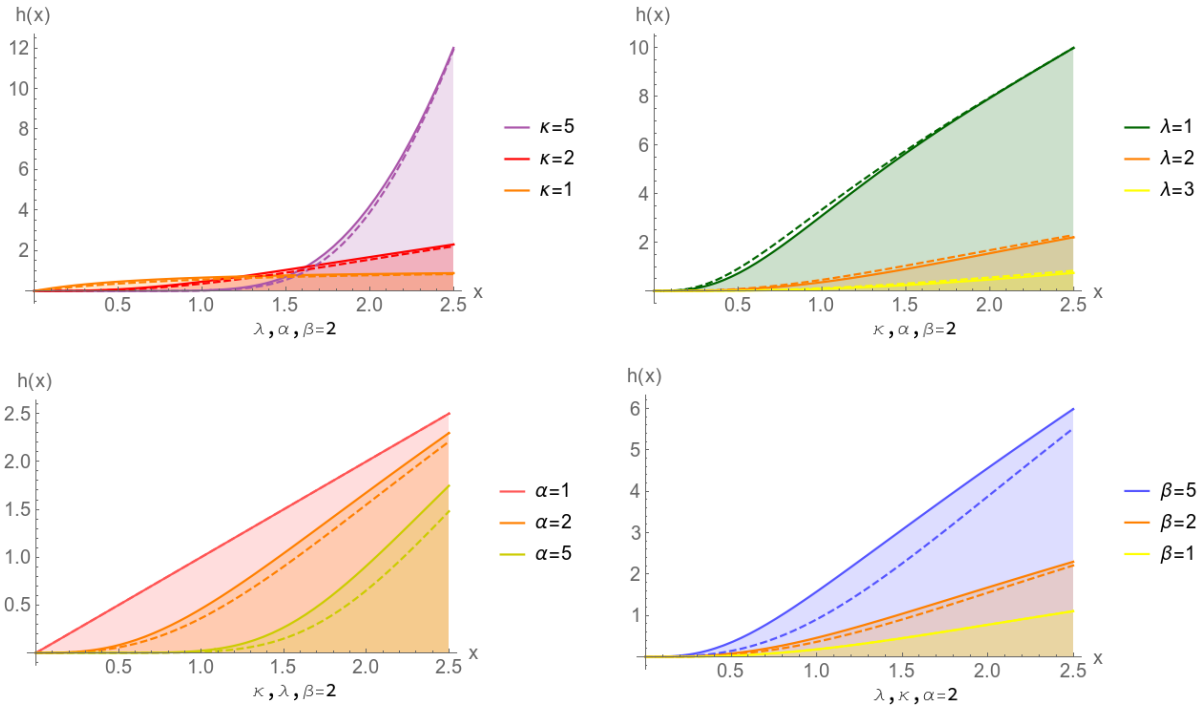


Figure 7: The effect of the parameters on KW hazard function (dashed) compared to the BW hazard function (shaded from the axis). Note the BGN and KGN hazard functions are mathematically equal for $\alpha = 1$ (lower left corner).

3.3 Infinite expansion of the probability density function

From the result (*Appendix eq.*(82)) and (*Appendix eq.*(68)), (20) is expanded [6]:

$$\begin{aligned} h(x) &= \alpha\beta g(x)G(x)^{\alpha-1}(1-G(x)^{\alpha})^{\beta-1} \\ &= \alpha\beta g(x)G(x)^{\alpha-1}\sum_{i=0}^{\infty}\binom{\beta-1}{i}(-G(x)^{\alpha})^{i} \\ &= \sum_{i=0}^{\infty}\frac{\Gamma(\beta)\,(-1)^{i}}{\Gamma(\beta-i)i!}\alpha\beta g(x)G(x)^{\alpha(i+1)-1} \\ &= \sum_{i=0}^{\infty}\beta\frac{\Gamma(\beta)\,(-1)^{i}}{\Gamma(\beta-i)(i+1)i!}\alpha(i+1)g(x)G(x)^{\alpha(i+1)-1} \\ &= \sum_{i=0}^{\infty}w_i(\beta)\alpha(i+1)g(x)G(x)^{\alpha(i+1)-1}, \end{aligned} \tag{24}$$

where

$$\begin{aligned} w_i(\beta) &= \beta\frac{\Gamma(\beta)\,(-1)^{i}}{\Gamma(\beta-i)}\frac{1}{(i+1)i!} \\ &= \beta\frac{(1-\beta)_i}{(i+1)i!}, 0<x<\infty, 0<k, 0<\lambda, 0<\alpha, 0<\beta, \end{aligned} \tag{25}$$

see (67)). Note that the upper bound of the sum is $\beta-1$ for integer values of β, see (*Appendix eq.*(83)).

From (*Appendix eq.*(75)) the PDF (20) is now given by an infinite weighted sum of exponentiated Weibull PDFs $g_{v_i,\lambda,k}(x)$ with parameter $v_i=\alpha(i+1)$ and weights $w_i(\beta)$.

3.4 Moments

The r′th moment of the KW is derived using the infinite weighted sum of exponentiated Weibull PDFs with parameters λ, k and $v_i=\alpha(i+1)$ (24).

From the definition of a moment, (24) and (25), it follows that the moments of the KW is [6]:

$$\begin{aligned} E\left(X^{r}\right) &= \int_0^{\infty}x^{r}\sum_{i=0}^{\infty}w_i(\beta)g_{v_i,\lambda,k}(x)dx \\ &= \sum_{i=0}^{\infty}w_i(\beta)\int_0^{\infty}x^{r}g_{v_i,\lambda,k}(x)dx. \end{aligned}$$

From the moment $\mu'_{i,r}=v_i\lambda^{r}\Gamma(r/k)\sum_{j=0}^{\infty}\frac{(1-v_i)_j}{j!(j+1)^{(r+k)/k}}$ see (*Appendix eq.*(77)) and v_i defined above. Note

that the upper bound of the sums are $\beta - 1$ and $v_i - 1$ for integer values of β and v_i, see (*Appendix eq.*(83)).

3.5 Moment generating function

From the definition of a moment generating function, the infinite expansion of the PDF (21), (*Appendix eq.*(82)), (*Appendix eq.*(68)) and (*Appendix eq.*(67)). The KW moment generating function is [6]:

$$\begin{aligned}
M(t) &= E[\exp\{tX\}] \\
&= \int_0^\infty \exp\{tx\}\,\alpha\beta\frac{k}{\lambda^k}x^{k-1}\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right)^{\alpha-1} \\
&\quad\cdot\left(1-\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right)^\alpha\right)^{\beta-1}dx \\
&= \alpha\beta\frac{k}{\lambda^k}\int_0^\infty x^{k-1}\exp\left\{tx-\left(\frac{x}{\lambda}\right)^k\right\}\sum_{i=0}^\infty\binom{\alpha-1}{i}\left(-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right)^i \\
&\quad\cdot\sum_{j=0}^\infty\binom{\beta-1}{j}^j\left(-\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right)^\alpha\right)^j dx \\
&= \alpha\beta\frac{k}{\lambda^k}\int_0^\infty x^{k-1}\sum_{i=0}^\infty\binom{\alpha-1}{i}(-1)^i\exp\left\{tx-\left(\frac{x}{\lambda}\right)^k\right\}\exp\left\{-i\left(\frac{x}{\lambda}\right)^k\right\} \\
&\quad\cdot\sum_{j=0}^\infty\binom{\beta-1}{j}^j(-1)^j\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right)^{\alpha j}dx \\
&= \alpha\beta\frac{k}{\lambda^k}\sum_{i=0}^\infty\sum_{j=0}^\infty\binom{\alpha-1}{i}(-1)^i\binom{\beta-1}{j}^j(-1)^j\int_0^\infty x^{k-1}\exp\left\{tx-(i+1)\left(\frac{x}{\lambda}\right)^k\right\} \\
&\quad\cdot\left(\sum_{l=0}^\infty\binom{\alpha j}{l}\left(-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right)^l\right)dx \\
&= \alpha\beta\frac{k}{\lambda^k}\sum_{i=0}^\infty\sum_{j=0}^\infty\frac{\Gamma(\alpha)(-1)^i}{\Gamma(\alpha-i)i!}\frac{\Gamma(\beta)(-1)^j}{\Gamma(\beta-j)j!}\int_0^\infty x^{k-1}\exp\left\{tx-(i+1)\left(\frac{x}{\lambda}\right)^k\right\} \\
&\quad\cdot\left(\sum_{l=0}^\infty\binom{\alpha j}{l}(-1)^l\exp\left\{-l\left(\frac{x}{\lambda}\right)^k\right\}\right)dx \\
&= \alpha\beta\frac{k}{\lambda^k}\sum_{i=0}^\infty\sum_{j=0}^\infty\sum_{l=0}^\infty\frac{(1-\alpha)_i}{i!}\frac{(1-\beta)_j}{j!}\binom{\alpha j}{l}(-1)^l\int_0^\infty x^{k-1}\exp\left\{tx-(i+1)\left(\frac{x}{\lambda}\right)^k\right\} \\
&\quad\cdot\exp\left\{-l\left(\frac{x}{\lambda}\right)^k\right\}dx \\
&= \alpha\beta\frac{k}{\lambda^k}\sum_{i=0}^\infty\sum_{j=0}^\infty\sum_{l=0}^\infty\frac{(1-\alpha)_i}{i!}\frac{(1-\beta)_j}{j!}\frac{(1-\alpha j)_i}{i!}\int_0^\infty x^{k-1}\exp\left\{tx-(i+l+1)\left(\frac{x}{\lambda}\right)^k\right\}dx \\
&= \alpha\beta\frac{k}{\lambda^k}\sum_{i=0}^\infty\sum_{j=0}^\infty\sum_{l=0}^\infty w_{i,j}(\alpha,\beta)\int_0^\infty x^{k-1}\exp\left\{tx-\left((i+l+1)^{1/k}\frac{x}{\lambda}\right)^k\right\}dx, \qquad (26)
\end{aligned}$$

where

$$w_{i,j}(\alpha,\beta) = \frac{(1-\alpha)_i}{i!}\frac{(1-\beta)_j}{j!}\frac{(1-\alpha j)_i}{i!}.$$

Note that the upper bound of the sums are $\alpha - 1$ and $\beta - 1$ for integer values of α and β, see (*Appendix eq.*(83)).

Remark:

For and alternative expression of the moment generating function, consider the transformation $y = x^k$, which implies $x = y^{\frac{1}{k}}$ and $\frac{dx}{dy} = \frac{1}{k} y^{\frac{1}{k}-1}$ on the integral in (26):

$$\begin{aligned}
\int_0^\infty x^{k-1} \exp\left\{tx - (i+l+1)\left(\frac{x}{\lambda}\right)^k\right\} dx &= \frac{1}{k}\int_0^\infty y^{\frac{1}{k}-1} y^{\frac{k-1}{k}} \exp\left\{tx - (i+l+1)\left(\frac{y^{\frac{1}{k}}}{\lambda}\right)^k\right\} dy \\
&= \frac{1}{k}\int_0^\infty \exp\left\{-\frac{i+l+1}{\lambda^k} y\right\} \exp\left\{t y^{\frac{1}{k}}\right\} dy \\
&= \frac{1}{k} E_Y\left[\exp\left\{t y^{\frac{1}{k}}\right\}\right],
\end{aligned}$$

where $Y \sim Exp\left(\frac{i+l+1}{\lambda^k}\right)$ distributed, see (71). Therefore,

$$M(t) = \alpha\beta \frac{1}{\lambda^k} \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\sum_{l=0}^{\infty} w_{i,j}(\alpha,\beta) E_Y\left[\exp\left\{t y^{\frac{1}{k}}\right\}\right].$$

3.6 Likelihood function

The likelihood function follows from (21),

$$\begin{aligned}
L(x;\lambda,\alpha,\beta) &= \prod_{i=1}^{n} h(x_i) \\
&= \prod_{i=1}^{n} \alpha\beta \frac{k}{\lambda^k} x_i^{k-1} \exp\left\{-\left(\frac{x_i}{\lambda}\right)^k\right\} \left(1 - \exp\left\{-\left(\frac{x_i}{\lambda}\right)^k\right\}\right)^{\alpha-1} \\
&\quad \cdot \left(1 - \left(1 - \exp\left\{-\left(\frac{x_i}{\lambda}\right)^k\right\}\right)^{\alpha}\right)^{\beta-1}. \qquad (27)
\end{aligned}$$

Due to the complexity of the solutions to the maximum likelihood estimators (MLE) in (27) are not derived analytically. This function is however used to numerically obtain the MLE estimates in Section 6 for estimates of λ, α, β.

4 The beta type I generalised normal distribution

The beta type I generalised normal (BGN) distribution was introduced by Cintra et al. [3] using the beta generator approach. The derivations of the PDF, CDF, hazard function, moments, moment generating function and likelihood function are included in this Section.

4.1 Introduction

The PDF $h(x)$ and CDF $H(x)$ is obtained using definition (2), baseline CDF (11), generator CDF (4) and (12) [3]:

$$\begin{aligned} h(x) &= \frac{1}{B(\alpha,\beta)}\left[\Phi_s\left(\frac{x-\mu}{\sigma}\right)\right]^{\alpha-1}\left[1-\Phi_s\left(\frac{x-\mu}{\sigma}\right)\right]^{\beta-1} \\ &\quad \cdot\frac{s}{2\sigma\Gamma\left(1/s\right)}\exp\left\{-\left|\frac{x-\mu}{\sigma}\right|^s\right\} \\ &= \frac{1}{\sigma B(\alpha,\beta)}\left[\Phi_s\left(\frac{x-\mu}{\sigma}\right)\right]^{\alpha-1}\left[1-\Phi_s\left(\frac{x-\mu}{\sigma}\right)\right]^{\beta-1}\phi_s\left(\frac{x-\mu}{\sigma}\right) \end{aligned} \tag{28}$$

and

$$H(x)=\frac{1}{B(\alpha,\beta)}\int_0^{\Phi_s\left(\frac{x-\mu}{\sigma}\right)} t^{\alpha-1}(1-t)^{\beta-1}dt, -\infty<x<\infty, -\infty<\mu<\infty, s,\sigma,\alpha,\beta>0, \tag{29}$$

where $\phi_s(\cdot)$ and $\Phi_s(\cdot)$ is defined in (78) and (79). This random variable will be denoted as $X\sim BGN(\mu,\sigma,s,\alpha,\beta)$.

In Figure 8 the PDF of the BGN is shown for various combinations of its parameters. The flexibility of the BGN is shown to fit real data with pronounced skewness and bi-modality. The generalised normal baseline distribution, with $\alpha=1$ and $\beta=1$, is shown as reference (dashed). Note that the BGN is equal to the beta normal (BN) distribution for $s=2$, with the only difference of $\sigma_{BN}=\sqrt{2}\sigma$, and the normal distribution $s=2,\ \alpha,\beta=1$, see (28) and Figure 8.

4.2 Hazard function

The hazard rate function of the BGN distribution is derived from (*Appendix eq.*(56)), (28) and (29) [3]:

$$\tau(x) = \frac{\frac{1}{B(\alpha,\beta)}\left[\Phi_s\left(\frac{x-\mu}{\sigma}\right)\right]^{\alpha-1}\left[1-\Phi_s\left(\frac{x-\mu}{\sigma}\right)\right]^{\beta-1}\frac{s}{2\sigma\Gamma(1/s)}\exp\{-|\frac{x-\mu}{\sigma}|^s\}}{1-\frac{1}{B(\alpha,\beta)}\int_0^{G(x)}t^{\alpha-1}(1-t)^{\beta-1}dt}$$

$$
\begin{aligned}
&= \frac{\frac{1}{\sigma B(\alpha,\beta)}[\Phi_s(z)]^{\alpha-1}[1-\Phi_s(z)]^{\beta-1}\phi_s(z)}{1-\frac{1}{B(\alpha,\beta)}\int_0^{\Phi_s(z)} t^{\alpha-1}(1-t)^{\beta-1}dt} \\
&= \frac{[\Phi_s(z)]^{\alpha-1}[1-\Phi_s(z)]^{\beta-1}\phi_s(z)}{\sigma\left(B(\alpha,\beta)-\int_0^{\Phi_s(z)} t^{\alpha-1}(1-t)^{\beta-1}dt\right)} \\
&= \frac{[\Phi_s(z)]^{\alpha-1}[1-\Phi_s(z)]^{\beta-1}\phi_s(z)}{\sigma\left(B(\alpha,\beta)-I_{\Phi_s(z)}(\alpha,\beta)\right)},
\end{aligned}
$$

where $I.(\cdot,\cdot)$ is the incomplete beta function ratio (*Appendix eq.*(64))). In Figure 9 the hazard function is depicted for various combinations of its parameters. The generalised baseline distribution hazard function is shown as reference (dashed).

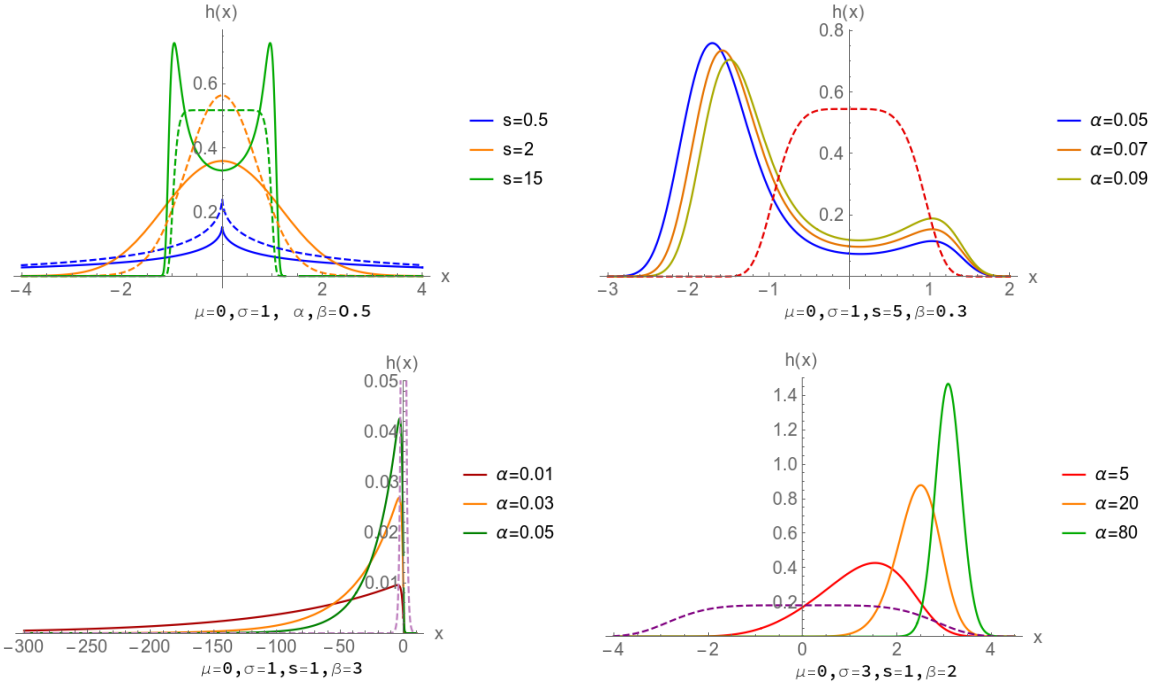


Figure 8: The effect of parameters $\mu, \sigma, s, \alpha, \beta$ on the BGN PDF. The generalised normal baseline distribution is shown as reference (dashed). Note that in the upper left corner we have the BN PDF (solid orange) and the standard normal PDF (dashed orange).

