Honours Research Reports

WST795/STK795

Department of Statistics, University of Pretoria

2016

Booklet compiled by IN Fabris-Rotelli

The analysis of multilevel models for hierarchical data

Michelle Alexander 13106075

STK795 Research Report

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

Supervisor: Dr G Crafford

Department of Statistics, University of Pretoria

30 September 2016

Abstract

Multilevel models are used to model data that are nested within an organisational hierarchy. Owing to the availability of large amounts of data that specifically inhibits a hierarchical structure, traditional methods of modeling cannot accommodate the dependence within hierarchical levels as multilevel modeling. The regression coefficients are treated as random coefficients to accommodate the hierarchial structure, and explanatory variables at various levels are incorporated in the model.

Declaration

I, *Michelle Lynne Alexander,* 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.

_____
Michelle Lynne Alexander

_____
Gretel Crafford

_____
Date

Acknowledgements

I would like to thank the Centre for Artificial Intelligence Research (CAIR) and Statomed for financial support in the form of a post graduate bursary.

Contents

1 Introduction **6**

2 Literature Review **6**

3 Background theory and Application **7**
3.1 The PIRLS study . . . 7
3.2 Unconditional means model . . . 8
3.3 Model including the effects of level one predictors . . . 10
3.4 Model including the effects of level two predictors . . . 15
3.4.1 Model including the effects of emphasis on reading in early grades . . . 15
3.4.2 Model including the effects of a library . . . 17
3.4.3 Model including the effects of a library and the emphasis on reading in early grades . 19
3.5 Modeling including the effects of both level one and level two predictors . . . 20
3.5.1 Full model including all possible predictors . . . 20
3.5.2 Model without any interaction effects . . . 21
3.5.3 Final model . . . 22

4 Conclusion **25**

Appendix **28**

List of Figures

1 Comparison of PIRLS reading score between bench marking countries.[3] . . . 7
2 The impact on reading score if a student that owns books . . . 8
3 The impact on reading score if a student can access internet at home . . . 8
4 The impact on reading score if a student's parent tells them stories . . . 8
5 The impact on reading score for gender . . . 8
6 Output for Unconditional Means Model. . . . 10
7 Regression line for school 33. . . . 11
8 Regression line for school 50. . . . 12
9 Output for level one predictors . . . 14
10 Regression line for model with HRL predictor . . . 14
11 Output for level two predictor EREG . . . 16
12 Regression line for model with EREG predictor . . . 16
13 Output for level two predictor LIB . . . 18
14 Effect of LIB predictor on the intercept . . . 18
15 Output for level two predictors EREG and LIB . . . 19
16 Regression line for model with EREG and LIB predictors . . . 20
17 Output of full model. . . . 21
18 Output for the combined model without interaction effects . . . 22
19 Output for final model . . . 23
20 Regression line for HRL and LIB predictors and holding EREG constant . . . 24
21 Model with random intercepts only . . . 25

List of Tables

1 Comparison of Fit Statistics . . . 25

1 Introduction

Multilevel modeling techniques have become more attractive over the last two decades for analyzing data that has a hierarchical structure [5]. This type of data is frequently found within educational, clinical and research enviroments [1]. Multilevel models can be seen as regression models taking place at multiple levels. This report will exclusively examine a two level approach that can be easily adjusted to more complex models where more nested levels are considered. As mentioned previously, disregarding the levels within the hierarchy can influence estimated variances and covariance effects, leading to statistical significance test results being inaccurately interpreted. As stated by Bell [1], "Multilevel models can vary in terms of the number of levels (e.g. two level, three level), type of design (e.g. hierarchial, longitudinal with repeated measures), scale of the outcome variable (e.g. categorical, continuous), and number of outcomes (e.g. univariate, multivariate)". The two level model that will be built will consist of both fixed and random effects with special attention being given to categorical or continuous independent variables. In this report, the reading score of invidual students will be modelled and the impact of characteristics asssociated with the school those students attend will be studied. This will produce a basic two level structure with students at level one and schools, within which the students are nested, at level two. The multilevel model will allow questions concerning fixed effects to be explored, in addition to questions regarding random level one and level two coefficients and the variance-covariance components. In this report the theory behind multilevel modeling will be explained with the use of practical examples to illustrate the concept, for which the programming code will be presented. All programing will be coded in SAS® software[1]. The motivation behind this particular topic is a result of the recent increase in popularity of this field, and to gain insight as to what influences the reading score - not only within in the student's socio-economic context, but also the schools educational quality. As a result of this research, it is hoped that the concept of multilevel modeling will be mastered. Due to the heterogeneous population within the educational system, there is a great need to investigate hierarchial levels within South Africa.