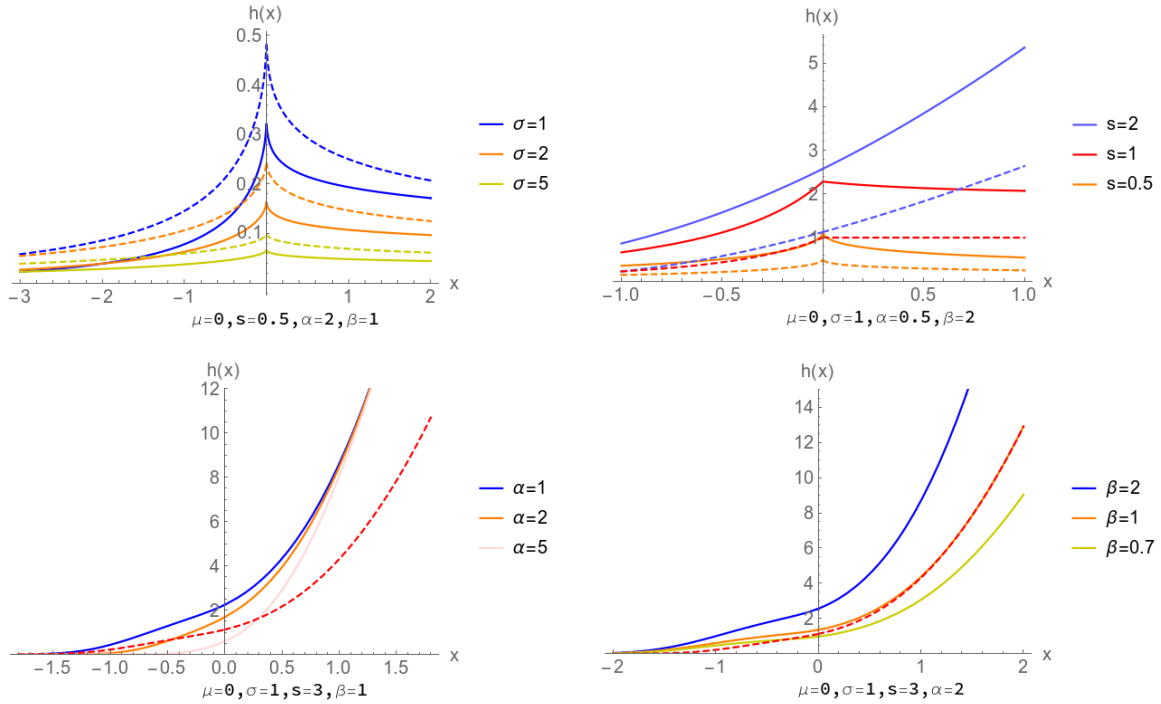


Figure 9: The effect of parameters σ, s, α, β on the BGN hazard function. The generalised normal baseline hazard function is shown as reference (dashed).

4.3 Infinite expansion of the probability density function

From (2), (3), (*Appendix eq.*(82)), (*Appendix eq.*(68)) and defining weights $w_i(\beta)$ it follows that [3]:

$$
\begin{aligned}
h(x) &= \frac{1}{B(\alpha,\beta)}\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha-1}\left(1-\Phi_s\left(\frac{x-\mu}{\sigma}\right)\right)^{\beta-1}\phi_s\left(\frac{x-\mu}{\sigma}\right) \\
&= \frac{1}{B(\alpha,\beta)}\phi_s\left(\frac{x-\mu}{\sigma}\right)\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha-1}\sum_{i=0}^{\infty}\binom{\beta-1}{i}\left(-\Phi_s\left(\frac{x-\mu}{\sigma}\right)\right)^{i} \\
&= \frac{1}{B(\alpha,\beta)}\phi_s\left(\frac{x-\mu}{\sigma}\right)\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha-1}\sum_{i=0}^{\infty}\frac{\Gamma(\beta)\,(-1)^i}{\Gamma(\beta-i)i!}\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{i} \\
&= \frac{1}{\sigma B(\alpha,\beta)}\sum_{i=0}^{\infty}\frac{\Gamma(\beta)\,(-1)^i}{\Gamma(\beta-i)i!}\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{i+\alpha-1}\phi_s\left(\frac{x-\mu}{\sigma}\right) \\
&= \frac{1}{\sigma B(\alpha,\beta)}\sum_{i=0}^{\infty}w_i(\beta)\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{i+\alpha-1}\phi_s\left(\frac{x-\mu}{\sigma}\right), -\infty<x<\infty, 0<\alpha, 0<\beta, \qquad (30)
\end{aligned}
$$

where

$$
w_i(\beta) = \frac{\Gamma(\beta)\,(-1)^i}{\Gamma(\beta-i)i!}
$$

$$= \frac{(1-\beta)_i}{i!} \tag{31}$$

with $\phi_s(\cdot)$ and $\Phi_s(\cdot)$ is defined in ((9)&(11)) also see (67). Note that the upper bound of the sums are $\alpha - 1$ and $\beta - 1$ for integer values of α and β, see (*Appendix eq.*(83)).

4.4 Moments

From the definition of moments, (30) and (31) it follows that [3]:

$$\begin{aligned} E\left(X^r\right) &= \int_{-\infty}^{\infty} x^r \frac{1}{\sigma B(\alpha,\beta)} \sum_{i=0}^{\infty} w_i(\beta) \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{i+\alpha-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx \\ &= \frac{1}{\sigma B(\alpha,\beta)} \sum_{i=0}^{\infty} w_i(\beta) \int_{-\infty}^{\infty} x^r \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{i+\alpha-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx. \end{aligned} \tag{32}$$

Using the transformation $z = {}^{(x-\mu)}/_{\sigma}$, which implies $x = \sigma z + \mu$ and $\frac{dx}{dz} = \sigma$, and (*Appendix eq.*(82)). The integral in (32) specifically becomes:

$$\begin{aligned} \int_{-\infty}^{\infty} x^r \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{i+\alpha-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx &= \sigma \int_{-\infty}^{\infty} (\sigma z + \mu)^r \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz \\ &= \sigma \int_{-\infty}^{\infty} \sum_{k=0}^{r} \binom{r}{k} \mu^{r-k} (\sigma z)^k \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz \\ &= \sigma \mu^r \sum_{k=0}^{r} \binom{r}{k} \sigma^k \mu^{-k} \int_{-\infty}^{\infty} z^k \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz \\ &= \sigma \mu^r \sum_{k=0}^{r} \binom{r}{k} \left(\frac{\sigma}{\mu}\right)^k \int_{-\infty}^{\infty} z^k \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz. \end{aligned} \tag{33}$$

Splitting the range of the integral (33), using (*Appendix eq.*(80)),(*Appendix eq.*(81)) and (*Appendix eq.*(82)) we obtain:

$$\begin{aligned} \int_{-\infty}^{\infty} z^k \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz &= \int_{0}^{\infty} z^k \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz \\ &\quad + \int_{-\infty}^{0} z^k \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz \end{aligned}$$

$$
\begin{aligned}
&= \int_0^\infty z^k \Phi_s(z)^{i+\alpha-1}\phi_s(z)dz \\
&\quad + \int_0^\infty (-z)^k (1-\Phi_s(z))^{i+\alpha-1}\phi_s(z)dz \\
&= \int_0^\infty z^k \Phi_s(z)^{i+\alpha-1}\phi_s(z)dz \\
&\quad + (-1)^k \int_0^\infty z^k \sum_{l=0}^{\infty} \binom{i+\alpha-1}{l} (-\Phi_s(z))^l \phi_s(z)dz \\
&= \int_0^\infty z^k \Phi_s(z)^{i+\alpha-1}\phi_s(z)dz \\
&\quad + \sum_{l=0}^{\infty} \binom{i+\alpha-1}{l} (-1)^{k+l} \int_0^\infty z^k \Phi_s(z)^l \phi_s(z)dz.
\end{aligned}
$$

Defining the quantity $Q_{i,j}^{(s)} = \int_0^\infty z^i \Phi_s(z)^j \phi_s(z)dz$, using (*Appendix eq.*(85)) and reindexing the sums:

$$
\begin{aligned}
\int_{-\infty}^\infty z^k \Phi_s(z)^{i+\alpha-1}\phi_s(z)dz &= \int_0^\infty z^k \sum_{m=0}^{\infty} v_m(i+\alpha-1)\Phi_s(z)^m \phi_s(z)dz \\
&\quad + \sum_{l=0}^{\infty} \binom{i+\alpha-1}{l} (-1)^{k+l} \int_0^\infty z^k \Phi_s(z)^l \phi_s(z)dz \\
&= \sum_{m=0}^{\infty} v_m(i+\alpha-1) \int_0^\infty z^k \Phi_s(z)^m \phi_s(z)dz \\
&\quad + \sum_{l=0}^{\infty} \binom{i+\alpha-1}{l} (-1)^{k+l} \int_0^\infty z^k \Phi_s(z)^l \phi_s(z)dz \\
&= \sum_{m=0}^{\infty} v_m(i+\alpha-1) Q_{k,m}^{(s)} + \sum_{l=0}^{\infty} \binom{i+\alpha-1}{l} (-1)^{k+l} Q_{k,l}^{(s)} \\
&= \sum_{n=0}^{\infty} \left(v_n(i+\alpha-1) + \binom{i+\alpha-1}{n} (-1)^{k+n} \right) Q_{k,n}^{(s)}, \qquad (34)
\end{aligned}
$$

where

$$
v_i(\alpha) = \sum_{j=i}^{\infty} (-1)^{i+j} \binom{\alpha}{j} \binom{j}{i}. \qquad (35)
$$

Finally from (*Appendix eq.*(87)) and substituting backward, (34) into (33) and (33) into (32), it follows that:

$$
E\left(X^r\right) = \frac{\mu^r}{B(\alpha,\beta)} \sum_{i=0}^{\infty} w_i(\beta) \sum_{k=0}^{r} \binom{r}{k} \left(\frac{\sigma}{\mu}\right)^k \sum_{n=0}^{\infty} \left(v_n(i+\alpha-1) + \binom{i+\alpha-1}{n} (-1)^{k+n} \right) Q_{k,n}^{(s)},
$$

where

$$Q_{k,n}^{(s)} = \frac{1}{(2\Gamma(1/s))^{n+1}} \sum_{m=0}^{n} \binom{n}{m} \Gamma(1/s)^{n-m} \sum_{l=0}^{\infty} c_{l,m} \Gamma\left(l + \frac{k+m+1}{s}\right)$$

and $c_{0,m} = s^m$, $c_{l,m} = 1/ls \sum_{r=1}^{l} (rm - l + r) \left((-1)^r / (1/s + r) r!\right) c_{l-r,m}$ for all $l \geq 1$.

4.5 Moment generating function

From the definition of a moment generating function, (30) and (31), we have that:

$$\begin{aligned} E\left(e^{tX}\right) &= \int_{-\infty}^{\infty} e^{tx} \frac{1}{\sigma B(\alpha,\beta)} \sum_{i=0}^{\infty} w_i(\beta) \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{i+\alpha-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx \\ &= \frac{1}{\sigma B(\alpha,\beta)} \sum_{i=0}^{\infty} w_i(\beta) \int_{-\infty}^{\infty} e^{tx} \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{i+\alpha-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx \end{aligned} \tag{36}$$

Using the transformation $z = (x-\mu)/\sigma$, which implies $x = \sigma z + \mu$ and $\frac{dx}{dz} = \sigma$, and (*Appendix eq.*(82)). The integral in (36) specifically becomes:

$$\int_{-\infty}^{\infty} e^{tx} \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{i+\alpha-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx = \sigma \int_{-\infty}^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz. \tag{37}$$

Splitting the range of the integral (37), using (*Appendix eq.*(80)),(*Appendix eq.*(81)) and (*Appendix eq.*(82)) we obtain:

$$\begin{aligned} \int_{-\infty}^{\infty} e^{tx} \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{i+\alpha-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx &= \sigma \int_{0}^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz \\ &\quad + \sigma \int_{-\infty}^{0} e^{t(\sigma z+\mu)} \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz \\ &= \sigma \int_{0}^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz \\ &\quad + \sigma \int_{0}^{\infty} e^{t(-\sigma z+\mu)} (1 - \Phi_s(z))^{i+\alpha-1} \phi_s(z) dz \end{aligned}$$

$$
\begin{aligned}
&= \sigma \int_0^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz \\
&\quad + \sigma \int_0^{\infty} e^{t(-\sigma z+\mu)} \sum_{n=0}^{\infty} \binom{i+\alpha-1}{n} (-\Phi_s(z))^n \phi_s(z) dz \\
&= \sigma \int_0^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz \\
&\quad + \sum_{n=0}^{\infty} \binom{i+\alpha-1}{n} (-1)^n \sigma \int_0^{\infty} e^{t(-\sigma z+\mu)} \Phi_s(z)^n \phi_s(z) dz.
\end{aligned}
$$

Defining the quantities $M_j^{(s)} = \int_0^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^j \phi_s(z) dz$, $M_j^{(s,-)} = \int_0^{\infty} e^{t(-\sigma z+\mu)} \Phi_s(z)^j \phi_s(z) dz$, using (*Appendix eq.*(85)) and reindexing the sums:

$$
\begin{aligned}
\sigma \int_{-\infty}^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^{i+\alpha-1} \phi_s(z) dz &= \sigma \int_0^{\infty} e^{t(\sigma z+\mu)} \sum_{m=0}^{\infty} v_m (i+\alpha-1) \Phi(z)^m \phi_s(z) dz \\
&\quad + \sum_{n=0}^{\infty} \binom{i+\alpha-1}{n} (-1)^n \sigma \int_0^{\infty} e^{t(-\sigma z+\mu)} \Phi_s(z)^n \phi_s(z) dz \\
&= \sum_{m=0}^{\infty} v_m (i+\alpha-1) \sigma \int_0^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^m \phi_s(z) dz \\
&\quad + \sum_{n=0}^{\infty} \binom{i+\alpha-1}{n} (-1)^n \sigma \int_0^{\infty} e^{t(-\sigma z+\mu)} \Phi_s(z)^n \phi_s(z) dz \\
&= \sum_{m=0}^{\infty} v_m (i+\alpha-1) \sigma M_m^{(s)} + \sum_{n=0}^{\infty} \binom{i+\alpha-1}{n} (-1)^{i+n} \sigma M_n^{(s,-)} \\
&= \sum_{n=0}^{\infty} \left[v_n (i+\alpha-1) \sigma M_n^{(s)} + \binom{i+\alpha-1}{n} (-1)^{i+n} \sigma M_n^{(s,-)} \right]. \quad (38)
\end{aligned}
$$

Finally from (*Appendix eq.*(89)&(90)) and substituting backward, (38) into (37) and (37) into (36), it follows that:

$$
E\left(e^{tX}\right) = \frac{1}{B(\alpha,\beta)} \sum_{i=0}^{\infty} w_i(\beta) \left(\sum_{n=0}^{\infty} \left[v_n (i+\alpha-1) M_n^{(s)} + \binom{i+\alpha-1}{n} (-1)^{i+n} M_n^{(s,-)} \right] \right),
$$

where

$$
M_n^{(s)} = \frac{e^{t\mu}}{2\Gamma\left(1/s\right)} \sum_{k=0}^{n} \binom{n}{k} \Gamma\left(1/s\right)^{n-k} \sum_{m=0}^{\infty} c_{m,k} \sum_{p=0}^{\infty} \frac{t\sigma^p}{p!} \Gamma\left(m + \frac{i+k+p+1}{s}\right),
$$

$$M_n^{(s,-)} = \frac{e^{t\mu}}{2\Gamma\left(1/s\right)} \sum_{k=0}^{n} \binom{n}{k} \Gamma\left(1/s\right)^{n-k} \sum_{m=0}^{\infty} c_{m,k} \sum_{p=0}^{\infty} \frac{(-t\sigma)^p}{p!} \Gamma\left(m + \frac{i+k+p+1}{s}\right)$$

and $c_{0,m} = s^m$, $c_{l,m} = 1/ls \sum_{r=1}^{l} (rm - l + r) \left((-1)^r/(1/s+r)r!\right) c_{l-r,m}$ for all $l \geq 1$.

4.6 Likelihood function

The likelihood function follows from (28),

$$\begin{aligned} L(x;\mu,\sigma,s,\alpha,\beta) &= \prod_{i=1}^{n} h(x_i) \\ &= \prod_{i=1}^{n} \frac{1}{\sigma B(\alpha,\beta)} \left[\Phi_s\left(\frac{x_i-\mu}{\sigma}\right)\right]^{\alpha-1} \left[1-\Phi_s\left(\frac{x_i-\mu}{\sigma}\right)\right]^{\beta-1} \phi_s\left(\frac{x_i-\mu}{\sigma}\right) \end{aligned} \quad (39)$$

Due to the complexity of the solutions to the maximum likelihood estimators (MLE) in (39) are not derived analytically. This function is however used to numerically obtain the MLE estimates in Section 6 for estimates of $\mu, \sigma, s, \alpha, \beta$.

5 The Kumaraswamy generalised normal distribution

In this Section the Kumaraswamy generalised normal distribution (KGN) distribution is proposed for the first time using the generator approach. The derivations of the PDF, CDF, hazard function, moments, moment generating function and likelihood function are included in this Section.

5.1 Introduction

The PDF $h(x)$ and CDF $H(x)$ is obtained using definition (2), baseline CDF (11), generator CDF (6), (20) and (22):

$$h(x) = \frac{\alpha\beta}{\sigma} \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha-1} \left(1-\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha}\right)^{\beta-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) \quad (40)$$

and

$$\begin{aligned} H(x) &= 1 - \left(1-\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha}\right)^{\beta} \\ &\quad -\infty < x < \infty, -\infty < \mu < \infty, s, \sigma, \alpha, \beta > 0. \end{aligned} \quad (41)$$

This random variable will be denoted as $X \sim KGN(\mu, \sigma, s, \alpha, \beta)$.

In Figure 10 the PDF of the KGN (dashed) is compared to the PDF of the BGN (shaded from axis) for various combinations of its parameters. The KGN has the same flexibility in real data fitting as the BGN but with previously mentioned tractability advantages. Notice that the two PDFs are markedly different for sufficiently small or large values of α. Note that the KGN is equal to the Kumaraswamy normal (KN) distribution for $s = 2$, with the only difference of $\sigma_{KN} = \sqrt{2}\sigma$, see [4].

5.2 Hazard function

The hazard rate function of the KGN distribution is derived from (*Appendix eq.*(56)), (28) and (29):

$$\begin{aligned} \tau(x) &= \frac{\alpha\beta\left(\Phi_s\left(\frac{x-\mu}{\sigma}\right)\right)^{\alpha-1}\left(1-\left(\Phi_s\left(\frac{x-\mu}{\sigma}\right)\right)^{\alpha}\right)^{\beta-1}\phi_s\left(\frac{x-\mu}{\sigma}\right)}{\left(1-\Phi_s\left(\frac{x-\mu}{\sigma}\right)\right)^{\beta}} \\ &= \frac{\alpha\beta\left(\Phi_s\left(\frac{x-\mu}{\sigma}\right)\right)^{\alpha-1}\phi_s\left(\frac{x-\mu}{\sigma}\right)}{1-\Phi_s\left(\frac{x-\mu}{\sigma}\right)}. \end{aligned}$$

In Figure 11 the KGN hazard function (dashed) is compared to the BGN hazard function (shaded from axis). Note the pronounced effect of the σ parameter on KGN hazard function compared to the BGN hazard function in Figure 11 (upper left corner).

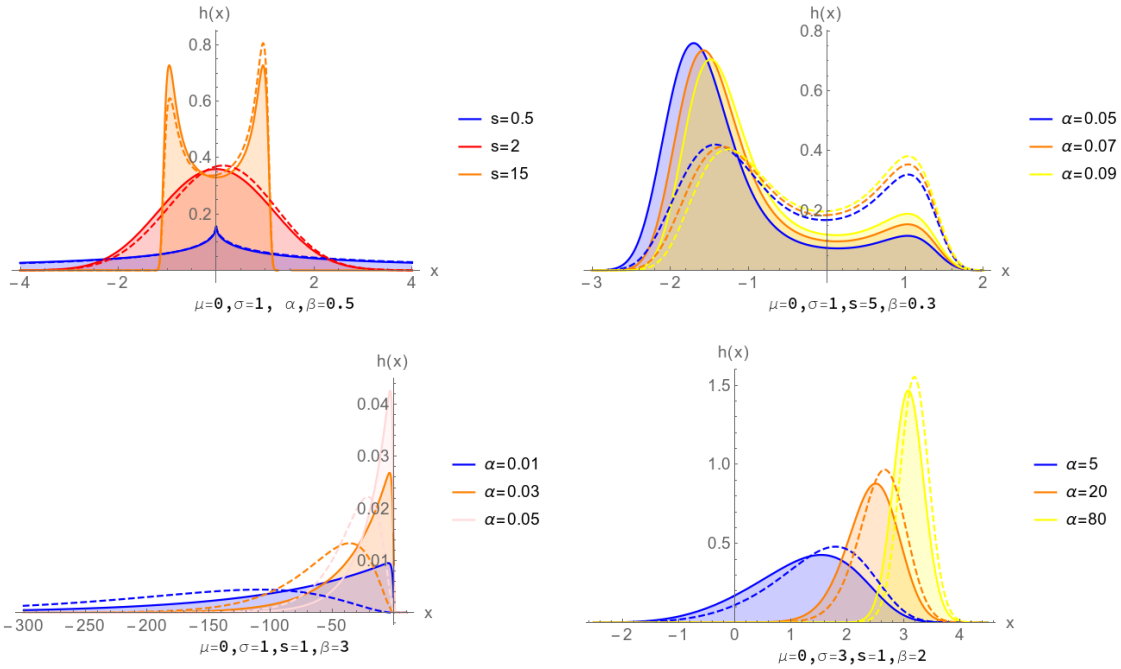


Figure 10: The effect of the parameters on KGN PDF (dashed) compared to the BGN PDF (shaded from the axis).