2 Literature Review

Two of Singer's articles [5, 6], centred on multilevel modeling, have been reviewed to explore and understand this concept. Theoretical knowledge behind the idea of multilevel models, specifically hierarchial structures, is thoroughly explained with the aid of practical examples. In addition, there are step-by-step tutorials on how SAS PROC MIXED works and is used to construct the models. This provided a guide for effective use of SAS's procedure PROC MIXED and demonstrated the positive attributes of this procedure. Within Singer's article [5], the method of gradually building the model one step at a time was found to be extremely helpful. Furthermore, the output was interpreted and the code was evaluated in a very clear and understandable manner. Singer's articles have portrayed the concept of multilevel modeling brilliantly and have been of great assistance to grasp the motivation behind multilevel modeling. Goldstein's [2] book expressed alternative means of creating and motivating the two level model, presenting methods for estimating parameters, constructing confidence intervals, and testing functions of the parameters. It provided illustrative examples on observational data that has a clustered structure and how clusters occur. Specific attention was given to school effectiveness and the advantages associated with multilevel modeling in this regard. Goldstein focused on the construction of the model in the form of matrices and explicitly explained the importance of multiple residual terms. An article by Bell [1] provided a brief introduction to the field of multilevel modeling, along with real world examples to help with explanations. The practical examples were used in conjunction with narrative explanations on how PROC MIXED worked and the equivalent code was presented. Both random intercept and random intercept and slope models were demonstrated. Raudenbush's book [4] introduces the concept of multilevel modeling with specific attention being given to data with a hierarchical structure. Raudenbush begins by describing the vastness of data that inhibits a nested structure such, as organizational studies, demographic studies and educational research and provides the problems encounted with this type of data. A short history on the development of the statistical anaylsis of multilevel models is explained and

[1]The data analysis for this essay was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA.

the applications in which it has been most widely used. It was interesting to note the advancements that have been made in this field as well as the areas that languished and needed an alternative approach. Due to these recent developments, there is now an intergrated set of techniques that allows efficient estimation [4]. Raudenbush's book continues by studying the logic behind hierarchial linear models and emphasizes the significnace of the random coefficients allowing for the variability in the regression coefficients across the two levels. The concept of centering is touched on which has been of great assistance as not many articles studied so far have introduced this concept. In addition to this theory, in subsequent chapters Raudenbush illustrates all the examples and provides applications in organizational research.

3 Background theory and Application

3.1 The PIRLS study

As mentioned in the introduction, we will be examining a two level school effects model. Data that has an organizational hierarchy, such as students within a school can be used to predict a specific outcome using predictors from not only level-one but level-two as well. For the purposes of this report the data is sourced from PIRLS (Progress in International Reading Literacy Study) in 2011. The aim is to identify factors, on both student and school level that affect a South African student's reading score. Questionnaires were distributed to grade four students, their parents and to their schools. A number of variables were included in the questionnaire but for this report only a few of those variables will be used. These variables will be discussed as the report develops. South Africa, with an average point of 461 has performed considerably poorer than the International centre point of 500 which is illustrated in Figure 1. [3]

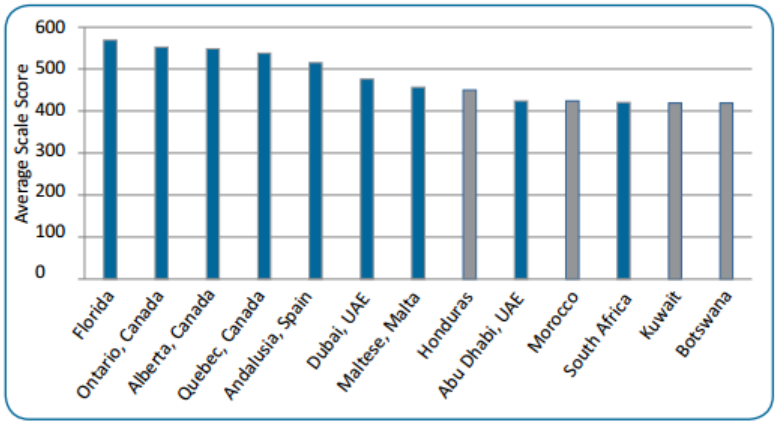


Figure 1: Comparison of PIRLS reading score between bench marking countries.[3]

Before the model is constructed a few interesting findings can be noted regarding the various influences on the reading score. The stronger the school rules are, the higher the reading score is. Schools that discipline more than three times a year have a mean reading score of 434.067, whereas schools that discipline between 2 to 3 times a year have a mean score of 391.92. It can be seen that there is a substantial difference in reading scores from students at different schools and there is a heavy impact of discipline within an eduation institution on the reading scores. In the figures below, four variables have been chosen to examine the strength of the influence it has on the reading score of a student. A quick summary of the results has been made. The reading score of a student that own books on average is 62.5 points higher than a student that does not own books. On average, a girl's reading score is considerably higherthan a boy's reading score. The more often a parent tells their child a story, the higher the reading score is. On average, the reading score of a student that does not have access to the internet is 66 points lower than a student that does have access to the internet.

Analysis Variable : Reading_score						
Student_owns_books	N Obs	N	Mean	Std Dev	Minimum	Maximum
NO	494	494	389.4791568	98.4871176	125.1641800	673.1830100
YES	1375	1375	452.0138667	109.6615884	112.3200700	799.9665900

Figure 2: The impact on reading score if a student that owns books

Analysis Variable : Reading_score						
Student_access_internet	N Obs	N	Mean	Std Dev	Minimum	Maximum
NO	1177	1177	410.3645597	101.8917942	112.3200700	736.8918400
YES	695	695	476.7636006	111.7993025	157.9869400	799.9665900

Figure 3: The impact on reading score if a student can access internet at home

Analysis Variable : Reading_score						
Parents_tell_stories	N Obs	N	Mean	Std Dev	Minimum	Maximum
NEVER OR ALMOST NEVER	125	125	385.0347600	106.3374032	113.4359700	764.0342400
OFTEN	763	763	461.1470587	104.7591957	132.8281400	799.9665900
SOMETIMES	875	875	428.2450188	108.0089195	157.4305600	758.2255300

Figure 4: The impact on reading score if a student's parent tells them stories

Analysis Variable : Reading_score						
Gender	N Obs	N	Mean	Std Dev	Minimum	Maximum
BOY	916	916	418.6976883	112.2249580	112.3200700	721.7233900
GIRL	969	969	449.0139231	107.3455767	153.8942800	799.9665900

Figure 5: The impact on reading score for gender

3.2 Unconditional means model

An unconditional means model is a model that contains neither level-one nor level-two predictors. It is the most basic model which can be used as a foundation for building more complex models [4]. This model will be fitted to analyze the variation in the reading scores across several schools. This can also be viewed as a one-way random effects ANOVA model. The group effects are seen as random which gives rise to the name, random-effects model [4]. This is presented in 1:

$$Y_{ij} = \mu + \alpha_j + r_{ij} \tag{1}$$

$$\text{where } \alpha_j \sim iidN(0, \tau_{00}) \text{ and } r_{ij} \sim iid\, N(0, \sigma^2).$$

It is important to note that this model comprises of two components; the fixed effect and the random effects. In this model there is only one fixed effect, namely μ and there are two random effects, α_j and r_{ij}. The variation component of the first effect (τ_{00}), denotes the variation between schools where as the variance

component of the second effect (σ^2) denotes the variation between students within a school. The equation (1) can be re-written in a broader approach that will be useful when building more complex models. The two models that are used are related, a model at level-one (student) and a model at level-two (school) which together will yield a combined model. The student's outcome at level-one will be made up of an intercept for the students school (β_{0i}) and a random error (r_{ij}) that is related to the i^{th} student within the j^{th} school.

$$Y = \beta_{0j} + r_{ij} \tag{2}$$

$$\text{where } r_{ij} \sim \text{N}(0,\sigma^2).$$

The school level at level-two comprises of the grand school mean (γ_{00}) and the random deviations from the mean (u_{0j}).

$$\beta_{0j} = \gamma_{00} + u_{0j} \tag{3}$$

$$\text{where } u_{0j} \sim \text{N}(0,\tau_{00}).$$

If equation 3 is substituted into equation 2 it will create the combined two level model:

$$Y_{ij} = \gamma_{00} + u_{0j} + r_{ij} \tag{4}$$

$$\text{where } u_{0j} \sim N(0, \tau_{00}) \text{ and } r_{ij} \sim N(0, \sigma^2).$$

Just like before this combined model 4 can be broken into two portions, the fixed effects and the random effects.

- γ_{00} is a fixed effect and expresses the average reading score in the population.
- τ_{00} is the variance component of the first random effect which represents the variability between school means.
- σ^2 is the variance component of the second random effect which represnts the variability within schools.

If the variance components from the two random effects are selected and placed into a matrix, assuming that u_{0j} and r_{ij} are independent, it will produce a block diagonal matrix [5]. The level-one outcome is predicted using a single parameter in level-two, i.e. the intercept. This model offers insight on the variability of the specific outcome across all levels, which makes it extremely useful. As well as creating a point estimate and confidence intervals for the grand mean which can be used as a starting point for analyzing the effectiveness of the model [4], the intraclass correlation coefficient can be used to calculate the proportion of variation in the outcome within level-two units. This can be calculated as follows:

$$\rho = \frac{\tau_{00}}{\tau_{00} + \sigma^2}$$

```
proc mixed data=schools.new6 noclprint covtest;
class idschool;
model Reading_score= /solution;
random intercept/sub=idschool;
run;
```

Covariance Parameter Estimates					
Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z
Intercept	IDSCHOOL	7254.51	1287.70	5.63	<.0001
Residual		5616.76	186.42	30.13	<.0001

Fit Statistics	
-2 Res Log Likelihood	21859.1
AIC (Smaller is Better)	21863.1
AICC (Smaller is Better)	21863.1
BIC (Smaller is Better)	21867.6

Solution for Fixed Effects					
Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	432.66	10.4205	68	41.52	<.0001

Figure 6: Output for Unconditional Means Model.

The two variance components from Figure 6 are significantly different from 0 with τ_{00} estimated at 7254.51 and σ^2 at 5616.76. From these results it can be seen that between school variance varies slightly to the within school variance. This can be an outcome of school predictors having a greater impact than student predictors. In South Africa, due to factors such as socio-economic status, it can be seen that a school can define the students within the school. For example, a school with a high socio-economic status is most likely to have students that have a high socio-economic status. Hence, the greater variance between schools compared to within schools. Another way to understand what creates variation in the reading score can be found through the intraclass correlation. The intraclass correlation is denoted by, ρ and it shows us the how much of the total variation is made up of variance between schools. Using the formula given previously, we can calculate out intraclass correlation as follows:

$$\rho = \frac{7254.51}{7254.51 + 5616.76}$$

$$\rho = 0.5636$$

From these results, it can be seen that more than half of the total variation stems from the variation between schools. These results coincide with our previous results when the estimated values were interpreted.