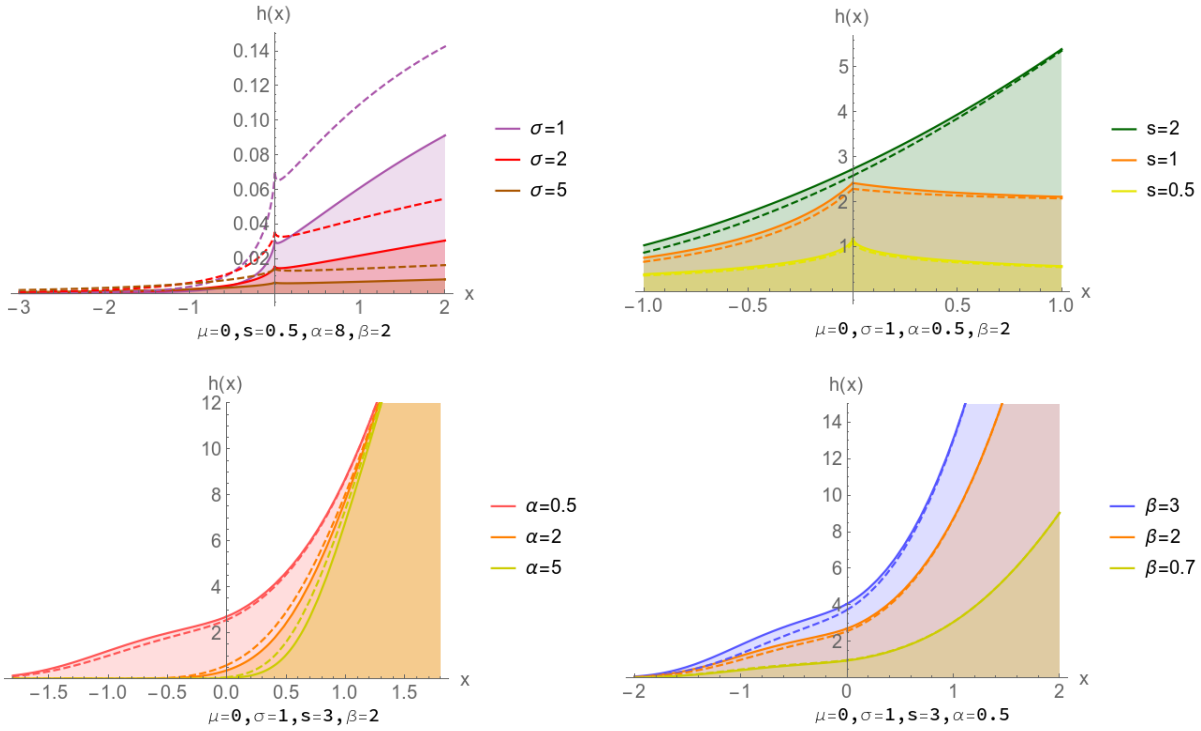


Figure 11: The effect of the parameters on KGN hazard function (dashed) compared to the BGN hazard function (shaded from the axis).

5.3 Infinite expansion of the probability density function

From (*Appendix eq.*(82)),(*Appendix eq.*(68)) and defining weights $w_i(\beta)$ it follows that:

$$
\begin{aligned}
h(x) &= \alpha\beta\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha-1}\left(1-\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha}\right)^{\beta-1}\phi_s\left(\frac{x-\mu}{\sigma}\right)\\
&= \alpha\beta\phi_s\left(\frac{x-\mu}{\sigma}\right)\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha-1}\sum_{i=0}^{\infty}\binom{\beta-1}{i}\left(-\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha}\right)^{i}\\
&= \alpha\beta\sum_{i=0}^{\infty}\binom{\beta-1}{i}(-1)^i\phi_s\left(\frac{x-\mu}{\sigma}\right)\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha i+\alpha-1}\\
&= \frac{\alpha\beta}{\sigma}\sum_{i=0}^{\infty}\frac{\Gamma(\beta)\,(-1)^i}{\Gamma(\beta-i)i!}\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha(i+1)-1}\phi_s\left(\frac{x-\mu}{\sigma}\right)\\
&= \frac{\alpha\beta}{\sigma}\sum_{i=1}^{\infty}w_i(\beta)\Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha(i+1)-1}\phi_s\left(\frac{x-\mu}{\sigma}\right), -\infty<x<\infty, 0<\alpha, 0<\beta, \qquad (42)
\end{aligned}
$$

where $w_i(\beta)$ is defined in (31). Note that the upper bound of the sums are $\alpha-1$ and $\beta-1$ for integer values of α and β, see (*Appendix eq.*(83)).

5.4 Moments

From the definition of moments, (42) and (31) it follows that:

$$\begin{aligned} E\left(X^r\right) &= \int_{-\infty}^{\infty} x^r \frac{\alpha\beta}{\sigma} \sum_{i=0}^{\infty} w_i(\beta) \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha(i+1)-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx \\ &= \frac{\alpha\beta}{\sigma} \sum_{i=0}^{\infty} w_i(\beta) \int_{-\infty}^{\infty} x^r \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha(i+1)-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx \end{aligned} \tag{43}$$

Using the transformation $z = {}^{(x-\mu)}/_{\sigma}$, which implies $x = \sigma z + \mu$ and $\frac{dx}{dz} = \sigma$, and (*Appendix eq.*(82)). The integral in (43) specifically becomes:

$$\begin{aligned} \int_{-\infty}^{\infty} x^r \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha(i+1)-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx &= \sigma \int_{-\infty}^{\infty} (\sigma z+\mu)^r \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\ &= \sigma \int_0^{\infty} \sum_{k=0}^{r} \binom{r}{k} \mu^{r-k} (\sigma z)^k \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\ &= \sigma\mu^r \sum_{k=0}^{r} \binom{r}{k} \sigma^k \mu^{-k} \int_{-\infty}^{\infty} z^k \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\ &= \sigma\mu^r \sum_{k=0}^{r} \binom{r}{k} \left(\frac{\sigma}{\mu}\right)^k \int_{-\infty}^{\infty} z^k \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \end{aligned} \tag{44}$$

Splitting the range of the integral (44), using (*Appendix eq.*(80)),(*Appendix eq.*(81)) and (*Appendix eq.*(82)) we obtain:

$$\begin{aligned} \int_{-\infty}^{\infty} z^k \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz &= \int_0^{\infty} z^k \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\ &\quad + \int_{-\infty}^{0} z^k \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\ &= \int_0^{\infty} z^k \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\ &\quad + \int_0^{\infty} (-z)^k (1-\Phi_s(z))^{\alpha(i+1)-1} \phi_s(z) dz \\ &= \int_0^{\infty} z^k \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\ &\quad + (-1)^k \int_0^{\infty} z^k \sum_{l=0}^{\infty} \binom{\alpha(i+1)-1}{l} (-\Phi_s(z))^l \phi_s(z) dz \end{aligned}$$

$$
\begin{aligned}
&= \int_0^\infty z^k \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\
&\quad + \sum_{l=0}^{\infty} \binom{\alpha(i+1)-1}{l} (-1)^{k+l} \int_0^\infty z^k \Phi_s(z)^l \phi_s(z) dz
\end{aligned}
$$

Defining the quantity $Q_{i,j}^{(s)} = \int_0^\infty z^i \Phi_s(z)^j \phi_s(z) dz$, using (*Appendix eq.*(85)) and reindexing the sums:

$$
\begin{aligned}
\int_{-\infty}^{\infty} z^k \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz &= \int_0^\infty z^k \Phi_s \sum_{m=0}^{\infty} v_m(\alpha(i+1)-1) \Phi_s(z)^m \phi_s(z) dz \\
&\quad + \sum_{l=0}^{\infty} \binom{\alpha(i+1)-1}{l} (-1)^{k+l} \int_0^\infty z^k \Phi_s(z)^l \phi_s(z) dz \\
&= \sum_{m=0}^{\infty} v_m(\alpha(i+1)-1) \int_0^\infty z^k \Phi_s(z)^m \phi_s(z) dz \\
&\quad + \sum_{l=0}^{\infty} \binom{\alpha(i+1)-1}{l} (-1)^{k+l} \int_0^\infty z^k \Phi_s(z)^l \phi_s(z) dz \\
&= \sum_{m=0}^{\infty} v_m(\alpha(i+1)-1) Q_{k,m}^{(s)} + \sum_{l=0}^{\infty} \binom{\alpha(i+1)-1}{l} (-1)^{k+l} Q_{k,l}^{(s)} \\
&= \sum_{n=0}^{\infty} \left(v_n(\alpha(i+1)-1) + \binom{\alpha(i+1)-1}{n} (-1)^{k+n} \right) Q_{k,n}^{(s)}, \qquad (45)
\end{aligned}
$$

where $v_i(\alpha)$ is defined in (35).

Finally from (*Appendix eq.*(87)) and substituting backward, (45) into (44) and (44) into (43), it follows that:

$$
E(X^r) = \alpha\beta\mu^r \sum_{i=0}^{\infty} w_i(\beta) \sum_{k=0}^{r} \binom{r}{k} \left(\frac{\sigma}{\mu}\right)^k \sum_{n=0}^{\infty} \left(v_n(\alpha(i+1)-1) + \binom{\alpha(i+1)-1}{n} (-1)^{k+n} \right) Q_{k,n}^{(s)},
$$

where

$$
Q_{k,n}^{(s)} = \frac{1}{(2\Gamma(1/s))^{n+1}} \sum_{m=0}^{n} \binom{n}{m} \Gamma(1/s)^{n-m} \sum_{l=0}^{\infty} c_{l,m} \Gamma\left(l + \frac{k+m+1}{s}\right)
$$

and $c_{0,m} = s^m$, $c_{l,m} = 1/ls \sum_{r=1}^{l} (rm - l + r) \left((-1)^r / (1/s + r) r!\right) c_{l-r,m}$ for all $l \geq 1$.

5.5 Moment generating function

From the definition of a moment generating function, (42) and (31) it follows that:

$$
\begin{aligned}
E\left(e^{tX}\right) &= \int_{-\infty}^{\infty} e^{tx} \frac{\alpha\beta}{\sigma} \sum_{i=0}^{\infty} w_i(\beta) \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha(i+1)-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx \\
&= \frac{\alpha\beta}{\sigma} \sum_{i=0}^{\infty} w_i(\beta) \int_{-\infty}^{\infty} e^{tx} \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha(i+1)-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx \qquad (46)
\end{aligned}
$$

Using the transformation $z = {}^{(x-\mu)}/_{\sigma}$, which implies $x = \sigma z + \mu$ and $\frac{dx}{dz} = \sigma$, and (*Appendix eq.*(82)). The integral in (46) specifically becomes:

$$
\int_{-\infty}^{\infty} e^{tx} \Phi_s\left(\frac{x-\mu}{\sigma}\right)^{\alpha(i+1)-1} \phi_s\left(\frac{x-\mu}{\sigma}\right) dx = \sigma \int_{-\infty}^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \qquad (47)
$$

Splitting the range of the integral (47), using (*Appendix eq.*(80)),(*Appendix eq.*(81)) and (*Appendix eq.*(82)) we obtain:

$$
\begin{aligned}
\int_{-\infty}^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz &= \sigma \int_{0}^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\
&\quad + \sigma \int_{-\infty}^{0} e^{t(\sigma z+\mu)} \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\
&= \sigma \int_{0}^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\
&\quad + \sigma \int_{0}^{\infty} e^{t(-\sigma z+\mu)} (1-\Phi_s(z))^{\alpha(i+1)-1} \phi_s(z) dz \\
&= \sigma \int_{0}^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\
&\quad + \sigma \int_{0}^{\infty} e^{t(-\sigma z+\mu)} \sum_{n=0}^{\infty} \binom{\alpha(i+1)-1}{l} (-\Phi_s(z))^n \phi_s(z) dz \\
&= \sigma \int_{0}^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^{\alpha(i+1)-1} \phi_s(z) dz \\
&\quad + \sum_{n=0}^{\infty} \binom{\alpha(i+1)-1}{n} (-1)^n \sigma \int_{0}^{\infty} e^{t(-\sigma z+\mu)} \Phi_s(z)^n \phi_s(z) dz
\end{aligned} \qquad (48)
$$

Defining the quantities $M_j^{(s)} = \int_0^{\infty} e^{t(\sigma z+\mu)} \Phi_s(z)^j \phi_s(z) dz$, $M_j^{(s,-)} = \int_0^{\infty} e^{t(-\sigma z+\mu)} \Phi_s(z)^j \phi_s(z) dz$, as

before using (*Appendix eq.*(85)) and reindexing the sums:

$$\begin{aligned}
\sigma\int_{-\infty}^{\infty} e^{t(\sigma z+\mu)}\Phi_s(z)^{\alpha(i+1)-1}\phi_s(z)dz &= \sigma\int_0^{\infty} e^{t(\sigma z+\mu)}\sum_{m=0}^{\infty} v_m(\alpha(i+1)-1)\Phi(z)^m\phi_s(z)dz \\
&\quad+\sum_{n=0}^{\infty}\binom{\alpha(i+1)-1}{n}(-1)^n\sigma\int_0^{\infty} e^{t(-\sigma z+\mu)}\Phi_s(z)^n\phi_s(z)dz \\
&= \sum_{m=0}^{\infty} v_m(\alpha(i+1)-1)\sigma\int_0^{\infty} e^{t(\sigma z+\mu)}\Phi_s(z)^m\phi_s(z)dz \\
&\quad+\sum_{n=0}^{\infty}\binom{\alpha(i+1)-1}{n}(-1)^n\sigma\int_0^{\infty} e^{t(-\sigma z+\mu)}\Phi_s(z)^n\phi_s(z)dz \\
&= \sum_{m=0}^{\infty} v_m(i+\alpha-1)\sigma M_m^{(s)}+\sum_{n=0}^{\infty}\binom{\alpha(i+1)-1}{n}(-1)^{i+n}\sigma M_n^{(s,-)} \\
&= \sum_{n=0}^{\infty}\left[v_n(i+\alpha-1)\sigma M_n^{(s)}+\binom{\alpha(i+1)-1}{n}(-1)^{i+n}\sigma M_n^{(s,-)}\right],
\end{aligned} \tag{49}$$

where $v_i(\alpha)$ is defined in (35).

Finally from (*Appendix eq.*(89)&(90)) and substituting backward, (49) into (45) and (44) into (46), it follows that:

$$E\left(e^{tX}\right) = \alpha\beta\sum_{i=0}^{\infty} w_i(\beta)\left(\sum_{n=0}^{\infty}\left[v_n(i+\alpha-1)M_n^{(s)}+\binom{\alpha(i+1)-1}{n}(-1)^{i+n}M_n^{(s,-)}\right]\right),$$

where

$$M_n^{(s)} = \frac{e^{t\mu}}{2\Gamma\left(1/s\right)}\sum_{k=0}^{n}\binom{n}{k}\Gamma\left(1/s\right)^{n-k}\sum_{m=0}^{\infty} c_{m,k}\sum_{p=0}^{\infty}\frac{t\sigma^p}{p!}\Gamma\left(m+\frac{i+k+p+1}{s}\right),$$

$$M_n^{(s,-)} = \frac{e^{t\mu}}{2\Gamma\left(1/s\right)}\sum_{k=0}^{n}\binom{n}{k}\Gamma\left(1/s\right)^{n-k}\sum_{m=0}^{\infty} c_{m,k}\sum_{p=0}^{\infty}\frac{(-t\sigma)^p}{p!}\Gamma\left(m+\frac{i+k+p+1}{s}\right)$$

and $c_{0,m}=s^m$, $c_{l,m}=1/ls\sum_{r=1}^{l}(rm-l+r)\left((-1)^r/(1/s+r)r!\right)c_{l-r,m}$ for all $l\geq 1$.

5.6 Likelihood function

The likelihood function follows from (40),

$$
\begin{aligned}
L(x;\lambda,\alpha,\beta) &= \prod_{i=1}^{n} h(x_i) \\
&= \prod_{i=1}^{n} \frac{\alpha\beta}{\sigma}\Phi_s\left(\frac{x_i-\mu}{\sigma}\right)^{\alpha-1}\left(1-\Phi_s\left(\frac{x_i-\mu}{\sigma}\right)^{\alpha}\right)^{\beta-1}\phi_s\left(\frac{x_i-\mu}{\sigma}\right) \qquad (50)
\end{aligned}
$$

Due to the complexity of the solutions to the maximum likelihood estimators (MLE) in (50) are not derived analytically. This function is however used to numerically obtain the MLE estimates in Section 6 for estimates of $\mu, \sigma, s, \alpha, \beta$.

6 Application and comparison of the generators

In this Section a random number generator (RNG) for the BW, KW, BGN and KGN are introduced. The two generators are then compared by application to real data sets and their performance is measured using AIC, BIC and CAIC.

6.1 Random number generator

Specifically we show the RNG algorithm for the BGN as done by [3] only, since the other RNGs work similarly.

Let be a $X \sim BGN(\mu, \sigma, s, \alpha, \beta)$ distribution, CDF $H(\cdot)$ of X is given below by definition (2):

$$
H(x) = F(G(x)),
$$

where $F(\cdot)$ is the CDF of $Beta(\alpha, \beta)$ distribution and $G(\cdot)$ is the CDF of a $GN(\mu, \sigma, s)$ distribution.

From the inverse probability integral transformation it follow that [27]:

$$
H(X) = F(G(X)) \overset{D}{=} U, \qquad (51)
$$

where $U \sim Uni(0, 1)$ and " $\overset{D}{=}$ " indicates equality in distribution.

From (51) we have that:

$$
G(X) = F^{-1}(U) \overset{D}{=} B, \qquad (52)
$$

where $B \sim Beta(\alpha, \beta)$ and $F^{-1}(\cdot)$ is the quantile function of a $Beta(\alpha, \beta)$ distribution.

Lastly from (52):

$$X = G^{-1}(B), \tag{53}$$

where $G^{-1}(\cdot)$ is the quantile function of a $GN(\mu, \sigma, s)$ distribution.

Considering the baseline CDF (11), from (58) we have:

$$\begin{aligned}
G(x) &= \begin{cases} \dfrac{\Gamma\left(1/s, \left(\frac{\mu-x}{\sigma}\right)^s\right)}{2\Gamma\left(1/s\right)} \text{if } x \leq \mu \\ 1 - \dfrac{\Gamma\left(1/s, \left(\frac{x-\mu}{\sigma}\right)^s\right)}{2\Gamma\left(1/s\right)} \text{if } x > \mu \end{cases} \\
&= \begin{cases} \dfrac{1}{2}\left(\dfrac{\Gamma(1/s) - \gamma(1/s, \left(\frac{x-\mu}{\sigma}\right)^s)}{\Gamma\left(1/s\right)}\right) \text{if } x \leq \mu \\ \dfrac{1}{2}\left(2 - \dfrac{\Gamma(1/s) - \gamma(1/s, \left(\frac{x-\mu}{\sigma}\right)^s)}{\Gamma\left(1/s\right)}\right) \text{if } x > \mu \end{cases} \\
&= \begin{cases} \dfrac{1}{2}\left\{1 - \dfrac{\gamma\left(1/s, \left(\frac{\mu-x}{\sigma}\right)^s\right)}{\Gamma\left(1/s\right)}\right\} \text{if } x \leq \mu \\ \dfrac{1}{2}\left\{1 + \dfrac{\gamma\left(1/s, \left(\frac{x-\mu}{\sigma}\right)^s\right)}{\Gamma\left(1/s\right)}\right\} \text{if } x > \mu \end{cases}.
\end{aligned}$$

Lastly from (70) it follows:

$$G(x) = \begin{cases} \dfrac{1}{2}\left\{1 - H\left(\left(\dfrac{\mu - x}{\sigma}\right)^s\right)\right\} \text{if } x \leq \mu \\ \dfrac{1}{2}\left\{1 + H\left(\dfrac{x - \mu}{\sigma}\right)^s\right\} \text{if } x > \mu \end{cases} \tag{54}$$

where $H(\cdot)$ is the CDF of a $Gamma(1/s, 1)$ distribution.

To derive the inverse function $G^{-1}(\cdot)$ of $G(\cdot)$, for the use in statement (53), we set (54) equal to dummy variable f.