The section of output named, 'Fit Statistics' can be used to compare goodness of fit between various models that differ in their random effects but have the same fixed effects. In this example, the best fit would be the smallest value (as indicated on the SAS output). Without other models to compare, this information renders useless. The values will be compared later in this paper. The last estimate provided in the output is the estimate for the fixed effect. In this case, the estimate of 432.66 tells us the average reading score used in this sample of South African schools. Unfortunately, it is seen that the average reading score in a South African school is relatively below the global average score of 500.

3.3 Model including the effects of level one predictors

This section specifically looks at how the model changes when a level one (student) predictor is added to the model without involving any level two predictors. The student level predictor that will be used in this illustration will be denoted by 'HRL', which represents home resources for learning. This will be of assistance

to predict the outcome variable (reading score). The first model presented will be the level one model for only one school.

$$Y_i = \beta_0 + \beta_1 HRL_i + r_i \tag{5}$$

Take note that this model is only examining the relationship between the home resources for learning and reading score within one school. The variable within the equation 5 can be defined as follows:

- β_0 is the achievement mark of an individual student (i) when the HRL is equal to zero
- β_1 explains the relationship between the HRL of a student and that student's reading score, ie. if the HRL was to be increased by one unit, how this would affect the reading score
- r_i the unique error component that corresponds to student i

Firstly, the model expressing the relationship between HRL and the reading score will be examined within two schools. Two models can be built to represent school one and school two. These models are provided below:

$$Y_{i1} = \beta_{01} + \beta_{11} HRL_{i1} + r_{i1}$$

$$Y_{i2} = \beta_{02} + \beta_{12} HRL_{i2} + r_{i2}$$

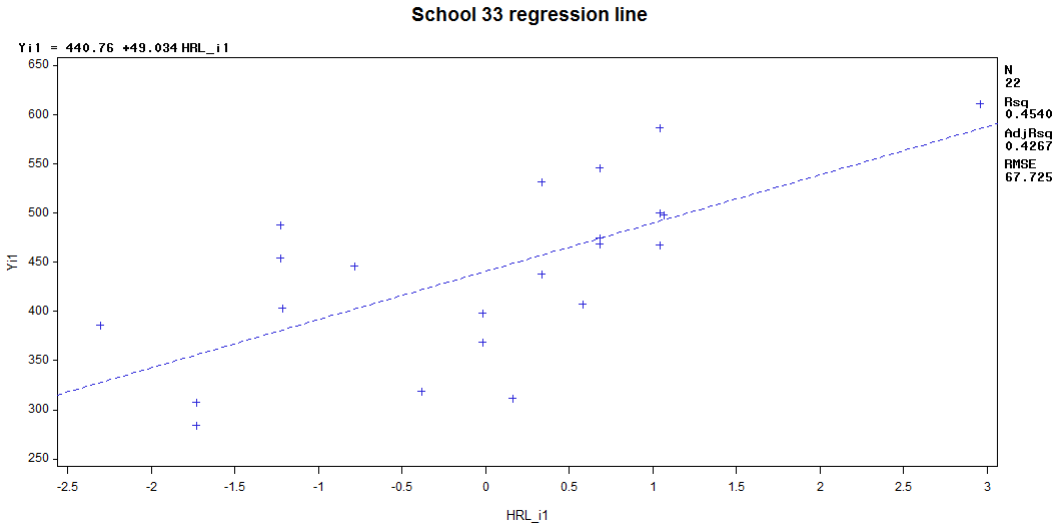


Figure 7: Regression line for school 33.

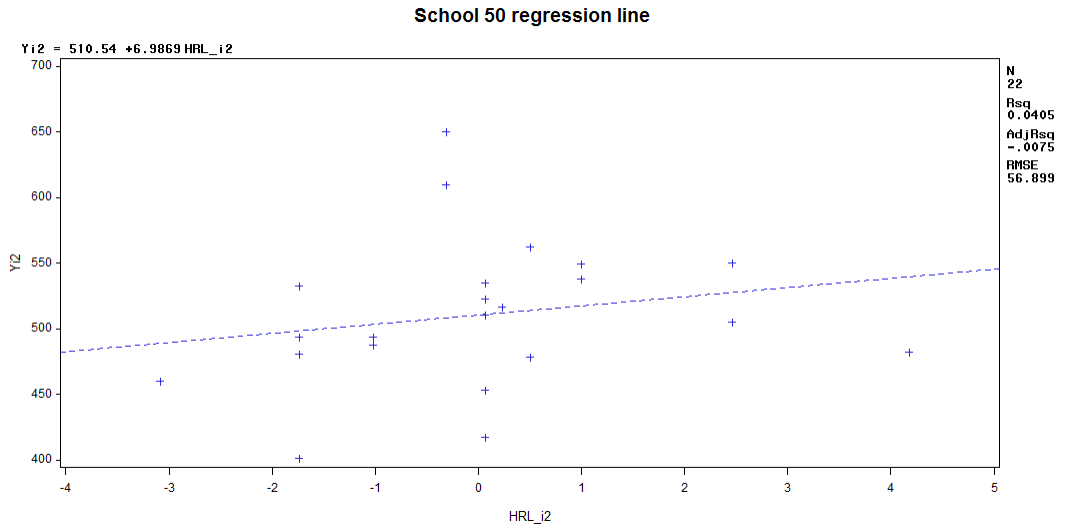


Figure 8: Regression line for school 50.

By making use of Figure 7 and Figure 8, two differences concerning the two schools can be observed. Firstly, school 50 has a higher overall mean of 510 in comparison to school 33 which had an overall mean 440. This can be seen by the intercept values ($\beta_{02} \geq \beta_{01}$) which can bring about the conclusion that school one is more effective than school two. Secondly, HRL has less of an influence on reading score in school 50, compared to school 33. This can be seen by school 50 having a slope of 7 and school 33 having a steeper slope of 50. Therefore, school 50 is not only more effective than school 33 but also more equitable.[4] Finally, the model concerning just one school can be easily generalized for j schools. Each school will have a unique intercept and slope specific to that corresponding school. Using the level one model, a pair of level two models can be presented to describe the level one parameter's variation[6]. The student level model is presented in Equation 6 followed by the school level models in Equation 7 and 8:

$$Y_{ij} = \beta_{0j} + \beta_{1j} HRL_{ij} + r_{ij} \tag{6}$$

$$\beta_{0j} = \gamma_{00} + u_{0j} \tag{7}$$

$$\beta_{1j} = \gamma_{10} + u_{1j} \tag{8}$$

$$\text{where } r_{ij} \sim N(0, \sigma^2) \text{ and } \begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix} \right]$$

.

With including the student predictor (HRL) which is a fixed effect, there is also an extra random effect (u_{1j}). This shows that there is not only a relationship between HRL and reading score but in addition that the relationship between HRL can vary between schools. Furthermore, with the addition of the level one predictor, this allows each school (j) to have an unique slope and intercept. This produces the usual variance components from the intercept and slope, in addition to a covariance component that shows the correlation between the intercept and slope [5]. For easier or more meaningful interpretation of the model, centering will be used. If the interpretation of β_{0j} in Equation 6 is considered across the entire sample, it can be seen that HRL has a mean of zero. Furthermore, β_{0j} will be interpreted as the mean resading score for a student with a average HRL[5]. Centering can be accomplished by subtracting the average HRL from each individual HRL score ($C\bar{H}RL$). The interpretation of β_{0j} now becomes the mean reading score for students. With the use of centering, the model can be re-written as follows:

$$Y_{ij} = \beta_{0j} + \beta_{1j} C\bar{H}RL_{ij} + r_{ij} \tag{9}$$

$$\beta_{0j} = \gamma_{00} + u_{0j} \tag{10}$$

$$\beta_{1j} = \gamma_{10} + u_{1j} \tag{11}$$

$$\text{where } r_{ij} \sim N(0, \sigma^2) \text{ and } \begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix}\right]$$

.

By subsituting Equation 10 and 11 into Equation 9 the combined model is produced:

$$Y_{ij} = \gamma_{00} + u_{0j} + \gamma_{10} C\bar{H}RL_{ij} + u_{1j} C\bar{H}RL_{ij} + r_{ij} \tag{12}$$

$$\text{where } r_{ij} \sim N(0, \sigma^2) \text{ and } \begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix}\right].$$

The model in Equation 12 now has two fixed effects and three random effects. Through the fixed and random effects it can be seen that the reading score of a student varies both between schools and within schools. The model has been written out in a format that groups that fixed effects and random effects for easier understanding. The fixed effects are grouped together on the left hand side and the random effects on the right hand side.

$$Y_{ij} = \left[\gamma_{00} + \gamma_{10} C\bar{H}RL_{ij}\right] + \left[u_{0j} + u_{1j} C\bar{H}RL_{ij} + r_{ij}\right]$$

Without knowing the interpretation of each of the variables, the model renders powerless. Therefore, it is of great importance to examine each of the variables:

- γ_{00} is the overall school mean reading score for all schools in the population
- γ_{10} is the average HRL-reading score slope
- u_{0j} is the random effect for school j associated with the mean reading score
- u_{1j} is the random effect for school j associated with the HRK-reading score slope
- r_{ij} is the error component for student i within school j

The difference between the combined model and the standard ANCOVA model is that u_{0j}, the random effect of school j is seen as a random effect instead of a fixed effect[4].

```
proc mixed data=schools.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=C_Home_resources_for_learning/solution ddfm=bw notest;
random intercept C_Home_resources_for_learning/sub=idschool type=un;
run;
```

Covariance Parameter Estimates					
Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z
UN(1,1)	IDSCHOOL	7258.70	1286.22	5.64	<.0001
UN(2,1)	IDSCHOOL	377.46	137.34	2.75	0.0060
UN(2,2)	IDSCHOOL	46.1306	24.1106	1.91	0.0279
Residual		5400.41	182.15	29.65	<.0001

Fit Statistics	
-2 Res Log Likelihood	21799.9
AIC (Smaller is Better)	21807.9
AICC (Smaller is Better)	21807.9
BIC (Smaller is Better)	21816.8

Null Model Likelihood Ratio Test		
DF	Chi-Square	Pr > ChiSq
3	1270.26	<.0001

Solution for Fixed Effects					
Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	432.68	10.4170	68	41.54	<.0001
C_Home_resources_for	7.8539	1.5029	1815	5.23	<.0001