For $x \leq \mu$:

$$\begin{aligned}
f &= \frac{1}{2}\left\{1 - H\left(\left(\frac{\mu - x}{\sigma}\right)^s\right)\right\} \\
1 - 2f &= H\left(\left(\frac{\mu - x}{\sigma}\right)^s\right) \\
\frac{\mu - x}{\sigma} &= \left[H^{-1}\left(1 - 2f\right)\right]^{\frac{1}{s}} \\
\therefore x &= \mu - \sigma\left[H^{-1}(1 - 2f)\right]^{\frac{1}{s}}
\end{aligned} \tag{55}$$

with a bound of

$$\begin{aligned} 0 &\leq H\left(\left(\tfrac{\mu-x}{\sigma}\right)^s\right) \leq 1 \\ 0 &\leq 1-2f \leq 1 \end{aligned}$$

$$0 \leq f \leq \tfrac{1}{2},$$

which follows from (55) and since $H(\cdot)$ is a valid CDF.

Similarly for $x > \mu$:

$$f = \frac{1}{2}\left\{1 + H\left(\frac{x-\mu}{\sigma}\right)^s\right\},$$

with implies

$$x = \mu + \sigma\left[H^{-1}(2f-1)\right]^{\frac{1}{s}},$$

with a bound of $\frac{1}{2} < f \leq 1$.

where $H^{-1}(\cdot)$ is the quantile function of a $Gamma(1/s, 1)$ distribution.

Therefore the RNG is algorithmically described below:

1. *Generate* a value f from a $Beta(\alpha, \beta)$ distribution.
2. *If* $0 \leq f < \frac{1}{2}$ *then* calculate $x = \mu - \sigma\left[H^{-1}(1-2f)\right]^{\frac{1}{s}}$ *else* $x = \mu - \sigma\left[H^{-1}(1-2f)\right]^{\frac{1}{s}}$, using the quantile function of a $Gamma(1/s, 1)$ distribution.
3. *return* x.

We now generate a 5000000 points from a $BGN(104.93, 331.57, 0.25, 285.58, 248.88)$ distribution using SAS[1] proc iml Listing 1. This simulation corresponds to the estimated BGN distribution fitted to the relapse-time data in Section 6.2. Table 1 gives a comparison of the first two theoretical and empirical moments and Figure 12 displays a histogram and kernel density of the sample. Comparing Figure 12 to Figure 14 it can be seen that the simulated PDF approximates the theoretical PDF. In Table 1 it is clear the empirical moment values approximate the theoretical moment values.

[1]

Moment	μ_1	μ_2
Theoretical	2158.01	7323340
Empirical	2157.95	7319369.1
Absolute Error	0.06	3970.92

Table 1: Theoretical and empirical moments for 5000000 $BGN(104.93, 331.57, 0.25, 285.58, 248.88)$ values.

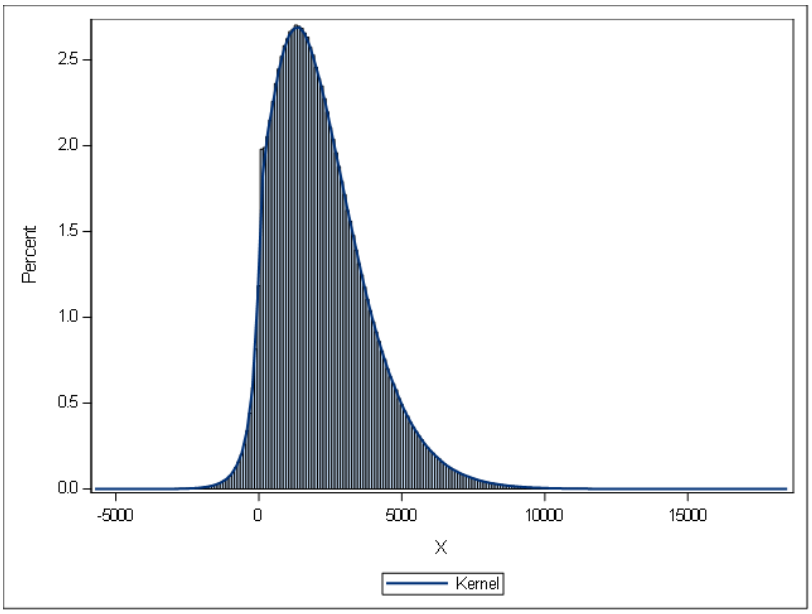


Figure 12: Sample of 5000000 generated $BGN(104.93, 331.57, 0.25, 285.58, 248.88)$ points with estimated kernel density.

6.2 Application to data

In this Section, we obtain MLEs and various goodness-of-fit statistics for the applications of each model to real data sets. The maximisation of the log-likelihood function is done by the optim subroutine in R [26].

Firstly, we consider data that represents the price (US dollars) of 308 diamonds. We evaluate the data with the BW and KW distributions. The fitted PDFs are shown in Figure 2; Table 2 contains the MLEs of the parameters; and Table 3 contains the values of the AIC, BIC and CAIC for the price data. Lower values of these statistics indicate better model fitting.

	κ	λ	α	β
BW	0.170	335.125	61.531	19.753
KW	0.151	7.305	38.110	21.660

Table 2: MLEs for the price data.

	AIC	BIC	CAIC
BW	5,605.745	5,620.665	5,624.665
KW	5,599.298	5,614.218	5,618.218

Table 3: Information criteria for the price data.

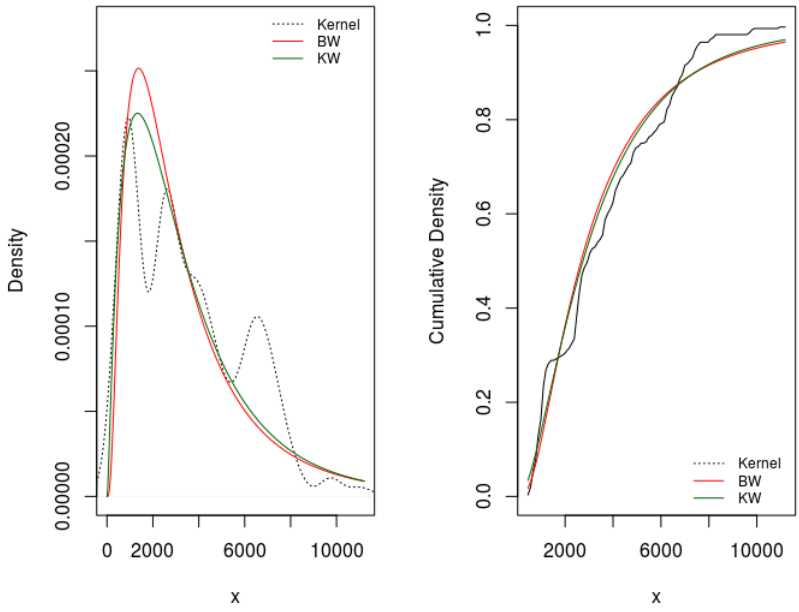


Figure 13: Fitted PDF of the BW, KW and empirical kernel density for the price data.

Since all the values of the statistics in 3 are smaller for the KW distribution than the BW distribution, the KW is a better model to explain the weight data.

Secondly, we compare the KGN model with the beta generalised normal (BGN). We consider data based on the National Wilm's Tumor Study [2]. The data represents observed relapse times of 460 children diagnosed with stage 4 embryonal cancer of the kidney known as Wilm's tumour. The fitted PDFs are shown in Figure 14; Table 4 contains the MLEs of the parameters; and Table 5 contains the values of the AIC, BIC and CAIC for the relapse-time data.

	μ	σ	s	α	β
BGN	104.938	331.574	0.250	285.582	248.876
KGN	118.172	113.295	0.273	11.195	190.856

Table 4: MLEs for the relapse-time data.

	AIC	BIC	CAIC
BGN	8,019.805	8,036.330	8,040.330
KGN	7,995.733	8,012.258	8,016.258

Table 5: Information criteria for the relapse-time data.

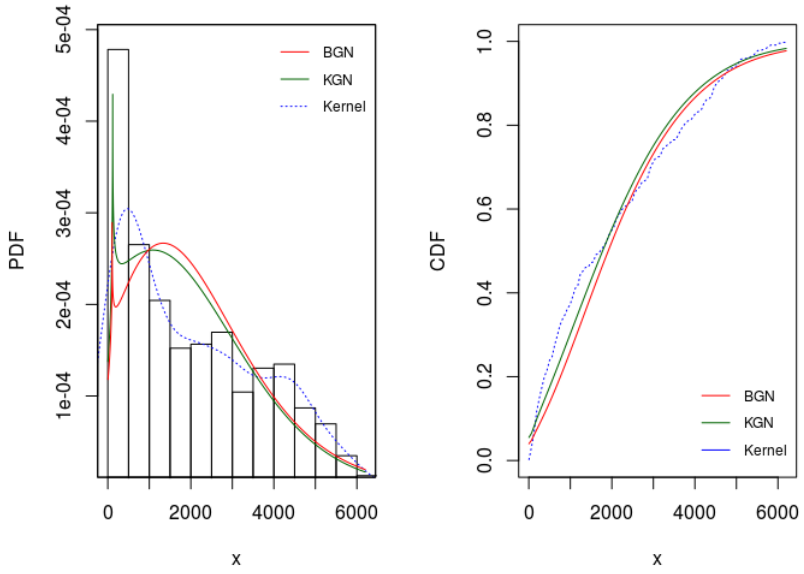


Figure 14: Fitted PDF of the BGN, KGN and empirical kernel density for the relapse-time data.

Since all the values of the criteria in Table 5 are smaller for the KGN distribution than the BGN distribution, the KGN is a better model to explain the relapse-time data.

7 Conclusion

The research presented in this report is focussed with the construction of Kumaraswamy and beta generated distributions. The distribution function of such a distribution is defined as $H(\cdot) = F(G(\cdot))$, where F is the distribution function of the generator distribution and G is the distribution function of the baseline distribution.

By combining the distribution functions of the Kumaraswamy and generalised normal distributions in this way, we introduce the KGN distribution. Computationally manageable expressions for the moments and moment generating functions are derived for the KGN distribution. The performance of the KGN on observed data sets shows that this distribution is an useful contribution to the Kumaraswamy generated family of distributions.

The beta and Kumaraswamy generators function in fundamentally similar ways which is evident in

the performance of these distributions on real data sets. The Kumaraswamy generator has tractability advantages over the beta generator, since it has a closed-form distribution function. In contrast to this, the infinite expansions of mathematical properties are less complex for the beta generator. The right tail prominence of the Kumaraswamy generator is useful in modelling specific data sets as shown by the applications of the study.

Future work could include the derivation of the order statistic distribution of the KGN, the moments of these order statistics and extensions of the model to the multivariate case.

References

[1] L. J. Bain and M. Engelhart. *Introduction to Probability and Mathematical Statistics*. Brooks/Cole, second edition, 1992.

[2] N. E. Breslow and N. Chatterjee. Design and analysis of two-phase studies with binary outcome applied to Wilms tumour prognosis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(4):457–468, 1999.

[3] R.J. Cintra, L.C. Rêgo, G.M. Cordeiro, and A.D.C. Nascimento. Beta generalized normal distribution with an application for SAR image processing. *Statistics*, 48(2):279–294, 2014.

[4] G. M. Cordeiro and R. d. S. Bager. Moments for some Kumaraswamy generalized distributions. *Communications in Statistics-Theory and Methods*, 44(13):2720–2737, 2014.

[5] G. M. Cordeiro, S. Nadarajah, and E. M. M. Ortega. General results for the beta Weibull distribution. *Journal of Statistical Computation and Simulation*, 83(6):1082–1114, 2013.

[6] G. M. Cordeiro, E. MM. Ortega, and S. Nadarajah. The Kumaraswamy Weibull distribution with application to failure data. *Journal of the Franklin Institute*, 347(8):1399–1429, 2010.

[7] G.M. Cordeiro and M. de Castro. A new family of generalized distributions. *Journal of School Statictical Computation and Simulation*, 81(7):883–898, 2011.

[8] N. Eugene, C. Lee, and F. Famoye. Beta-normal distribution and its applications. *Communications in Statistics-Theory and methods*, 31(4):497–512, 2002.

[9] S. Evert and M. Baroni. *zipfR*: Word Frequency Distributions in R. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Posters and Demonstrations Sessions*, pages 29–32, Prague, Czech Republic, 2007. (R package version 0.6-6 of 2012-04-03).

[10] I.S. Gradshteyn and I.M. Ryhzhik. *Table of Integrals, Series and Products*. Academic Press, 2007.

[11] Wolfram Research, Inc. Mathematica, Version 11.2. Champaign, IL, 2017.

[12] M.C. Jones. Families of Distributions Arising from Distributions of Order Statistics. *Test*, 13(1):1–43, 2004.

[13] M.C. Jones. Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology*, 6(1):70–81, 2009.

[14] S. Kotz, N. Balakrishnan, and N. L. Johnson. *Continuous multivariate distributions, models and applications*. John Wiley & Sons, 2004.

[15] P. Kumaraswamy. A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2):79–88, 1980.

[16] C. Lee, F. Famoye, and O. Olumolade. Beta-Weibull distribution: Some Properties and Applications to Censored Data. *Journal of Modern Applied Statistical Methods*, 6(1):173–186, 2007.

[17] S.L. Makgai, A. Bekker, J.T. Ferreira, and M. Arashi. New results from a beta-Pareto family. *South African Statistical Journal*, 51(2):345–360, 2017.

[18] A.W. Marshall and I. Olkin. A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families. *Biometrika*, 84(3):641–652, 1997.

[19] F. Merovci, M. A. Khaleel, N. A. Ibrahim, and M. Shitan. The beta Burr type X distribution properties with application. *SpringerPlus*, 5(1):1–18, 2016.

[20] G. S. Mudholkar and D. K. Srivastava. Exponentiated Weibull family for analyzing bathtub failure-rate data. *IEEE Transactions on Reliability*, 42(2):299–302, 1993.

[21] S. Nadarajah. A generalized Normal Distribution. *Journal of Applied Statistics*, 32(7):685–694, 2005.

[22] S. Nadarajah. Explicit expressions for moments of order statistics. *Statistics & Probability Letters*, 78(2):196–205, 2008.

[23] S. Nadarajah and A. K. Gupta. On the moments of the exponentiated Weibull distribution. *Communications in Statistics-Theory and Methods*, 34(2):253–256, 2005.

[24] S. Nadarajah, M. Teimouri, and S. H. Shih. Modified beta distributions. *Sankhya B*, 76(1):19–48, 2014.

[25] K. B. Oldham, J. Myland, and J. Spanier. *An Atlas of Functions*. Springer-Verlag New York, 2 edition, 2009.

[26] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[27] Ross S. *Simulation*. Academic Press, 2006.

[28] M. Subbotin. On the law of frequency of error. *Matematicheskii Sbornik*, 31(2):296–301, 1923.

[29] M.H. Tahir and S. Nadarajah. Parameter induction in continuous univariate distributions: Well-established G families. *Anais da Academia Brasileira de Ciências*, 87(2):539–568, 2015.

[30] T.M. Therneau and P.M. Grambsch. *Modeling Survival Data: Extending the Cox Model*. Springer, New York, 2000.

[31] T.W. Yee, J. Stoklosa, and R.M. Huggins. The VGAM Package for Capture-Recapture Data Using the Conditional Likelihood. *Journal of Statistical Software*, 65(5):1–33, 2015.

[32] K. Zografos and N. Balakrishnan. On families of beta-and generalized gamma-generated distributions and associated inference. *Statistical Methodology*, 6(4):344–362, 2009.

Appendix

Abbreviations and symbols

PDF	Probability density function
CDF	Cumulative density function
BW	Beta Weibull
KW	Kumaraswamy Weibull
GN	Generalised normal
BN	Beta normal
KN	Kumaraswamy normal
BGN	Beta generalised normal
KGN	Kumaraswamy generalised normal
AIC	Akaike information criterion
BIC	Bayesian information criterion
CAIC	Consistent Akaike information criterion
RGN	Random number generator
$h(x)$	PDF of random variable X
$H(x)$	CDF of random variable X
$g(x)$	PDF of the baseline distribution
$G(x)$	CDF of the baseline distribution
$f(x)$	PDF of the generator distribution
$F(x)$	CDF of the generator distribution
$\tau(x)$	Hazard function of distribution X
$L(x;\theta)$	Likelihood function of parameters θ given values x
$\Gamma(\alpha)$	Gamma function
$\gamma(\alpha, x)$	Incomplete lower gamma function
$\Gamma(\alpha, y)$	Incomplete upper gamma function
$B(\alpha, \beta)$	Beta function
$B(x;\alpha, \beta)$	Incomplete beta function
$I_x(\alpha, \beta)$	Beta function ratio
$(\alpha)_i$	Pochhammer coefficient
$\binom{i}{j}$	Binomial coefficient
$\phi_s(z)$	PDF of $GN \sim (0, 1, s)$ distribution

$\Phi(z)$	CDF of $GN \sim (0, 1, s)$ distribution
Re	Set of real numbers
$X \sim Normal(\mu, \sigma)$	Normal distribution with parameters μ and σ
$X \sim Uni(a, b)$	Uniform distribution with parameters a and b
$X \sim Exp(\theta)$	Exponential distribution with parameter θ
$X \sim Beta(\alpha, \beta)$	Beta type I distribution with parameters α and β
$X \sim Kumaraswamy(\alpha, \beta)$	Kumaraswamy distribution with parameters α and β
$X \sim BW(k, \lambda, \alpha, \beta)$	Beta Weibull distribution with parameters k, λ, α and β
$X \sim KW(k, \lambda, \alpha, \beta)$	Kumaraswamy Weibull distribution with parameters k, λ, α and β
$X \sim BGN(\mu, \sigma, s, \alpha, \beta)$	Beta generalised normal distribution with parameters μ, σ, s, α and β
$X \sim KGN(\mu, \sigma, s, \alpha, \beta)$	Kumaraswamy generalised normal distribution with parameters μ, σ, s, α and β

Results

Result 1

The hazard function for a random variable X with PDF $f(x)$ and CDF $F(x)$ is defined as

$$\tau(x) = \frac{f(x)}{1 - F(x)}. \tag{56}$$

([1], p. 541, eq.16.2.2).

Result 2

The gamma function, denoted $\Gamma(\alpha)$, is defined as

$$\Gamma(\alpha) = \int_0^\infty e^{-t} t^{\alpha-1} dt, \tag{57}$$

where Re $\alpha > 0$ ([10], p. 892, eq.8.310.1).

Remark: By splitting the range of the integral it is clear that the gamma function can be represented by (59) and (60):

$$\begin{aligned}\Gamma(\alpha) &= \int_0^x e^{-t}t^{\alpha-1}dt + \int_x^\infty e^{-t}t^{\alpha-1}dt \\ &= \gamma(\alpha,x) + \Gamma(\alpha,x). \end{aligned} \tag{58}$$

Result 3

The incomplete lower gamma function, denoted $\gamma(\alpha,x)$, is defined as

$$\gamma(\alpha,x) = \int_0^x e^{-t}t^{\alpha-1}dt, \tag{59}$$

where Re $\alpha > 0$ ([10], p. 899, eq.8.350.1).

Result 4

The incomplete upper gamma function, denoted $\Gamma(\alpha,y)$, is defined as

$$\Gamma(\alpha,x) = \int_x^\infty e^{-t}t^{\alpha-1}dt, \tag{60}$$

where Re $\alpha > 0$ ([10], p. 899, eq.8.350.2).

Result 5

The beta function, denoted $B(\alpha,\beta)$, is defined as

$$B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt \tag{61}$$

and

$$B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \tag{62}$$

where Re $\alpha > 0, \text{Re } \beta > 0$ ([10], p. 908, eq.9.380.1).

Result 6

The incomplete beta function, denoted $B(x;\alpha,\beta)$, is defined as

$$B(x;\alpha,\beta) = \int_0^x t^{\alpha-1}(1-t)^{\beta-1}dt \tag{63}$$

where $0 \leq x \leq 1, \text{Re}\ \alpha > 0, \text{Re}\ \beta > 0$ ([10], p. 910, eq.8.391).

Result 7

The incomplete beta function ratio, denoted $I_x(\alpha,\beta)$, is defined as

$$I_x(\alpha,\beta) = \frac{B(x;\alpha,\beta)}{B(\alpha,\beta)} \tag{64}$$

where $\text{Re}\ \alpha > 0, \text{Re}\ \beta > 0$ ([10], p. 910, eq.8.392).