Figure 9: Output for level one predictors

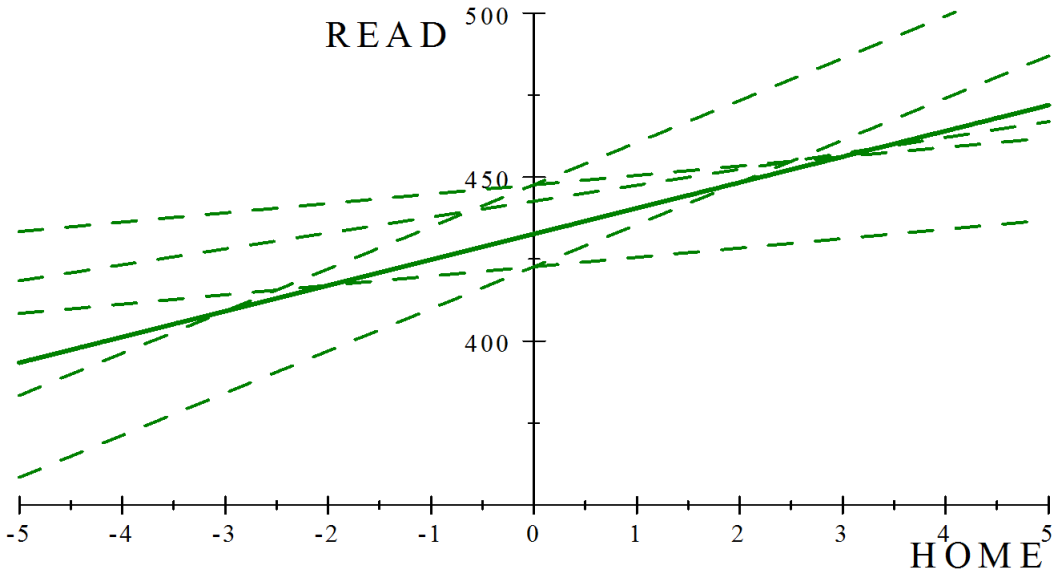


Figure 10: Regression line for model with HRL predictor

Holding all predictors equal to zero, the average reading score is 432.68. This is the intercept term and is

denoted in the model as γ_{00}. The relationship between the reading score and home resources for learning can be represented by γ_{01} . An increase in one unit of home resources for learning will increase the reading score by 7.8539 points. The standard errors for both of these terms are relatively small, creating large t-values and significant p-values. This allows us to conclude that there is a positive relationship between home resources for learning and reading score. The covariance component can be written in matrix notation:

$$\begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix} = \begin{pmatrix} 7258.70 & 377.46 \\ 377.46 & 46.13 \end{pmatrix}$$

As shown above in the covariance matrix, 7258.70 represents the variability in the intercepts, 46.1306 represents the variability in slopes and 377.46 represents the covariance between intercepts and slopes. The covariance output also provides tests of the null hypothesis that each covariance component equals 0. Since the intercepts are very variable, it can be concluded that even after controlling for the effects of HRL schools do differ in reading score. In order to find out how much of the within school variance in reading score is explained by HRL, the percentage can be computed using a simple formula as used previously. Please note that comparisons will be made to the 'Unconditional Means Model'. It was seen that for the Unconditional Means model, $\sigma^2 = 5616.76$. For the model with a level one predictor, the conditional estimate, $\sigma^2 = 5400.41$. By including the variable HRL it has explained 3.85 % of the explainable variation within schools. By comparison, it will be proved later that the school variable, emphasis on reading in early grades explains much more of the variation in school level reading score than 'HRL' explains in the within school variation in student level for reading score. This was calculated as follows:

$$\frac{5616.76 - 5400.41}{5616.76} = 0.0385$$

3.4 Model including the effects of level two predictors

3.4.1 Model including the effects of emphasis on reading in early grades

Up to this point characteristics associated with the student namely, HRL has been used as a predictor variable to measure the outcome variable. This section will focus on the use of school characteristics (level two) to help predict the outcome variable. The predictor variable that will be used will be 'EREG', which is the emphasis on reading in early grades that particular school enforces. Again, as in the previous section, the predictor variable EREG is centered around the overall mean ($CE\bar{R}EG$). This will be of help in interpreting the intercept. The model predicting the reading score as a function of EREG can be written as follows:

$$Y_{ij} = \beta_{0j} + r_{ij} \tag{13}$$

$$\beta_{0j} = \gamma_{00} + \gamma_{01} CE\bar{R}EG_j + u_{0j} \tag{14}$$

$$\text{where } r_{ij} \sim N(0, \sigma^2) \text{ and } u_{0j} \sim N(0, \tau_{00})$$

By substituting Equation 14 into Equation 13:

$$Y_{ij} = \gamma_{00} + \gamma_{01} CE\bar{R}EG_j + u_{0j} + r_{ij} \tag{15}$$

This model has two components, a fixed component and a random component. The fixed components consists of the first two terms and the random component consists of the last two terms. The random effects are denoted by σ^2 and τ_{00}, their corresponding variance components [5]. These variance components are the conditional variance in β_{0j} after controlling for EREG [4]. The random terms can be defined as follows; u_{0j} is the random variation in intercepts between schools and r_{ij} is the variation within the school [5].

```
proc mixed data=schools.new6 noclprint covtest;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades/solution ddfm=bw;
random intercept/sub=idschool;
run;
```

Covariance Parameter Estimates					
Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z
Intercept	IDSCHOOL	6377.35	1147.30	5.56	<.0001
Residual		5617.25	186.45	30.13	<.0001

Fit Statistics	
-2 Res Log Likelihood	21845.2
AIC (Smaller is Better)	21849.2
AICC (Smaller is Better)	21849.2
BIC (Smaller is Better)	21853.6

Solution for Fixed Effects					
Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	431.22	9.8020	67	43.99	<.0001
c_Emphasis_reading_e	12.5902	4.0098	67	3.14	0.0025

Figure 11: Output for level two predictor EREG

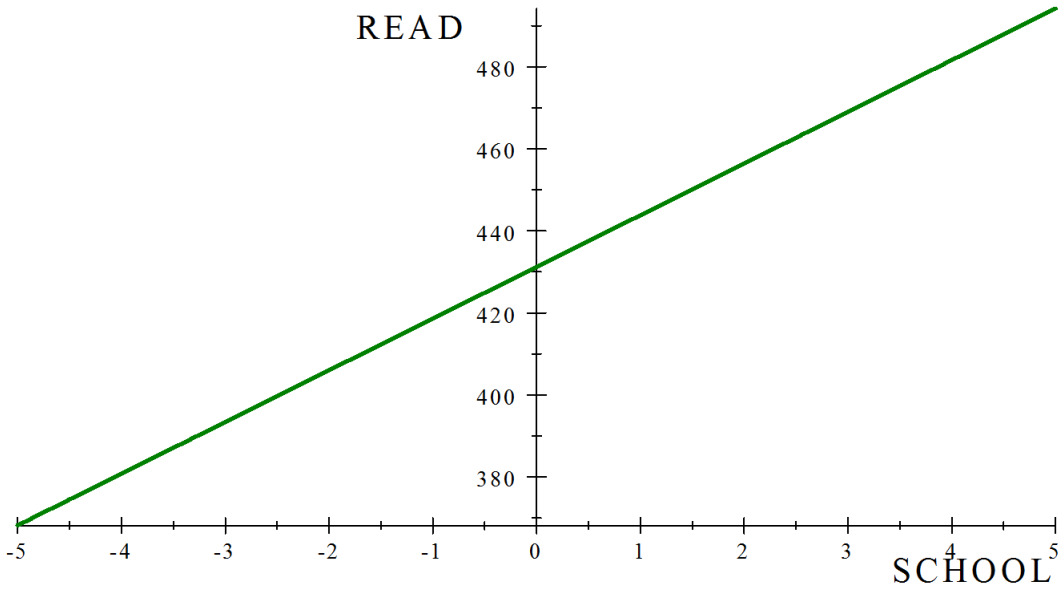


Figure 12: Regression line for model with EREG predictor

The average reading score in a particular school is 431.22, when holding EREG constant. This is the estimated value of the intercept γ_{00}. The term for the other fixed effect γ_{01} is the relationship between the reading score and the emphasis on reading in early grades. The estimate of γ_{01} is 12.59 which can be interpreted as follows; a unit increase of emphasis on reading in early grades will result in an increase of the reading score by 12.59. Both terms are significant, but the intercept has a greater standard error. Since EREG is centered around a grand mean, γ_{00} is the estimated reading score in a school of average EREG. The standard error for this term (EREG) produces a t-value of 3.14 and a corresponding p-value of 0.0025 which will reject the null hypothesis of no relationship between emphasis on reading in early grades and the

reading score of the students. Therefore, it can be concluded that there is a relationship between emphasis on reading in early grades and reading score. The covariance parameter estimates provide more information about the random effects, where τ_{00} can be estimated to be 6377.35 and σ^2 to be 5617.25. Despite the same symbols being used in the 'Unconditional Means' section, the interpretation of these terms are different. Note that comparisons made in this section will refer to estimates in the 'Unconditional Means' section. Now that there is a predictor within level two, these become conditional components. The estimate for σ^2 has only changed slightly (from 5616.76 to 5617.25), whereas the variance component τ_{00} has reduced substantially (from 7254.51 to 6377.35). It can be seen from this observation that a small percentage of the school-to-school variation in the average reading score can be explained by the predictor EREG. This percentage can be calculated as follows:

$$\frac{7254.51 - 6377.35}{7254.51} = 0.1209$$

This is interpreted as 12% of school-to-school variation in the average reading school can be attributed to EREG. Considering the data used has 92 school variables (for the purposes of this report only a few variables were selected) this is a reasonably high percentage for one variable to attain. Considering the variance component, σ^2 because there was only a very small increase in variance, there will be no percentage explained. But this makes sense, because σ^2 measures the within school variation (level 1) and nothing has changed within level one. The slight increase in variation can therefore be seen as random.

After explaining away 12% of the explain variation, it might be helpful to see if there is any variation within school means remaining that still needs to be explained. One way in which this can be done is through a simple hypothesis test. τ_{00} has a z-statistic of 5.56 and a corresponding p-value of $<.0001$. The null hypothesis stating that τ_{00} is 0, can be rejected. It can be concluded that there is still additional explainable variation present, despite EREG being included. A second way is to use the intraclass correlation coefficient to estimate the fraction of the sum of both variance components that occurs at school level. [5] This can be calculated as follows:

$$\rho = \frac{6377.35}{6377.35 + 5617.25}$$

$$\rho = 0.5317$$

This intraclass correlation can be seen as a partial correlation. It can be interpreted as the similarity of reading score among students within schools after controlling for the effect of EREG.