Remark:

$$I_x(\alpha,\beta) = 1 - I_{1-x}(\beta,\alpha)$$

Proof: From the properties of an integral:

$$\begin{aligned}
1 - I_{1-x}(\beta,\alpha) &= 1 - \frac{B(1-x;\beta,\alpha)}{B(\beta,\alpha)} \\
&= \frac{B(\beta,\alpha) - B(1-x;\beta,\alpha)}{B(\beta,\alpha)} \\
&= \frac{\int_0^1 t^{\beta-1}(1-t)^{\alpha-1}dt - \int_0^{1-x} t^{\beta-1}(1-t)^{\alpha-1}dt}{B(\beta,\alpha)} \\
&= \frac{\int_{1-x}^1 t^{\alpha-1}(1-t)^{\beta-1}dt}{B(\beta,\alpha)}.
\end{aligned}$$

Consider the transformation $s = 1 - t$, which implies $t = 1 - s$ and $\frac{dt}{ds} = -1$,and the properties of an integral:

$$\begin{aligned}\frac{\int_{1-x}^{1} t^{\beta-1}(1-t)^{\alpha-1}dt}{B(\beta,\alpha)} &= \frac{-\int_{x}^{0} s^{\alpha-1}(1-s)^{\beta-1})ds}{B(\beta,\alpha)} \\ &= \frac{\int_{0}^{x} s^{\alpha-1}(1-s)^{\beta-1})ds}{B(\beta,\alpha)} \\ &= \frac{B(x;\alpha,\beta)}{B(\beta,\alpha)} \\ &= I_x(\alpha,\beta). \end{aligned} \tag{65}$$

Result 8

The Pochhammer coefficient is defined as

$$(\alpha)_i = \alpha(\alpha+1)\cdots(\alpha+i-1) = \tfrac{\Gamma(\alpha+i)}{\Gamma(\alpha)} \tag{66}$$

where $i = 1, 2, \ldots,(\alpha)_0 = 1$,Re $\alpha > 0, \text{Re } \alpha + i > 0$ and $\Gamma(\cdot)$ is the gamma function (*Appendix eq.*(57)). ([25], p. 164, eq.18:12:1)

Remark:

$$(1-\alpha)_i = \frac{\Gamma(\alpha)(-1)^i}{\Gamma(\alpha-i)} \tag{67}$$

where Re $\alpha > 0$, i an integer and $\Gamma(\cdot)$ is the gamma function (*Appendix eq.*(57)).

Proof: From equation (66):

$$\begin{aligned}\frac{\Gamma(\alpha)(-1)^i}{\Gamma(\alpha-i)} &= (-1)^i \frac{\Gamma((\alpha-i)+i)}{\Gamma(\alpha-i)} \\ &= (-1)^i (\alpha-i)_i \\ &= (-1)^i ((\alpha-1)-i+1)_i \end{aligned}$$

Since Pochhammer polynomials obey the reflection formula $(-x)_i = (-1)^i (x-i+1)_i$ ([25], p. 161, eq.18:5:1),

$$\begin{aligned}\frac{\Gamma(\alpha)(-1)^i}{\Gamma(\alpha-i)} &= (-1)^i ((\alpha-1)-i+1)_i \\ &= (-(\alpha-1))_i \\ &= (1-\alpha)_i. \end{aligned}$$

Result 9

$$\binom{i}{j} = \frac{\Gamma(i+1)}{\Gamma(i+1-j)j!} \tag{68}$$

where Re $i > 0$,$j > 0$ an integer and $\Gamma(\cdot)$ is the gamma function (*Appendix eq.*(57)).

Motivation: It is known that the binomial coefficient is defined as:

$$\binom{i}{j} = \frac{i!}{j!(i-j)!}.$$

Using the generalisation of the factorial function $n! = \Gamma(n+1)$ ([25], p. 25, eq.2:12:1),

$$\binom{i}{j} = \frac{\Gamma(i+1)}{\Gamma(i+1-j)j!}.$$

Result 10

The random variable X has a generated distribution if:

$$X = G^{-1}(F), \tag{69}$$

where $G^{-1}(\cdot)$ is the quantile function of the baseline distribution and F a random variable distributed as the generator distribution.

Proof: Let $X = G^{-1}(F)$ as defined above. Then for $H(\cdot)$ the CDF of the generated distribution and $F(\cdot)$ the CDF of the generator distribution, see Section 1.1 of report, we have:

$$\begin{aligned} H(x) &= P\left(X \leq x\right) \\ &= P\left(G^{-1}(F) \leq x\right) \\ &= P(F \leq G(x)) \\ &= F(G(x)), \end{aligned}$$

which satisfies the definition (2) of a generated distribution.

Result 11

$$H(x) = \frac{\gamma(\alpha, x)}{\Gamma(k)}, \tag{70}$$

where $x > 0, \mathrm{Re}\ \alpha > 0, \mathrm{Re}\ \beta > 0$ and $H(\cdot)$ the CDF of a $Gamma(k, 1)$ distribution.

Proof: The CDF, $H(\cdot)$, of a $Gamma(k, \theta)$ distribution is given by (73) and rewritten with (59):

$$\begin{aligned} H(x) &= \frac{1}{\Gamma(k)\theta^k} \int_0^x t^{k-1} e^{-\frac{x}{\theta}} dt \\ &= \frac{\gamma\left(\alpha, \frac{x}{\theta}\right)}{\Gamma(k)\theta^k}. \end{aligned}$$

Setting $\theta = 1$ proves the result.

Result 12

A continuous random variable X is said to have an exponential distribution with parameter k and $\theta > 0$ if it's PDF takes the form below:

$$f(x) = \begin{cases} \frac{1}{\theta} e^{-\frac{1}{\theta}} \text{ if } x > 0 \\ 0 \text{ if } x \leq 0 \end{cases} \tag{71}$$

([1], p. 115, eq.3.3.16).

Result 13

A continuous random variable X is said to have an gamma distribution with parameters $\theta > 0$ and $k > 0$ if it's PDF and PDF takes the form below:

$$f(x) = \begin{cases} \frac{1}{\theta^k \Gamma(\kappa)} x^{k-1} e^{-\frac{x}{\theta}} \text{ if } x > 0 \\ 0 \text{ if } x \leq 0 \end{cases} \tag{72}$$

and

$$F(x) = \begin{cases} \int_0^x \frac{1}{\theta^k \Gamma(\kappa)} t^{k-1} e^{-\frac{t}{\theta}} dt \text{ if } x > 0 \\ 0 \text{ if } x \leq 0 \end{cases} \tag{73}$$

([1], p. 111, eq.3.3.8 & 3.3.10).

Result 14

For a Weibull distribution with parameters k and λ distribution the r′th moment is given by [5]:

$$\mu'_r = \Gamma\left(r/k + 1\right)\lambda^r, r > 0 \tag{74}$$

where $\Gamma(\cdot)$ is the gamma function (*Appendix eq.*(57)).

Result 15

Mudholkar and Srivastava proposed the exponentiated Weibull distribution with PDF $h(x)$ and CDF $H(x)$ as [20]:

$$h(x) = \frac{vk}{\lambda^k}x^{k-1}\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right)^{v-1} \tag{75}$$

and

$$H(x) = \left(1-\exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\right)^v \tag{76}$$

where $0 \leq x < \infty, \lambda, k, v > 0$.

Result 16

For an for an exponentiated Weibull distribution with parameters k,v and λ distribution the r′th moment is given by [23]:

$$\mu'_r = v\lambda^r\Gamma\left(r/k\right)\sum_{j=0}^{\infty}\frac{(1-v)_j}{j!(j+1)^{(r+k)/k}}, r > -k \tag{77}$$

where $(\cdot)_i$ is the Pochhammer function (*Appendix eq.*(66)) and $\Gamma(\cdot)$ is the gamma function (*Appendix eq.*(57)). Note that the upper bound of the sum is $v-1$ for integer values of v, see (*Appendix eq.*(83)).

Result 17

For a $GN \sim (\mu, \sigma, s)$ distribution the the PDF and CDF of the standardised random variable $Z = (X - \mu)/\sigma$ is given by $\phi_s(z)$ and $\Phi(z)$ below [21]:

$$\phi_s(z) = \frac{s}{2\Gamma\left(1/s\right)} \exp\left\{-|z|^s\right\} \tag{78}$$

and

$$\begin{aligned} \Phi_s(z) &= \begin{cases} \dfrac{\Gamma\left(1/s, (-z)^s\right)}{2\Gamma\left(1/s\right)} \text{if } z \leq 0 \\ 1 - \dfrac{\Gamma\left(1/s, z^s\right)}{2\Gamma\left(1/s\right)} \text{if } z > 0 \end{cases} \\ & -\infty < z < \infty, s > 0. \end{aligned} \tag{79}$$

Proof: From (11) the transformation $z = {}^{(x-\mu)}/_{\sigma}$, which implies $x = \sigma z + \mu$ and $\frac{dx}{dz} = \sigma$, it follows that:

$$\begin{aligned} g(x) &= \frac{s}{2\sigma\Gamma\left(1/s\right)} \exp\left\{-\left|\frac{x-\mu}{\sigma}\right|^s\right\} \\ \therefore \phi_s(z) &= \frac{s}{2\Gamma\left(1/s\right)} \exp\{-|z|^s\} \end{aligned}$$

and

$$\begin{aligned} \therefore G(x) &= \begin{cases} \dfrac{s\int_{-\infty}^{x} \exp\left\{-\left(\frac{\mu-y}{\sigma}\right)^s\right\} dy}{2\sigma\Gamma\left(1/s\right)} \text{if } x \leq \mu \\ 1 - \dfrac{s\int_{x}^{\infty} \exp\left\{-\left(\frac{y-\mu}{\sigma}\right)^s\right\} dy}{2\sigma\Gamma\left(1/s\right)} \text{if } x > \mu \end{cases} \\ &= \begin{cases} \dfrac{s\int_{-\infty}^{x} \exp\left\{-\left(-z\right)^s\right\} dz}{2\sigma\Gamma\left(1/s\right)} \text{if } z \leq 0 \\ 1 - \dfrac{s\int_{x}^{\infty} \exp\left\{-z^s\right\} dz}{2\sigma\Gamma\left(1/s\right)} \text{if } z > 0 \end{cases}. \end{aligned}$$

Lastly from the transformation $k = z^s$, which implies $z = k^{\frac{1}{s}}$ and $\frac{dx}{dz} = \frac{1}{s}k^{\frac{1}{s}-1}$, and symmetry it follows that:

$$
\begin{aligned}
G(x) &= \begin{cases} \dfrac{\int_x^\infty k^{\frac{1}{s}-1}\exp\{k\}\,dk}{2\Gamma\left(1/s\right)} \text{if } k \leq 0 \\ 1 - \dfrac{\int_x^\infty k^{\frac{1}{s}-1}\exp\{k\}\,dk}{2\Gamma\left(1/s\right)} \text{if } k > 0 \end{cases} \\
\therefore \Phi_s(z) &= \begin{cases} \dfrac{\Gamma(1/s, (-z)^s)}{2\Gamma\left(1/s\right)} \text{if } z \leq 0 \\ 1 - \dfrac{\Gamma(1/s, z^s)}{2\Gamma\left(1/s\right)} \text{if } z > 0 \end{cases}
\end{aligned}
$$

and where $\Gamma(\cdot,\cdot)$ is the upper incomplete gamma function, see (*Appendix eq.*(60)).

Result 18

For a standardised generalised normal distribution, $Z = (X-\mu)/\sigma$ where $X \sim GN(\mu, \sigma, s)$, the following properties hold:

$$\Phi_s(-z) = 1-\Phi_s(z) \tag{80}$$

and

$$\phi_s(-z) = \phi_s(z) \tag{81}$$

where $\Phi_s(\cdot)$ and $\phi_s(\cdot)$ are defined in (*Appendix eq.*(78)&(79)).

Proof: These results follow directly from the symmetry of the PDF, see (9).

Result 19

$$(1+x)^q = \sum_{k=0}^{\infty} \binom{q}{k} x^k \tag{82}$$

where $q \neq 0$ not a natural number and $|x| < 1$ ([10], p. 25, eq.1.110).

Result 20

$$(1+x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k \tag{83}$$

where n is a natural number ([10], p. 25, eq.1.111).

Result 21

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} \tag{84}$$

where n is a natural number ([10], p. 26, eq.1.121.1).

Result 22

If Re $\alpha > 0$ non-integer, then

$$\Phi_s(z)^\alpha = \sum_{i=0}^{\infty}\sum_{j=i}^{\infty} \binom{\alpha}{j}\binom{j}{i}(-1)^{i+j}\Phi_s(z)^i. \tag{85}$$

[3].

Proof: If Re $\alpha > 0$ non-integer, then using (*Appendix eq.*(82)) successively:

$$\begin{aligned}
\Phi_s(z)^\alpha &= (1-(1-\Phi_s(z)))^\alpha \\
&= \sum_{i=0}^{\infty}\binom{\alpha}{i}(-1)^i(1-\Phi_s(z))^i \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{i}\binom{\alpha}{i}\binom{i}{j}(-1)^{i+j}\Phi_s(z)^j \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{i} a_{i,j}\Phi_s(z)^j
\end{aligned}$$

where $a_{i,j} = \binom{\alpha}{j}\binom{j}{i}(-1)^{i+j}$.

Expanding the sum and re-indexing into new sums:

$$\begin{aligned}
\sum_{i=0}^{\infty}\sum_{j=0}^{i} a_{i,j}\Phi_s(z)^j &= a_{0,0} \\
&\quad +a_{1,0} + a_{1,1}\Phi_s(z) \\
&\quad +a_{2,0} + a_{2,1}\Phi_s(z) + a_{2,2}\Phi_s(z)^2 \\
&\quad +a_{3,0} + a_{3,1}\Phi_s(z) + a_{3,2}\Phi(z)^2 + a_{3,3}\Phi_s(z)^3 \\
&\quad \dots
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=0}^{\infty} a_{i,0}\Phi_s(z)^0 + \sum_{i=1}^{\infty} a_{i,1}\Phi_s(z)^1 + \sum_{i=2}^{\infty} a_{i,2}\Phi_s(z)^2 + \sum_{i=3}^{\infty} a_{i,3}\Phi_s(z)^3 + \sum_{i=4}^{\infty} a_{i,4}\Phi_s(z)^4 + \ldots \\
&= \sum_{j=0}^{\infty}\sum_{i=j}^{\infty} a_{i,j}\Phi(z)^j.
\end{aligned}
$$

Lastly substituting $a_{i,j} = \binom{\alpha}{j}\binom{j}{i}(-1)^{i+j}$ into the previous step the result is proven:

$$
\Phi_s(z)^{\alpha} = \sum_{j=0}^{\infty}\sum_{i=j}^{\infty}\binom{\alpha}{i}\binom{i}{j}(-1)^{i+j}\Phi(z)^j.
$$

Result 23

$$
\left[\sum_{i=0}^{\infty}\frac{(-1)^i}{(1/s+i)i!}y^i\right]^j = \sum_{i=0}^{\infty} c_{i,j}y^i, \tag{86}
$$

where $c_{0,j} = s^j$ and $c_{i,j} = 1/is\sum_{k=1}^{i}(kj-i+k)(^{(-1)^k}/_{(1/s+k)k!})c_{i-k,j}$ for all $i \geq 1$ and j a natural number [3].

Proof: This result follows directly from ([10], p. 17, eq.0.314) with $a_k = {}^{(-1)^k}/_{(1/s+k)k!}$.

Result 24

$$
\begin{aligned}
Q_{i,j}^{(s)} &= \int_0^{\infty} z^i\Phi_s(z)^j\phi(z)dz \\
&= \frac{1}{(2\Gamma(1/s))^{j+1}}\sum_{k=0}^{j}\binom{j}{k}[\Gamma(1/s)]^{j-k}\sum_{l=0}^{\infty}c_{l,k}\Gamma\left(l+\frac{i+k+1}{s}\right)
\end{aligned} \tag{87}
$$

where $c_{0,k} = s^k$, $c_{l,k} = 1/ls\sum_{r=1}^{l}(rk-l+r)\,(^{(-1)^r}/_{(1/s+r)r!})\,c_{l-r,k}$ for all $l \geq 1$ and $\Gamma(\cdot)$ is the gamma function (*Appendix eq.*(57)) [3].

Proof: Define the more general quantity:

$$
Q_{i,j}^{(s)}(a,b) = \int_a^b z^i\Phi_s(z)^j\phi(z)dz, 0 < a < b
$$

Thus $Q_{i,j}^{(s)} \equiv Q_{i,j}^{(s)}(0,\infty)$.

From the definition of $\Phi_s(\cdot)$ (*Methodology eq.*(11)) and the transformation $y = z^s$, which implies $z = y^{\frac{1}{s}}$ and $\frac{dz}{dy} = \frac{1}{s}$, we have that:

$$
\begin{aligned}
Q_{i,j}^{(s)}(a,b) &= \int_a^b z^i \Phi_s(z)^j \phi(z) dz \\
&= \int_a^b z^i \left(1 - \frac{\Gamma(1/s, z^s)}{2\Gamma\left(1/s\right)}\right)^j \left(\frac{s}{2\Gamma\left(1/s\right)} \exp\left\{-z^s\right\}\right) dz \\
&= \int_a^b z^i \left(\frac{2\Gamma\left(1/s\right) - \Gamma(1/s, z^s)}{2\Gamma\left(1/s\right)}\right)^j \left(\frac{s}{2\Gamma\left(1/s\right)} \exp\left\{-z^s\right\}\right) dz \\
&= \frac{s}{\left(2\Gamma\left(1/s\right)\right)^{j+1}} \int_a^b z^i \left(2\Gamma\left(1/s\right) - \Gamma(1/s, z^s)\right)^j \exp\left\{-z^s\right\} dz \\
&= \frac{1}{\left(2\Gamma\left(1/s\right)\right)^{j+1}} \int_a^b y^{\frac{i+1}{s}-1} \left(2\Gamma\left(1/s\right) - \Gamma(1/s, y)\right)^j \exp\{-y\} dy
\end{aligned}
$$

where $\Gamma(\cdot,\cdot)$ is the upper incomplete gamma function (*Appendix eq.*(60)).

As shown by [22] the lower incomplete gamma function, $\gamma(\cdot,\cdot)$ (*Appendix eq.*(59)), admits the power series expansion $\gamma(a,x) = x^a \sum_{m=0}^{\infty} {}^{(-x)^m}/_{(a+m)m!)}$ and therefore also admits $\Gamma(a,x) = \Gamma(a) - \gamma(a,x) = \Gamma(a) - x^a \sum_{m=0}^{\infty} {}^{(-x)^m}/_{((a+m)m!)}$. Using the latter and (*Appendix eq.*(83)):

$$
\begin{aligned}
Q_{i,j}^{(s)}(a,b) &= \frac{1}{\left(2\Gamma\left(1/s\right)\right)^{j+1}} \int_a^b y^{i+1/s-1} \left(\Gamma\left(1/s\right) + \gamma\left(1/s, y\right)\right)^j \exp\{-y\} dy \\
&= \frac{1}{\left(2\Gamma\left(1/s\right)\right)^{j+1}} \int_a^b y^{i+1/s-1} \sum_{k=0}^{j} \binom{j}{k} \left(\gamma\left(1/s, y\right)\right)^k \left(\Gamma\left(1/s\right)\right)^{j-k} \exp\{-y\} dy \\
&= \frac{1}{\left(2\Gamma\left(1/s\right)\right)^{j+1}} \sum_{k=0}^{j} \binom{j}{k} \Gamma\left(1/s\right)^{j-k} \int_a^b y^{\frac{i+1}{s}-1} \gamma\left(1/s, y\right)^k \exp\{-y\} dy \\
&= \frac{1}{\left(2\Gamma\left(1/s\right)\right)^{j+1}} \sum_{k=0}^{j} \binom{j}{k} \Gamma\left(1/s\right)^{j-k} \int_a^b y^{\frac{i+1}{s}-1} \\
&\quad \cdot \left(y^{\frac{1}{s}} \sum_{m=0}^{\infty} {}^{(-y)^m}/_{(1/s+m)m!)}\right)^k \exp\{-y\} dy \\
&= \frac{1}{\left(2\Gamma\left(1/s\right)\right)^{j+1}} \sum_{k=0}^{j} \binom{j}{k} \Gamma\left(1/s\right)^{j-k} \int_a^b y^{\frac{i+k+1}{s}-1} \\
&\quad \cdot \left(\sum_{m=0}^{\infty} {}^{(-y)^m}/_{(1/s+m)m!)}\right)^k \exp\{-y\} dy
\end{aligned} \tag{88}
$$

Let $c_{0,k} = s^k$ and $c_{m,k} = {}^1/_{ms} \sum_{l=1}^{m} (lk - m + l) \left({}^{(-1)^l}/_{(1/s+l)l!}\right) c_{m-l,k}$ for all $m \geq 1$.

Then from (*Appendix eq.*(86)) the power series in the integral (88) is re-written:

$$\begin{aligned}
Q_{i,j}^{(s)}(a,b) &= \frac{1}{\left(2\Gamma\left(1/s\right)\right)^{j+1}}\sum_{k=0}^{j}\binom{j}{k}\Gamma\left(1/s\right)^{j-k} \\
&\quad \cdot\int_a^b y^{\frac{i+k+1}{s}-1}\left(\sum_{m=0}^{\infty}{}^{(-y)^m}/_{(1/s+m)m!}\right)^k\exp\{-y\}dy \\
&= \frac{1}{\left(2\Gamma\left(1/s\right)\right)^{j+1}}\sum_{k=0}^{j}\binom{j}{k}\Gamma\left(1/s\right)^{j-k}\int_a^b y^{\frac{i+k+1}{s}-1}\sum_{m=0}^{\infty}c_{m,k}y^m\exp\{-y\}dy \\
&= \frac{1}{\left(2\Gamma\left(1/s\right)\right)^{j+1}}\sum_{k=0}^{j}\binom{j}{k}\Gamma\left(1/s\right)^{j-k}\sum_{m=0}^{\infty}c_{m,k}\int_a^b y^{m+\frac{i+k+1}{s}-1}\exp\{-y\}dy \\
&= \frac{1}{\left(2\Gamma\left(1/s\right)\right)^{j+1}}\sum_{k=0}^{j}\binom{j}{k}\Gamma\left(1/s\right)^{j-k} \\
&\quad \cdot\sum_{m=0}^{\infty}c_{m,k}\left[\Gamma\left(m+\frac{i+k+1}{s},b\right)-\Gamma\left(m+\frac{i+k+1}{s},a\right)\right]
\end{aligned}$$

where where $\Gamma(\cdot,\cdot)$ is the upper incomplete gamma function, see (*Appendix eq.*(60)).

Letting $a=0$ and $b=\infty$ proves the result:

$$Q_{i,j}^{(s)} = \frac{1}{\left(2\Gamma\left(1/s\right)\right)^{j+1}}\sum_{k=0}^{j}\binom{j}{k}\Gamma\left(1/s\right)^{j-k}\sum_{m=0}^{\infty}c_{m,k}\Gamma\left(m+{}^{i+k+1}/_{s}\right).$$

Result 25

$$\begin{aligned}
M_j^{(s)} &= \int_0^{\infty}e^{t(\sigma z+\mu)}\Phi_s(z)^j\phi_s(z)dz \\
&= \frac{e^{t\mu}}{2\Gamma\left(1/s\right)}\sum_{k=0}^{j}\binom{j}{k}\Gamma\left(1/s\right)^{j-k}\sum_{m=0}^{\infty}c_{m,k}\sum_{p=0}^{\infty}\frac{t\sigma^p}{p!}\Gamma\left(m+\frac{i+k+p+1}{s}\right) \qquad (89)
\end{aligned}$$

$$\begin{aligned}
M_j^{(s,-)} &= \int_0^{\infty}e^{t(-\sigma z+\mu)}\Phi_s(z)^j\phi_s(z)dz \\
&= \frac{e^{t\mu}}{2\Gamma\left(1/s\right)}\sum_{k=0}^{j}\binom{j}{k}\Gamma\left(1/s\right)^{j-k}\sum_{m=0}^{\infty}c_{m,k}\sum_{p=0}^{\infty}\frac{(-t\sigma)^p}{p!}\Gamma\left(m+\frac{i+k+p+1}{s}\right) \qquad (90)
\end{aligned}$$

where $c_{0,k}=s^k$and $c_{l,k}={}^1/_{ls}\sum_{r=1}^{l}(rk-l+r)\left({}^{(-1)^k}/_{(1/s+r)r!}\right)c_{l-r,k}$ for all $l\geq 1$.

Proof: Define the more general quantities:

$$M_j^{(s)}(a,b) \quad = \quad \int_a^b e^{t(\sigma z+\mu)}\Phi_s(z)^j\phi_s(z)dz$$

and

$$M_j^{(s,-)}(a,b) \quad = \quad \int_a^b e^{t(-\sigma z+\mu)}\Phi_s(z)^j\phi_s(z)dz, 0<a<b.$$

Thus $M_{i,j}^{(s)} \equiv M_{i,j}^{(s)}(0,\infty)$ and $M_{i,j}^{(s,-)} \equiv M_{i,j}^{(s,-)}(0,\infty)$.

From the definition of $\Phi_s(\cdot)$ (*Methodology eq.*(11)) and the transformation $y = z^s$, which implies $z = y^{\frac{1}{s}}$ and $\frac{dz}{dy} = \frac{1}{s}$, we have that:

$$\begin{aligned}
M_j^{(s)}(a,b) &= \int_a^b e^{t(\sigma z+\mu)}\Phi_s(z)^j\phi_s(z)dz \\
&= \int_a^b e^{t(\sigma z+\mu)}\left(1-\frac{\Gamma(1/s,z^s)}{2\Gamma\left(1/s\right)}\right)^j\left(\frac{s}{2\Gamma\left(1/s\right)}\exp\left\{-|z|^s\right\}\right)dz \\
&= e^{t\mu}\int_a^b\left(\frac{2\Gamma\left(1/s\right)-\Gamma(1/s,z^s)}{2\Gamma\left(1/s\right)}\right)^j\left(\frac{s}{2\Gamma\left(1/s\right)}\exp\left\{t\sigma z-z^s\right\}\right)dz \\
&= \frac{se^{t\mu}}{2\Gamma\left(1/s\right)^{j+1}}\int_a^b\left(2\Gamma\left(1/s\right)-\Gamma(1/s,z^s)\right)^j\left(\exp\left\{t\sigma z-z^s\right\}\right)dz \\
&= \frac{e^{t\mu}}{2\Gamma\left(1/s\right)^{j+1}}\int_a^b y^{\frac{1}{s}-1}\left(2\Gamma\left(1/s\right)-\Gamma(1/s,y)\right)^j\exp\left\{t\sigma y^{\frac{1}{s}}-y\right\}dy
\end{aligned}$$

where $\Gamma(\cdot,\cdot)$ is the upper incomplete gamma function (*Appendix eq.*(60)).

As shown by [22] the lower incomplete gamma function, $\gamma(\cdot,\cdot)$ (*Appendix eq.*(59)), admits the power series expansion $\gamma(a,x) = x^a\sum_{m=0}^{\infty}{}^{(-x)^m}/_{(a+m)m!)}$ and therefore also admits $\Gamma(a,x) = \Gamma(a)-\gamma(a,x) = \Gamma(a)-x^a\sum_{m=0}^{\infty}{}^{(-x)^m}/_{((a+m)m!)}$. Using the latter and (*Appendix eq.*(83)):

$$\begin{aligned}
M_j^{(s)}(a,b) &= \frac{e^{t\mu}}{2\Gamma\left(1/s\right)^{j+1}}\int_a^b y^{\frac{1}{s}-1}\exp\{t\sigma y^{\frac{1}{s}}-y\}\left(\Gamma\left(1/s\right)+\gamma\left(1/s,y\right)\right)^j dy \\
&= \frac{e^{t\mu}}{2\Gamma\left(1/s\right)^{j+1}}\int_a^b y^{\frac{1}{s}-1}\exp\{t\sigma y^{\frac{1}{s}}-y\} \\
&\quad \cdot\sum_{k=0}^{j}\binom{j}{k}\gamma\left(1/s,y\right)^k\Gamma\left(1/s\right)^{j-k}dy
\end{aligned}$$

$$= \frac{e^{t\mu}}{2\Gamma\left(1/s\right)^{j+1}} \sum_{k=0}^{j} \binom{j}{k} \Gamma\left(1/s\right)^{j-k} \cdot \int_a^b y^{\frac{1}{s}-1} \exp\left\{t\sigma y^{\frac{1}{s}} - y\right\} \gamma\left(1/s, y\right)^k dy, \tag{91}$$

where $\gamma(\cdot,\cdot)$ is the lower incomplete gamma function (*Appendix eq.*(59)).

Let $c_{0,k} = s^k$ and $c_{m,k} = 1/ms \sum_{l=1}^{m} (lk - m + l)((-1)^l/(1/s+l)l!) c_{m-l,k}$ for all $m \geq 1$. Then from (*Appendix eq.*(86)) and (*Appendix eq.*(84)) the integral (91) is re-written:

$$\begin{aligned}
M_j^{(s)}(a,b) &= \frac{e^{t\mu}}{2\Gamma\left(1/s\right)^{j+1}} \sum_{k=0}^{j} \binom{j}{k} \Gamma\left(1/s\right)^{j-k} \\
&\quad \cdot \int_a^b y^{\frac{1}{s}-1} \exp\left\{t\sigma y^{\frac{1}{s}} - y\right\} \left(y^{\frac{1}{s}} \sum_{m=0}^{\infty} (-y)^m/(1/s+m)m! \right)^k dy \\
&= \frac{e^{t\mu}}{2\Gamma\left(1/s\right)^{j+1}} \sum_{k=0}^{j} \binom{j}{k} \Gamma\left(1/s\right)^{j-k} \\
&\quad \cdot \int_a^b y^{\frac{1+k}{s}-1} \exp\left\{t\sigma y^{\frac{1}{s}} - y\right\} \sum_{m=0}^{\infty} c_{m,k} y^m dy \\
&= \frac{e^{t\mu}}{2\Gamma\left(1/s\right)} \sum_{k=0}^{j} \binom{j}{k} \Gamma\left(1/s\right)^{j-k} \sum_{m=0}^{\infty} c_{m,k} \int_a^b y^{m+\frac{1+k}{s}-1} \exp\left\{t\sigma y^{\frac{1}{s}} - y\right\} dy \\
&= \frac{e^{t\mu}}{2\Gamma\left(1/s\right)} \sum_{k=0}^{j} \binom{j}{k} \Gamma\left(1/s\right)^{j-k} \sum_{m=0}^{\infty} c_{m,k} \int_a^b \sum_{p=0}^{\infty} \frac{\left(t\sigma y^{\frac{1}{s}}\right)^p}{p} y^{m+\frac{1+k}{s}-1} \exp\left\{-y\right\} dy \\
&= \frac{e^{t\mu}}{2\Gamma\left(1/s\right)} \sum_{k=0}^{j} \binom{j}{k} \Gamma\left(1/s\right)^{j-k} \sum_{m=0}^{\infty} c_{m,k} \sum_{p=0}^{\infty} \frac{t\sigma}{p!} \int_a^b y^{m+\frac{1+k+p}{s}-1} \exp\left\{-y\right\} dy \\
&= \frac{e^{t\mu}}{2\Gamma\left(1/s\right)} \sum_{k=0}^{j} \binom{j}{k} \Gamma\left(1/s\right)^{j-k} \sum_{m=0}^{\infty} c_{m,k} \sum_{p=0}^{\infty} \frac{t\sigma}{p!} \left[\Gamma\left(m + \frac{i+k+p+1}{s}, b\right) - \Gamma\left(m + \frac{i+k+p+1}{s}, a\right) \right]
\end{aligned}$$

where where $\Gamma(\cdot,\cdot)$ is the upper incomplete gamma function, see (*Appendix eq.*(60)).

Letting $a = 0$ and $b = \infty$ proves the result (89):

$$M_j^{(s,-)} = \frac{e^{t\mu}}{2\Gamma\left(1/s\right)} \sum_{k=0}^{j} \binom{j}{k} \Gamma\left(1/s\right)^{j-k} \sum_{m=0}^{\infty} c_{m,k} \sum_{p=0}^{\infty} \frac{t\sigma}{p!} \Gamma\left(m + \frac{i+k+p+1}{s}\right)$$

Similarly it follows for result (90) that:

$$M_j^{(s,-)} = \frac{e^{t\mu}}{2\Gamma\left(1/s\right)} \sum_{k=0}^{j} \binom{j}{k} \Gamma\left(1/s\right)^{j-k} \sum_{m=0}^{\infty} c_{m,k} \sum_{p=0}^{\infty} \frac{(-t\sigma)^p}{p!} \Gamma\left(m + \frac{i+k+p+1}{s}\right).$$

Tables

Tables 7, 8 & 9 contain most of the beta and Kumaraswamy generated distributions, the surnames of the relevant authors and the year of publication. The distributions are organised alphabetically according to their generator distribution and then according to the year of study.

Generator	Baseline	Authors	Year
Beta	Normal	Eugene et al.	2002
Beta	Hyperbolic Secant	Fischer and Vaughan	2004
Beta	Gumbel	Nadarajah and Kotz	2004
Beta	Exponential	Nadarajah and Kotz	2006
Beta	Gamma	Kong, Carl and Sepanski	2007
Beta	Laplace	Kozubowski and Nadarajah (first) & Cordeiro and Lemonte (2011)	2008
Beta	Pareto	Akinsete, Famoye and Lee	2008
Beta	Generalized exponential	Barreto-Souza, Santos and Cordeiro	2010
Beta	Modified Weibull	Silva, Ortega and Cordeiro	2010
Beta	Birnbaum-Saunders	Cordeiro and Lemonte	2011
Beta	Burr XII	Paranaba, Ortega, Cordeiro and Pescim	2011
Beta	Frechet	Barreto-Souza, Cordeiro and Simas	2011
Beta	Generalized Pareto	Mahmoudi & Nassar and Nada (separately)	2011
Beta	Half-Cauchy	Cordeiro and Lemonte	2011
Beta	Exponential geometric	Bidram & Nassar and Nada (separately)	2012
Beta	Exponentiated Pareto	Zea, Silva, Bourguignon, Santos and Cordeiro	2012
Beta	Generalized Weibull	Singla, Jain and Sharma	2012
Beta	Power	Cordeiro and Brito	2012
Beta	Burr III	Gomes, da Silva, Cordeiro and Ortega	2013
Beta	Dagum	Domma and Condino	2013
Beta	Exponentiated Weibull	Cordeiro, Gomes, da Silva and Ortega	2013
Beta	Generalized gamma	Cordeiro, Castellares, Montenegro and de Castro	2013
Beta	Generalized half normal geometric	Ramires, Ortega, Cordeiro and Hamedani	2013
Beta	Generalized logistic	Morais, Cordeiro and Cysneiros	2013
Beta	Generalized Rayleigh	Cordeiro, Cristino, Hashimoto and Ortega	2013
Beta	Inverse Weibull	Hanook, Shahbaz, Mohsin, and Golam Kibria	2013
Beta	Lognormal	Montenegro and Cordeiro	2013
Beta	Lomax	Rajab, Aleem, Nawaz and Daniya	2013
Beta	Nakagami	Shittu and Adepoju	2013
Beta	Skew normal	Mameli and Musio	2013
Beta	Weibull geometric	Cordeir,o Silva and Ortega & (separately) Bidram, Behboodian and Towhidi	2013
Beta	Weibull Poisson	Percontini, Blas and Cordeiro	2013
Beta	Weighted Weibull	Idowu and Ikegwu & (separately) Badmus and Bamiduro (2014)	2013
Beta	Cauchy	Alshawarbeh, Famoye and Lee	2014
Beta	Leneralized Lindley	Oluyede and Yang	2014
Beta	Generalized normal	Cintra, Rego, Cordeiro and Nascimento	2014

Table 7: List of Kumaraswamy and beta generated distributions.

Generator	Baseline	Authors	Year
Beta	Gompertz	Jafari, Tahmasebi and Alizadeh	2014
Beta	Inverse Rayleigh	Leao, Saulo, Bourguignon, Cintra, Rego and Cordeiro	2014
Beta	Linear failure rate	Jafari and Mahmoudi	2014
Beta	Lindley	Merovci and Sharma	2014
Beta	Moyal	Cordeiro, Nobre, Pescim and Ortega	2014
Beta	Power exponential	Adepoju, Chukwu and Wang	2014
Beta	Transmuted Weibull	Pal and Tiensuwan	2014
Beta	Truncated Pareto	Lourenzutti, Duarte and Azevedo	2014
Beta	Odd log-logistic generalized family	Cordeiro, Alizadeh, Tahir, Mansoor, Bourgiognon Hamedani	2015
Beta	Nadarajah Haghighi	Dias, Alizadeh and Cordeiro	2016
Kumaraswamy	Weibull	Cordeiro, Ortega	2010
Kumaraswamy	Ggeneralized gamma	de Pascoa, Ortega and Cordeiro	2011
Kumaraswamy	Birnbaum-Saunders	Saulo, Leao and Bourguignon	2012
Kumaraswamy	Generalized half normal	Cordeiro, Pescim and Ortega	2012
Kumaraswamy	Gumbel	Cordeiro, Nadarajah and Ortega	2012
Kumaraswamy	Inverse Weibull	Shahbaz, Shahbaz and Butt	2012
Kumaraswamy	Log-logistic	de Santana, Ortega, Cordeiro and Silva	2012
Kumaraswamy	Burr XII	Paranaiba, Ortega, Cordeiro and de Pascoa	2013
Kumaraswamy	Exponentiated Pareto	Elbatal	2013
Kumaraswamy	Generalized exponenti-ated Pareto	Shams	2013
Kumaraswamy	Generalized linear failure rate	Elbatal	2013
Kumaraswamy	Generalized Lomax	Shams	2013
Kumaraswamy	Generalized Pareto	Nadarajah and Eljabri	2013
Kumaraswamy	Pareto	Bourguignon, Silva, Zea and Cordeiro	2013
Kumaraswamy	Quasi Lindley	Elbatal and Elgarhy	2013
Kumaraswamy	Generalized Rayleigh	Gomes, da Silva, Cordeiro and Ortega	2014
Kumaraswamy	Geometric	Akinsete, Famoye and Lee	2014
Kumaraswamy	Half-Cauchy	Ghosh	2014
Kumaraswamy	Inverse exponential	Oguntunde, Babatunde and Ogunmola	2014
Kumaraswamy	Inverse Rayleigh	Roges, de Gusmao and Diniz	2014
Kumaraswamy	Kumaraswamy	El-Sherpieny and Ahme	2014
Kumaraswamy	Lindley	Cakmakyapan and Kadilar	2014
Kumaraswamy	Modified Weibull	Cordeiro, Ortega and Silva	2014
Kumaraswamy	Modified inverse Weibull	Aryal and Elbata	2015
Kumaraswamy	Odd log-logistic	Alizadeh, Emandi, Doostparast, Cordeiro and Pescim	2015
Kumaraswamy	Exponentiated Fretchet	Diab and Elbatal	2016

Table 8: Continued list of Kumaraswamy and beta generated distributions.

Generator	Baseline	Authors	Year
Kumaraswamy	Gompertz Makeham	Chukwu and Ogunde	2016
Kumaraswamy	Marshall-Olkin Fretchet	Afify, Yousof, Cordeiro and Ahmad	2016
Kumaraswamy	Exponentiated Inverse Rayleigh	Haq	2016
Kumaraswamy	Generalized Power Weibull	Selim and Badr	2016
Kumaraswamy	Kumaraswamy Fisher Snedecor	Adepoju and Chukwu	2016
Kumaraswamy	Trans. exp. additive Weibull	Nofal, Afify, Yuosof, Granzotto and Louzada	2016
Kumaraswamy	Exponentiated Rayleigh	Rashwan	2016
Kumaraswamy	Laplace	Nassar	2016
Kumaraswamy	Extended Inverse Weibull	Kumar and Nair	2016
Kumaraswamy	Marshall-Olkin	George and Thobias	2017

Table 9: Continued list of Kumaraswamy and beta generated distributions.

Programs

Listing 1 contains a user defined SAS software, a proc iml call function, for simulating values from a $X \sim BGN(\mu, \sigma, s, \alpha, \beta)$ distribuion. Listing 2 implements user defined R software functions, using packages [9] and [31], for the PDFs; CDFs; estimation of distribution parameters by MLE and KS estimation; and simulation of the BW, KW, BGN and KGN distributions. The Listings 3 & 4 contain programs of the applications in Section 6.2. The application programs function by calling the user defined functions in Listing 2 and uses the package [30]. Listing 5 contains Mathematica software [11] with functions of the PDFs; CDFs; and hazard functions of the BW, KW, BGN and KGN distributions. These functions were used to plot the graphs in this study.

Listing 1: Beta Generalised Normal RGN

```
proc iml;

start rbgn(mu,sigma,s,alpha,beta,n);
        x=J(n,1,0);
        f=rand('beta',J(n,1,alpha),J(n,1,beta));

        ind=loc(f<0.5);
        x[ind]=mu-sigma*quantile('gamma',(1-2*f[ind]),(1/s),1)##(1/s);

        ind=loc(f>=0.5);
        x[ind]=mu+sigma*quantile('gamma',(2*f[ind]-1),(1/s),1)##(1/s);
        return(x);
finish;

n=5000000;
call randseed(123,1);

x= rbgn(104.938,331.574,0.250,285.582,248.876,n);

theo=2158.01||7.32334*10**6||3.13892*10**10||1.62197*10**14;
mom=mean(x||x##2||x##3||x##4);
```

```
diff=theo-mom;

skew=skewness(x);
kurt=kurtosis(x);

print theo, mom, diff, skew, kurt;

create test from x[colname={x}];
append from x;
quit;

proc sgplot data=test;
*       title "Simulated Distribution";
        histogram x;
        density x / type=kernel;
*       keylegend / location=inside position=topright;
run;
```

Listing 2: User Defined R Functions

```
library(zipfR)
library(VGAM)
#############################################################################
#Beta-Weibull density function

dBW_func <- function(x,parameters)
{
  k      <-parameters[1]
  lambda<-parameters[2]
  alpha <-parameters[3]
  beta1 <-parameters[4]
if(k>0 & lambda>0 & alpha>0 & beta1>0)
    {
      T1 <- 1/(beta(alpha,beta1))*(k/lambda^k)*x^(k-1)
      T2 <- (1-exp(-(x/lambda)^k))^(alpha-1)
      T3 <- exp(-beta1*(x/lambda)^k)
      d  <- T1*T2*T3
    }
  else
    {
    d<-rep(0,length(x))
    }

  return(d)
}
#############################################################################
#Beta-Weibull cumulative density function
cdfBW_func <- function(x,parameters)
{
  k      <-parameters[1]
  lambda<-parameters[2]
  alpha <-parameters[3]
  beta1 <-parameters[4]
```

```
  if(k>0 & lambda>0 & alpha>0 & beta1>0)
  {
    GX <- 1-exp(-(x/lambda)^k)
    T1 <- 1/(beta(alpha,beta1))
    cd  <- T1*Ibeta(GX,alpha,beta1)
  }
  else
  {
    cd<-rep(0,length(x))
  }

  return(cd)
}

#############################################################################
#Kumaraswamy Weibull density function

dKW_func <- function(x,parameters)
{
  k      <-parameters[1]
  lambda<-parameters[2]
  alpha <-parameters[3]
  beta1 <-parameters[4]
  if(k>0 & lambda>0 & alpha>0 & beta1>0)
  {
    T1 <- alpha*beta1*k/(lambda^k)*x^(k-1)*exp(-(x/lambda)^k)
    T2 <- (1-exp(-(x/lambda)^k))^(alpha-1)
    T3 <- (1-(1-exp(-(x/lambda)^k))^alpha)^(beta1-1)
    d  <- T1*T2*T3
  }
  else
  {
    d<-rep(0,length(x))
  }
```

```
  return(d)
}

#########################################################################
#Kumaraswamy-Weibull cumulative density function
cdfKW_func <- function(x,parameters)
{
  k      <-parameters[1]
  lambda<-parameters[2]
  alpha <-parameters[3]
  beta1 <-parameters[4]
  if(k>0 & lambda>0 & alpha>0 & beta1>0)
  {
    cd  <- 1-(1-(1-exp(-(x/lambda)^k))^alpha)^beta1
  }
  else
  {
    cd<-rep(0,length(x))
  }

  return(cd)
}

#########################################################################
#Beta-Generalised normal density function

dBGN_func <- function(x,parameters)
{
  mu     <-parameters[1]
  sigma <-parameters[2]
  s      <-parameters[3]
  alpha <-parameters[4]
  beta1 <-parameters[5]
```

```
  z       <-(x-mu)/sigma
  if(sigma>0 & s>0 & alpha>0 & beta1>0)
  {
    T1 <- 1/(sigma*beta(alpha,beta1))
    T2 <- GNPhi_func(z,s)^(alpha-1)
    T3 <- (1-GNPhi_func(z,s))^(beta1-1)
    T4 <-GNphi_func(z,s)
    d  <- T1*T2*T3*T4
  }
  else
  {
    d<-rep(0,length(x))
  }

  return(d)
}

#########################################################################
#Beta-Generalised normal cumulative density function

cdfBGN_func <- function(x,parameters)
{
  mu    <-parameters[1]
  sigma <-parameters[2]
  s     <-parameters[3]
  alpha <-parameters[4]
  beta1 <-parameters[5]
  z       <-(x-mu)/sigma
if(sigma>0 & s>0 & alpha>0 & beta1>0)
  {
    GX <- GNPhi_func(z,s)
    T1 <- 1/(beta(alpha,beta1))
    cd<- T1*Ibeta(GX,alpha,beta1)
  }
```

```
  else
  {
    cd<-rep(0,length(x))
  }

return(cd)
}
#####################################################################
#Kumaraswamy Generalised normal density function

dKGN_func <- function(x,parameters)
{
  mu    <-parameters[1]
  sigma <-parameters[2]
  s     <-parameters[3]
  alpha <-parameters[4]
  beta1 <-parameters[5]
  z     <-(x-mu)/sigma
if(sigma>0 & s>0 & alpha>0 & beta1>0)
  {
    T1 <- alpha*beta1/sigma*GNphi_func(z,s)
    T2 <- GNPhi_func(z,s)^(alpha-1)
    T3 <- (1-GNPhi_func(z,s)^alpha)^(beta1-1)
    d  <- T1*T2*T3
  }
  else
  {
    d<-rep(0,length(x))
  }

  return(d)
}

#####################################################################
```

```
#Kumaraswamy-Generalised normal cumulative density function
cdfKGN_func <- function(x,parameters)
{
  mu    <-parameters[1]
  sigma <-parameters[2]
  s     <-parameters[3]
  alpha <-parameters[4]
  beta1 <-parameters[5]
  z     <-(x-mu)/sigma
if(sigma>0 & s>0 & alpha>0 & beta1>0)
  {
    cd  <- 1-(1-GNPhi_func(z,s)^alpha)^beta1
  }
  else
  {
    cd<-rep(0,length(x))
  }

return(cd)
}
##############################################################################
#Generalised normal density function

dGN_func <- function(x,parameters)
{
  mu    <-parameters[1]
  sigma <-parameters[2]
  s     <-parameters[3]
  z     <-(x-mu)/sigma
  if(sigma>0 & s>0)
  {
    d  <- GNphi_func(z,s)/sigma
  }
  else
```

```
  {
    d<-rep(0,length(x))
  }

  return(d)
}
#############################################################################
#Generalised normal cumulative density function

cdfGN_func <- function(x,parameters)
{
  mu    <-parameters[1]
  sigma <-parameters[2]
  s     <-parameters[3]
  z     <-(x-mu)/sigma
  if(sigma>0 & s>0)
  {
    cd <- GNPhi_func(z,s)
  }
  else
  {
    cd<-rep(0,length(x))
  }

  return(cd)
}
#############################################################################
#Kumaraswamy normal density function

dKN_func <- function(x,parameters)
{
    parameters<-c(parameters[1],parameters[2]*sqrt(2),2,parameters[3:4])
    d  <- dKGN_func(x,parameters)
  return(d)
```

```
}
#########################################################################
#Kumaraswamy normal cumulative density function

cdfKN_func <- function(x,parameters)
{
    parameters<-c(parameters[1],parameters[2]*sqrt(2),2,parameters[3:4])
    d  <- cdfKGN_func(x,parameters)
  return(d)
}
#########################################################################
#beta normal density function

dBN_func <- function(x,parameters)
{
  mu    <-parameters[1]
  sigma <-parameters[2]*sqrt(2)
  alpha <-parameters[3]
  beta1 <-parameters[4]
  if(sigma>0 & alpha>0 & beta1>0)
  {
    T1 <- 1/(sigma*beta(alpha,beta1))
    T2 <- pnorm(x, mean = mu, sd = sigma, lower.tail = TRUE, log.p = FALSE)^(alpha-1)
    T3 <- (1-pnorm(x, mean = mu, sd = sigma, lower.tail = TRUE, log.p = FALSE))^(beta1-1
    T4 <-dnorm(x, mean = mu, sd = sigma, log = FALSE)
    d  <- T1*T2*T3*T4
  }
  else
  {
    d<-rep(0,length(x))
  }

  return(d)
}
```

```
#########################################################################
#beta normal cumulative density function

cdfBN_func <- function(x,parameters)
{
  mu    <-parameters[1]
  sigma <-parameters[2]*sqrt(2)
  alpha <-parameters[3]
  beta1 <-parameters[4]
  if(sigma>0 & alpha>0 & beta1>0)
  {
    GX <- pnorm(x, mean = mu, sd = sigma, lower.tail = TRUE, log.p = FALSE)
    T1 <- 1/(beta(alpha,beta1))
    cd<- T1*Ibeta(GX,alpha,beta1)
  }
  else
  {
    cd<-rep(0,length(x))
  }

  return(cd)
}
#########################################################################
# Generalised Phi functions
GNphi_func <- function(z,s)
{
  if(s>0)
  {
    T1<-s/(2*gamma(1/s))
    T2<-exp(-abs(z)^s)
      d<-T1*T2
  }
  else
  {
```

```
    d<-rep(0,length(z))
  }

  return(d)
}

GNPhi_func <- function(z,s)
{

if(s>0)
  {
    pos<-z<=0
    neg<-z>0
    z[pos]<-Igamma(1/s,(-z[pos])^s,lower = FALSE)/(2*gamma(1/s)) # for z elements negati
    z[neg]<- 1-Igamma(1/s,z[neg]^s,lower = FALSE)/(2*gamma(1/s)) # for z elements positi
    phi<-z
  }
else
  {
    phi<-rep(0,length(z))
  }

return(phi)
}


#############################################################################
# minLikelihood function for given density_func and parameters
minLL_func <- function(density_func,x,parameters)
{
  maxLL <- -Inf
  LLvec <- density_func(x,parameters)
  if(all(is.finite(LLvec)))
  {
```

```
    if (all(LLvec!=0))
    {
      LLvec <- log(LLvec)
      maxLL <- sum(LLvec)
    }
  }
  minLL <- -maxLL
  return(minLL)
}

#############################################################################
# ML Selection function for given density_func and parameters

MLselect_func <- function(density_func,x,inputvalues)
{
minevaluation      <- Inf
    for (k in 1:ncol(inputvalues))
    {
      dummy          <- inputvalues[,k]
      evaluation     <- minLL_func(density_func,x,dummy)

      if (is.finite(evaluation) & evaluation<minevaluation)
      {
        minevaluation     <- evaluation
        minevaluationparameters <- dummy
      }
    }

if(minevaluation == Inf)
{
  minevaluationparameters<-NULL
}
return(minevaluationparameters)
}
```

```
#############################################################################
# minKS function for given parameters cdf_func and parameters

minKS_func<-function(cdf_func,x,cdfvalues,parameters)
{
  n<-length(x)
  statistic1 <- max(abs(cdf_func(x,parameters)-cdfvalues))
  statistic2 <- max(abs(cdf_func(x,parameters)+1/n-cdfvalues))
  statistic <- max(statistic1,statistic2)
  return(statistic)
}
#############################################################################
# KS Selection function for given parameters

KSselect_func <- function(cdf_func,x,cdfvalues,inputvalues)
{
  minevaluation      <- Inf
  for (k in 1:ncol(inputvalues))
  {
    dummy            <- inputvalues[,k]
    evaluation       <- minKS_func(cdf_func,x,cdfvalues,dummy)

    if (is.finite(evaluation) & evaluation<minevaluation)
    {
      minevaluation            <- evaluation
      minevaluationparameters <- dummy
    }
  }

  if(minevaluation == Inf)
  {
    minevaluationparameters<-NULL
  }
  return(minevaluationparameters)
```

```
}

#SIMULATION
#######################################################################
#Simulation of Beta Weibull
rBW_func<-function(parameters,n,seedval)
{
  set.seed(seedval)
  k      <-parameters[1]
  lambda<-parameters[2]
  alpha <-parameters[3]
  beta1 <-parameters[4]

  betaX <-rbeta(n,alpha,beta1)
  BWx   <-lambda*(-log(1-betaX))^(1/k)
  return(BWx)
}
#######################################################################
#Simulation of Kumaraswamy Weibull
rKW_func<-function(parameters,n,seedval)
{
  set.seed(seedval)
  k      <-parameters[1]
  lambda<-parameters[2]
  alpha <-parameters[3]
  beta1 <-parameters[4]

  KumX <- rkumar(n,alpha,beta1)
  KWx  <-    lambda*(-log(1-KumX))^(1/k)
  return(KWx)
}

#######################################################################
#Simulation of Beta Generalised Normal
```

```
rBGN_func<-function(parameters,n,seedval)
{
  set.seed(seedval)
  mu    <-parameters[1]
  sigma <-parameters[2]
  s     <-parameters[3]
  alpha <-parameters[4]
  beta1 <-parameters[5]

  y<-rbeta(n,alpha,beta1)
  neg<-y<=0.5
  pos<-y>0.5
  z<-rep(0,length(y))
  z[neg]<--Igamma.inv(1/s,((y[neg])*2*gamma(1/s)),lower = FALSE)^(1/s)
  z[pos]<-Igamma.inv(1/s,((1-y[pos])*2*gamma(1/s)),lower = FALSE)^(1/s)
  x<-sigma*z+mu

  return(x)
}

#############################################################################
#Simulation of Kumaraswamy Generalised Normal
rKGN_func<-function(parameters,n,seedval)
{
  set.seed(seedval)
  mu    <-parameters[1]
  sigma <-parameters[2]
  s     <-parameters[3]
  alpha <-parameters[4]
  beta1 <-parameters[5]

  y<-rkumar(n,alpha,beta1)
  neg<-y<=0.5
  pos<-y>0.5
```

```
  z<-rep(0,length(y))
  z[neg]<--Igamma.inv(1/s,((y[neg])*2*gamma(1/s)),lower = FALSE)^(1/s)
  z[pos]<-Igamma.inv(1/s,((1-y[pos])*2*gamma(1/s)),lower = FALSE)^(1/s)
  x<-sigma*z+mu

  return(x)
}
#########################################################################
#Simulation of Generalised Normal
rGN_func<-function(parameters,n,seedval)
{
  set.seed(seedval)
  mu    <-parameters[1]
  sigma <-parameters[2]
  s     <-parameters[3]

  y<-runif(n)
  neg<-y<=0.5
  pos<-y>0.5
  z<-rep(0,length(y))
  z[neg]<--Igamma.inv(1/s,((y[neg])*2*gamma(1/s)),lower = FALSE)^(1/s)
  z[pos]<-Igamma.inv(1/s,((1-y[pos])*2*gamma(1/s)),lower = FALSE)^(1/s)
  x<-sigma*z+mu

}
#########################################################################
#Simulation of Kumaraswamy normal
rKN_func<-function(parameters,n,seedval)
{
  set.seed(seedval)
  mu    <-parameters[1]
  sigma <-parameters[2]
  s     <-2
  alpha <-parameters[3]
```

```
  beta1 <-parameters[4]

  y<-rkumar(n,alpha,beta1)
  neg<-y<=0.5
  pos<-y>0.5
  z<-rep(0,length(y))
  z[neg]<--Igamma.inv(1/s,((y[neg])*2*gamma(1/s)),lower = FALSE)^(1/s)
  z[pos]<-Igamma.inv(1/s,((1-y[pos])*2*gamma(1/s)),lower = FALSE)^(1/s)
  x<-sigma*z+mu

  return(x)
}
#########################################################################
#Simulation of Kumaraswamy normal
rBN_func<-function(parameters,n,seedval)
{
  set.seed(seedval)
  mu    <-parameters[1]
  sigma <-parameters[2]
  s     <-2
  alpha <-parameters[3]
  beta1 <-parameters[4]

  y<-rbeta(n,alpha,beta1)
  neg<-y<=0.5
  pos<-y>0.5
  z<-rep(0,length(y))
  z[neg]<--Igamma.inv(1/s,((y[neg])*2*gamma(1/s)),lower = FALSE)^(1/s)
  z[pos]<-Igamma.inv(1/s,((1-y[pos])*2*gamma(1/s)),lower = FALSE)^(1/s)
  x<-sigma*z+mu

  return(x)
}
```

```
#ESTIMATION
#############################################################################
#Estimation of a Beta Weibull
minLL_BW<-function(x,parameters)
{
  return(minLL_func(dBW_func,x,parameters))
}

minKS_BW<-function(x,cdfvalues,parameters)
{
  return(minKS_func(cdfBW_func,x,cdfvalues,parameters))
}
#call function
eBW<-function(x,parameter_range,size,iterations,type)

{
  #Empirical cdf values
  cdensity_func<-ecdf(x)
  cdf_x<-cdensity_func(x)

  estimates<-rep(0.1,4)
  for(z in 1:iterations)
  {
    # Random allocation of starting values for optimisation
    k      <- runif(size,0.000001,parameter_range)
    lambda<- runif(size,0.000001,parameter_range)
    alpha <-runif(size,0.000001,parameter_range)
    beta1 <-runif(size,0.000001,parameter_range)
    n_startvalues<-rbind(k,lambda,alpha,beta1)

    if(type=="MLE")
    {
      # ML estimisation BW
      startpar <-MLselect_func(dBW_func,x,n_startvalues)
```

```
      optimised<-optim(startpar,minLL_BW,x = x)
      dummy     <-optimised$par
    }
    if(type=="KS")
    {
      #KS estimation BW
      startpar <-KSselect_func(cdfBW_func,x,cdf_x,n_startvalues)
      optimised<-optim(startpar,minKS_BW,x = x, cdfvalues = cdf_x)
      dummy     <-optimised$par
    }
    if(minKS_BW(x,cdf_x,dummy)<minKS_BW(x,cdf_x,estimates))
    {
      estimates<-dummy
    }
  }
  return(estimates)
}
#############################################################################
#Estimation of a Kumaraswamy Weibull
minLL_KW<-function(x,parameters)
{
  return(minLL_func(dKW_func,x,parameters))
}

minKS_KW<-function(x,cdfvalues,parameters)
{
  return(minKS_func(cdfKW_func,x,cdfvalues,parameters))
}
#call function
eKW<-function(x,parameter_range,size,iterations,type)

{
  #Empirical cdf values
  cdensity_func<-ecdf(x)
```

```
  cdf_x<-cdensity_func(x)


  estimates<-rep(0.1,4)
  for(z in 1:iterations)
  {
    # Random allocation of starting values for optimisation
    k      <- runif(size,0.000001,parameter_range)
    lambda<- runif(size,0.000001,parameter_range)
    alpha <-runif(size,0.000001,parameter_range)
    beta1 <-runif(size,0.000001,parameter_range)
    n_startvalues<-rbind(k,lambda,alpha,beta1)

    if(type=="MLE")
    {
      # ML estimisation KW
      startpar <-MLselect_func(dKW_func,x,n_startvalues)
      optimised<-optim(startpar,minLL_KW,x = x)
      dummy     <-optimised$par
    }
    if(type=="KS")
    {
      #KS estimation KW
      startpar <-KSselect_func(cdfKW_func,x,cdf_x,n_startvalues)
      optimised<-optim(startpar,minKS_KW,x = x, cdfvalues = cdf_x)
      dummy     <-optimised$par
    }
    if(minKS_KW(x,cdf_x,dummy)<minKS_KW(x,cdf_x,estimates))
    {
      estimates<-dummy
    }
  }
  return(estimates)
}
```

```
#Estimation of a Beta Generalised Normal
minLL_BGN<-function(x,parameters)
{
  return(minLL_func(dBGN_func,x,parameters))
}

minKS_BGN<-function(x,cdfvalues,parameters)
{
  return(minKS_func(cdfBGN_func,x,cdfvalues,parameters))
}
#call function
eBGN<-function(x,parameter_range,size,iterations,type)

{
  #Empirical cdf values
  cdensity_func<-ecdf(x)
  cdf_x<-cdensity_func(x)

  estimates<-rep(0.1,5)
  for(z in 1:iterations)
  {
    # Random allocation of starting values for optimisation
    mu    <-rep(mean(x),size)
    sigma <-runif(size,0.000001,parameter_range)
    s     <-runif(size,0.000001,parameter_range)
    alpha <-runif(size,0.000001,parameter_range)
    beta1 <-runif(size,0.000001,parameter_range)
    n_startvalues<-rbind(mu,sigma,s,alpha,beta1)

    if(type=="MLE")
    {
      # ML estimisation BGN
      startpar <-MLselect_func(dBGN_func,x,n_startvalues)
```

```
      optimised<-optim(startpar,minLL_BGN,x = x)
      dummy     <-optimised$par
    }
    if(type=="KS")
    {
      #KS estimation BGN
      startpar <-KSselect_func(cdfBGN_func,x,cdf_x,n_startvalues)
      optimised<-optim(startpar,minKS_BGN,x = x, cdfvalues = cdf_x)
      dummy     <-optimised$par
    }
    if(minKS_BGN(x,cdf_x,dummy)<minKS_BGN(x,cdf_x,estimates))
    {
      estimates<-dummy
    }
  }
  return(estimates)
}
#############################################################################
#Estimation of a Kumaraswamy Generalised Normal
minLL_KGN<-function(x,parameters)
{
  return(minLL_func(dKGN_func,x,parameters))
}

minKS_KGN<-function(x,cdfvalues,parameters)
{
  return(minKS_func(cdfKGN_func,x,cdfvalues,parameters))
}
#call function
eKGN<-function(x,parameter_range,size,iterations,type)

{
  #Empirical cdf values
  cdensity_func<-ecdf(x)
```

```
  cdf_x<-cdensity_func(x)

  estimates<-rep(0.1,5)
  for(z in 1:iterations)
  {
    # Random allocation of starting values for optimisation
    mu    <-rep(mean(x),size)
    sigma <-runif(size,0.000001,parameter_range)
    s     <-runif(size,0.000001,parameter_range)
    alpha <-runif(size,0.000001,parameter_range)
    beta1 <-runif(size,0.000001,parameter_range)
    n_startvalues<-rbind(mu,sigma,s,alpha,beta1)

    if(type=="MLE")
    {
      # ML estimisation KGN
      startpar <-MLselect_func(dKGN_func,x,n_startvalues)
      optimised<-optim(startpar,minLL_KGN,x = x)
      dummy    <-optimised$par
    }
    if(type=="KS")
    {
      #KS estimation KGN
      startpar <-KSselect_func(cdfKGN_func,x,cdf_x,n_startvalues)
      optimised<-optim(startpar,minKS_KGN,x = x, cdfvalues = cdf_x)
      dummy    <-optimised$par
    }
    if(minKS_KGN(x,cdf_x,dummy)<minKS_KGN(x,cdf_x,estimates))
    {
      estimates<-dummy
    }
  }
  return(estimates)
}
```

```
#############################################################################
#Estimation of a Generalised Normal
minLL_GN<-function(x,parameters)
{
  return(minLL_func(dGN_func,x,parameters))
}

minKS_GN<-function(x,cdfvalues,parameters)
{
  return(minKS_func(cdfGN_func,x,cdfvalues,parameters))
}
#call function
eGN<-function(x,parameter_range,size,iterations,type)

{
  #Empirical cdf values
  cdensity_func<-ecdf(x)
  cdf_x<-cdensity_func(x)

  estimates<-rep(0.1,5)
  for(z in 1:iterations)
  {
    # Random allocation of starting values for optimisation
    mu    <-rep(mean(x),size)
    sigma <-runif(size,0.000001,parameter_range)
    s     <-runif(size,0.000001,parameter_range)
    n_startvalues<-rbind(mu,sigma,s)

    if(type=="MLE")
    {
      # ML estimisation GN
      startpar <-MLselect_func(dGN_func,x,n_startvalues)
      optimised<-optim(startpar,minLL_GN,x = x)
      dummy    <-optimised$par
```

```
    }
    if(type=="KS")
    {
      #KS estimation GN
      startpar <-KSselect_func(cdfGN_func,x,cdf_x,n_startvalues)
      optimised<-optim(startpar,minKS_GN,x = x, cdfvalues = cdf_x)
      dummy    <-optimised$par
    }
    if(minKS_GN(x,cdf_x,dummy)<minKS_GN(x,cdf_x,estimates))
    {
      estimates<-dummy
    }
  }
  return(estimates)
}
##############################################################################
#Estimation of a Kumaraswamy normal
minLL_KN<-function(x,parameters)
{
  return(minLL_func(dKN_func,x,parameters))
}

minKS_KN<-function(x,cdfvalues,parameters)
{
  return(minKS_func(cdfKN_func,x,cdfvalues,parameters))
}
#call function
eKN<-function(x,parameter_range,size,iterations,type)

{
  #Empirical cdf values
  cdensity_func<-ecdf(x)
  cdf_x<-cdensity_func(x)
```

```
  estimates<-rep(0.1,4)
  for(z in 1:iterations)
  {
    # Random allocation of starting values for optimisation
    mu     <-rep(mean(x),size)
    sigma <-runif(size,0.000001,parameter_range)
    alpha <-runif(size,0.000001,parameter_range)
    beta1 <-runif(size,0.000001,parameter_range)
    n_startvalues<-rbind(mu,sigma,alpha,beta1)

    if(type=="MLE")
    {
      # ML estimisation KN
      startpar <-MLselect_func(dKN_func,x,n_startvalues)
      optimised<-optim(startpar,minLL_KN,x = x)
      dummy     <-optimised$par
    }
    if(type=="KS")
    {
      #KS estimation KN
      startpar <-KSselect_func(cdfKN_func,x,cdf_x,n_startvalues)
      optimised<-optim(startpar,minKS_KN,x = x, cdfvalues = cdf_x)
      dummy     <-optimised$par
    }
    if(minKS_KN(x,cdf_x,dummy)<minKS_KN(x,cdf_x,estimates))
    {
      estimates<-dummy
    }
  }
  return(estimates)
}
```

Listing 3: Diamond Price Data Estimation and Visualisation

```
source("UserFunc.R")
diamonds <- read.csv(file="Diamonds.csv"
                     , header=TRUE
                     , sep=",")

obs<-diamonds$USD
hist(obs)
t<-1
w<-10



cd_func<-ecdf(obs)

cdf_obs<-cd_func(obs)

estBW<-c(0.1702306,335.1246261,61.5313845,19.7527699)
estKW<-c(0.1507911,7.3050742,38.1096384,21.6602616)



# for(i in 1:5)
# {
#   dummy<-eBW(obs,i*w,1000,1,"MLE")
#   if(minKS_BW(obs,cdf_obs,dummy) < minKS_BW(obs,cdf_obs,estBW))
#   {
#     estBW<-dummy
#   }
#
#   dummy<-eKW(obs,i*w,1000,1,"MLE")
#   if(minKS_KW(obs,cdf_obs,dummy) < minKS_KW(obs,cdf_obs,estKW))
#   {
#     estKW<-dummy
#   }
# }
```

```
#########################################################################
# Comparison of results
# Plot:
# The black line is a kernel density estimate for the sample.
# The green line represents the density based on the BW.
# The green line represents the density based on the KW.
par(mfrow=c(1,2))
x <- seq(0,range(obs)[2], by = 0.05)

a <- dBW_func(x,estBW)
b <- dKW_func(x,estKW)
ylim<-c(0,1.1*max(dBW_func(obs,estBW)))

plot(density(obs,adjust = 0.7)
      ,type = "l"
      ,xlab = "x"
      ,ylab = "Density"
      , col ="black"
      , lty = "dotted"
      ,xlim = c(0,range(x)[2])
      ,ylim = ylim
      ,main = ""      )
lines(x,a,col = "red")
lines(x,b,col = "dark green")
legend(legend = c("Kernel"
                  ,"BW"
                  ,"KW")
       ,col = c("black"
                ,"red"
                ,"dark green")
```

```
        ,x ="topright"
        ,lty = c(3,1,1,1)
        ,cex = 0.75
        ,bty = "n")

Sys.sleep(t)


lx<-quantile(obs,probs = c(0.00025,0.99975))


curve(cd_func(x)
        ,lx[1]
        ,lx[2]
        ,ylim = c(0,1)
        ,ylab = "Cumulative Density")
curve(cdfBW_func(x,estBW)
        ,lx[1]
        ,lx[2]
        ,col = "red"
        ,add = TRUE)
curve(cdfKW_func(x,estKW)
        ,lx[1]
        ,lx[2]
        ,col = "dark green"
        ,add = TRUE)
legend(legend = c("Kernel"
                  ,"BW"
                  ,"KW")
        ,col = c("black","red","dark green")
        ,x ="bottomright"
        ,lty = c(3,1,1,1)
        ,cex = 0.75
        ,bty = "n")
```

```
KS_BW<-minKS_func(cdfBW_func,obs,cdf_obs,estBW)
KS_KW<-minKS_func(cdfKW_func,obs,cdf_obs,estKW)

KS_BW
KS_KW

par<-length(parameters)

maxLL<--minLL_BW(obs,estBW)
n<-length(obs)
AIC_BW<-2*(par-maxLL)
BIC_BW<-par*log(n)-2*maxLL
CAIC_BW<-par*(log(n)+1)-2*maxLL

maxLL<--minLL_KW(obs,estKW)
n<-length(obs)
AIC_KW<-2*(par-maxLL)
BIC_KW<-par*log(n)-2*maxLL
CAIC_KW<-par*(log(n)+1)-2*maxLL

c(KS_BW,KS_KW)
c(AIC_BW,AIC_KW)
c(BIC_BW,BIC_KW)
c(CAIC_BW,CAIC_KW)

estBW
estKW

library(stargazer)
tablBW<-estBW
tablKW<-estKW
tabl<-rbind(tablBW,tablKW)
```

```
tabl<-as.matrix(tabl
                ,nrow = 2
                ,ncol = 9
                ,byrow = TRUE)
rownames(tabl)<-c("BW","KW")
colnames(tabl)<-c("k","lambda","alpha","beta")
stargazer(tabl
          ,title = "Maximum Likelihood Estimates")

tablBW<-c(AIC_BW,BIC_BW,CAIC_BW)
tablKW<-c(AIC_KW,BIC_KW,CAIC_KW)
tabl<-rbind(tablBW,tablKW)
tabl<-as.matrix(tabl
                ,nrow = 2
                ,ncol = 9
                ,byrow = TRUE)
rownames(tabl)<-c("BW","KW")
colnames(tabl)<-c("AIC","BIC","CAIC","KS")
stargazer(tabl
          ,title = "Goodness-of-fit Statistics")
```

Listing 4: Relapse-time Data Estimation and Visualisation

```
source("UserFunc.R")
library(survival)
library(moments)
obs<-nwtco$edrel[nwtco$stage==4]
hist(obs)
skewness(obs)
kurtosis(obs)

t<-1
w<-0.0025

estBGN<-c(104.93845,331.57412,0.24972,285.58180,248.87601)
estKGN<-c(118.1720226,113.2945576,0.2732401,11.1949726,190.8560605)
estGN<-c(3102.270,3099.292,7355.291)

# for(i in 1:5)
# {
#   dummy<-eBGN(obs,i*10*w,1000,1,"MLE")
#   if(minLL_BGN(obs,dummy) < minLL_BGN(obs,estBGN))
#   {
#     estBGN<-dummy
#   }
#
#   dummy<-eKGN(obs,i*10*w,1000,1,"MLE")
#   if(minLL_KGN(obs,dummy) < minLL_KGN(obs,estKGN))
#   {
#     estKGN<-dummy
#   }
#   dummy<-eGN(obs,i*20,1000,1,"MLE")
#   if(minLL_GN(obs,dummy) < minLL_GN(obs,estGN))
#   {
#     estGN<-dummy
#   }
```

```
#
# }

#####################################################################
# Comparison of results
cd_func<-ecdf(obs)
cdf_obs<-cd_func(obs)

KS_BGN<-minKS_func(cdfBGN_func,obs,cdf_obs,estBGN)
KS_KGN<-minKS_func(cdfKGN_func,obs,cdf_obs,estKGN)
KS_GN<-minKS_func(cdfGN_func,obs,cdf_obs,estGN)

par<-length(parameters)

maxLL<-minLL_BGN(obs,estBGN)
n<-length(obs)
AIC_BGN<-2*(par-maxLL)
BIC_BGN<-par*log(n)-2*maxLL
CAIC_BGN<-par*(log(n)+1)-2*maxLL

maxLL<-minLL_KGN(obs,estKGN)
n<-length(obs)
AIC_KGN<-2*(par-maxLL)
BIC_KGN<-par*log(n)-2*maxLL
CAIC_KGN<-par*(log(n)+1)-2*maxLL

maxLL<-minLL_GN(obs,estGN)
n<-length(obs)
AIC_GN<-2*(par-maxLL)
BIC_GN<-par*log(n)-2*maxLL
CAIC_GN<-par*(log(n)+1)-2*maxLL

# AIC=2*(para-logval);
# BIC=para*log(n)-2*logval;
```

```
# CAIC=para*(log(n)+1)-2*logval;

c(KS_BGN,KS_KGN,KS_GN)
c(AIC_BGN,AIC_KGN,AIC_GN)
c(BIC_BGN,BIC_KGN,BIC_GN)
c(CAIC_BGN,CAIC_KGN,CAIC_GN)

c(KS_BGN,KS_KGN)
c(AIC_BGN,AIC_KGN)
c(BIC_BGN,BIC_KGN)
c(CAIC_BGN,CAIC_KGN)

estBGN
estKGN
estGN

#####################################Plot reproduction in article#################################
#SASA PLOT
#Axis limits
par(mfrow=c(1,2))
dfunc<-approxfun(density(obs))
yl<-1.6*range(dfunc(obs))

xl<-range(obs)
# x<-seq(0,xl[2],1)
x<-seq(xl[1],xl[2])

hist(obs
     ,breaks = 12
     ,probability = TRUE
     ,xlab = "x"
     ,ylab = "PDF"
     ,main = ""
     ,ylim= yl
```

```
      ,xlim = xl)
lines(density(obs),lty = "dotted",col = "blue")
{par(new = TRUE)}
plot(x,dKGN_func(x,estKGN)
      ,col = "dark green"
      ,type = "l"
      ,xaxt="n"
      ,yaxt ="n"
      ,ann = FALSE
      ,ylim= yl
      ,xlim = xl)
{par(new = TRUE)}
plot(x,dBGN_func(x,estBGN)
      ,col = "red"
      ,type = "l"
      ,xaxt="n"
      ,yaxt ="n"
      ,ann = FALSE
      ,ylim= yl
      ,xlim = xl)
title()
legend(legend = c("BGN"
                  ,"KGN"
                  ,"Kernel")
       ,col = c("red"
                ,"dark green"
                ,"blue")
       ,x ="topright"
       ,lty = c(1,1,3)
       ,cex = 0.75
       ,bty = "n" )

#########################CDF plots#########################################
```

```
lx<-quantile(obs,probs = c(0.00005,0.999975))


curve(cd_func(x)
      ,lx[1]
      ,lx[2]
      ,ylim = c(0,1)
      ,col ="blue"
      ,lty = 3
      ,ylab = "CDF")
curve(cdfBGN_func(x,estBGN)
      ,lx[1]
      ,lx[2]
      ,col = "red"
      ,lty = 1
      ,add = TRUE)
curve(cdfKGN_func(x,estKGN)
      ,lx[1]
      ,lx[2]
      ,col = "dark green"
      ,add = TRUE)
# title("Empirical & Estimated Cumulative Density")
legend(legend = c("BGN"
                  ,"KGN"
                  ,"Kernel")
       ,col = c("red"
                ,"dark green"
                ,"blue")
       ,x ="bottomright"
       ,lty = c(1,1,3)
       ,cex = 0.75
       ,bty = "n" )
#############################################################################################
library(stargazer)
tablKGN<-estKGN
```

```
tablBGN<-estBGN
tabl<-rbind(tablKGN,tablBGN)
tabl<-as.matrix(tabl
                ,nrow = 2
                ,ncol = 9
                ,byrow = TRUE)
rownames(tabl)<-c("KGN","BGN")
colnames(tabl)<-c("mu","sigma","s","alpha","beta")
stargazer(tabl
          ,title = "Maximum Likelihood Estimates")

tablKGN<-c(AIC_KGN,BIC_KGN,CAIC_KGN)
tablBGN<-c(AIC_BGN,BIC_BGN,CAIC_BGN)
tabl<-rbind(tablKGN,tablBGN)
tabl<-as.matrix(tabl
                ,nrow = 2
                ,ncol = 9
                ,byrow = TRUE)
rownames(tabl)<-c("KGN","BGN")
colnames(tabl)<-c("AIC","BIC","CAIC")
stargazer(tabl
          ,title = "Goodness-of-fit Statistics")
```

Listing 5: Mathematica Functions

$$\text{dbw}[\text{x\_}, \kappa\_, \lambda\_, \alpha\_, \beta\_]:=\frac{\kappa}{\text{Beta}[\alpha,\beta]} \times \frac{x^{\kappa-1}}{\lambda^{\kappa}} \times \left(1-\text{Exp}\left[-\left(\frac{x}{\lambda}\right)^{\kappa}\right]\right)^{\alpha-1} \times \text{Exp}\left[-\beta\left(\frac{x}{\lambda}\right)^{\kappa}\right]$$

$$\text{cdbw}[\text{x\_}, \kappa\_, \lambda\_, \alpha\_, \beta\_]:=\text{BetaRegularized}\left[1-\text{Exp}\left[-\left(\frac{x}{\lambda}\right)^{\kappa}\right], \alpha, \beta\right]$$

$$\text{hzbw}[\kappa\_, \lambda\_, \alpha\_, \beta\_]:=\frac{\text{dbw}[x,\kappa,\lambda,\alpha,\beta]}{1-\text{cdbw}[x,\kappa,\lambda,\alpha,\beta]}$$

$$\text{dkw}[\text{x\_}, \kappa\_, \lambda\_, \alpha\_, \beta\_]:=\alpha \times \beta \times \kappa \times \frac{x^{\kappa-1}}{\lambda^{\kappa}} \times \text{Exp}\left[-\left(\frac{x}{\lambda}\right)^{\kappa}\right] \times \left(1-\text{Exp}\left[-\left(\frac{x}{\lambda}\right)^{\kappa}\right]\right)^{\alpha-1} \times \left(1-\left(1-\text{Exp}\left[-\left(\frac{x}{\lambda}\right)^{\kappa}\right]\right)^{\alpha}\right)^{\beta-1}$$

$$\text{cdkw}[\text{x\_}, \kappa\_, \lambda\_, \alpha\_, \beta\_]:=1-\left(1-\left(1-\text{Exp}\left[-\left(\frac{x}{\lambda}\right)^{\kappa}\right]\right)^{\alpha}\right)^{\beta}$$

$$\text{hzkw}[\kappa\_, \lambda\_, \alpha\_, \beta\_]:=\frac{\text{dkw}[x,\kappa,\lambda,\alpha,\beta]}{1-\text{cdkw}[x,\kappa,\lambda,\alpha,\beta]}$$

$$\text{phi}[\text{s\_}, \text{z\_}]:=\frac{s}{2\times\text{Gamma}\left[\frac{1}{s}\right]} \times \text{Exp}\left[-(\text{Abs}[z])^{s}\right]$$

$$\text{Phi}[\text{s\_}, \text{z\_}]:=\text{If}\left[z \leq 0, \frac{\text{Gamma}\left[\frac{1}{s}, (-z)^{s}\right]}{2\times\text{Gamma}\left[\frac{1}{s}\right]}, 1-\frac{\text{Gamma}\left[\frac{1}{s}, z^{s}\right]}{2\times\text{Gamma}\left[\frac{1}{s}\right]}\right]$$

$$\text{dbgn}[\text{x\_}, \mu\_, \sigma\_, \text{s\_}, \alpha\_, \beta\_]:=$$
$$\frac{1}{\sigma\times\text{Beta}[\alpha,\beta]} \times \left(\text{Phi}\left[s, \frac{x-\mu}{\sigma}\right]\right)^{\alpha-1} \times \left(1-\left(\text{Phi}\left[s, \frac{x-\mu}{\sigma}\right]\right)\right)^{\beta-1} \times \text{phi}\left[s, \frac{x-\mu}{\sigma}\right]$$

$$\text{cdbgn}[\text{x\_}, \mu\_, \sigma\_, \text{s\_}, \alpha\_, \beta\_]:=\text{BetaRegularized}\left[\text{Phi}\left[s, \frac{x-\mu}{\sigma}\right], \alpha, \beta\right]$$

$$\text{hzbgn}[\mu\_, \sigma\_, \text{s\_}, \alpha\_, \beta\_]:=\frac{\text{dbgn}[x,\mu,\sigma,s,\alpha,\beta]}{1-\text{cdbgn}[x,\mu,\sigma,s,\alpha,\beta]}$$

$$\text{dkgn}[\mu\_, \sigma\_, \text{s\_}, \alpha\_, \beta\_]:=\frac{\alpha\times\beta}{\sigma} \times \left(\text{Phi}\left[s, \frac{x-\mu}{\sigma}\right]\right)^{\alpha-1} \times \left(1-\left(\text{Phi}\left[s, \frac{x-\mu}{\sigma}\right]\right)^{\alpha}\right)^{\beta-1} \times \text{phi}\left[s, \frac{x-\mu}{\sigma}\right]$$

$$\text{cdkgn}[\text{x\_}, \mu\_, \sigma\_, \text{s\_}, \alpha\_, \beta\_]:=1-\left(1-\left(\text{Phi}\left[s, \frac{x-\mu}{\sigma}\right]\right)^{\alpha}\right)^{\beta}$$

$$\text{hzkgn}[\mu\_, \sigma\_, \text{s\_}, \alpha\_, \beta\_]:=\frac{\text{dkgn}[x,\mu,\sigma,s,\alpha,\beta]}{1-\text{cdkgn}[x,\mu,\sigma,s,\alpha,\beta]}$$