3.4.2 Model including the effects of a library

A second predictor within the school level can be examined. The covariate will be called 'LIB' and it specifies whether a school has an existing library. LIB will take on the value as 0 for a school with no existing library and 1 for a schools with an existing library.[5] A model containing only the LIB predictor in the school level is written as follows:

$$Y_{ij} = \beta_{0j} + r_{ij} \tag{16}$$

$$\beta_{0j} = \gamma_{00} + \gamma_{02} LIB_j + u_{0j} \tag{17}$$

$$\text{where } u_{0j} \sim N(0, \tau_{00})$$

The combined model in Equation 18 is found by substituting Equation 17 into Equation 16:

$$Y_{ij} = \gamma_{00} + \gamma_{02} LIB_j + u_{0j} + r_{ij} \tag{18}$$

```
proc mixed data=schools.new6 noclprint covtest;
class idschool;
model Reading_score=library/solution ddfm=bw;
random intercept/sub=idschool;
run;
```

Covariance Parameter Estimates					
Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z
Intercept	IDSCHOOL	6081.38	1093.89	5.56	<.0001
Residual		5616.60	186.41	30.13	<.0001

Fit Statistics	
-2 Res Log Likelihood	21838.5
AIC (Smaller is Better)	21842.5
AICC (Smaller is Better)	21842.5
BIC (Smaller is Better)	21847.0

Solution for Fixed Effects					
Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	375.18	18.2514	67	20.56	<.0001
library	79.2630	21.4341	67	3.70	0.0004

Figure 13: Output for level two predictor LIB

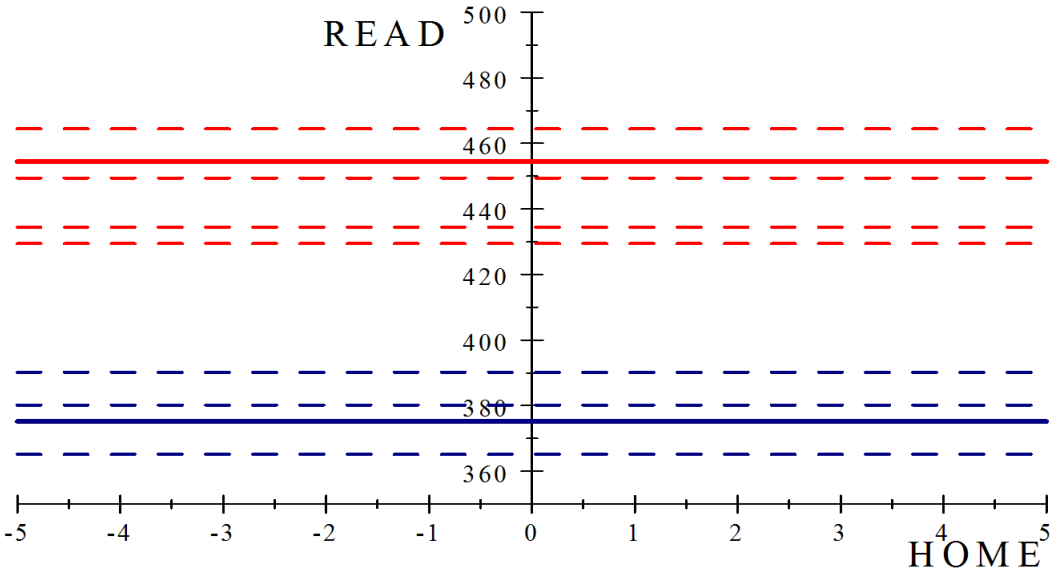


Figure 14: Effect of LIB predictor on the intercept

The average reading score of a student at a school that does not have an existing library is 375.18. This can be compared to a student at a school with an existing library who has average reading score 454.44.

The covariance parameter estimates provide more information about the random effects, where τ_{00} can be estimated to be 6081.38 and σ^2 to be 5616.60. Again note that comparisons made in this section will refer to estimates in the 'Unconditional Means' section. The estimate for σ^2 has only changed slightly (from 5616.76 to 5616.60) as was the case when EREG was explained. On the other hand, the variance component τ_{00} has reduced slightly more than when EREG was explained (from 7254.51 to 6081.38). It can be seen from this observation that 16% of the school-to-school variation in the average reading score can be explained by the predictor LIB. This percentage can be calculated as follows:

$$\frac{7254.51 - 6081.38}{7254.51} = 0.1617$$

3.4.3 Model including the effects of a library and the emphasis on reading in early grades

The first and second level equation for the intercept including both school predictors will look as follows:

$$Y_{ij} = \beta_{0j} + r_{ij} \tag{19}$$

$$\beta_{0j} = \gamma_{00} + \gamma_{01} C\bar{E}REG_j + \gamma_{02} LIB_j + u_{0j} \tag{20}$$

The combined model is obtained by substituting Equation 20 into Equation 19:

$$Y_{ij} = \gamma_{00} + \gamma_{01} C\bar{E}REG_j + \gamma_{02} LIB_j + u_{0j} + r_{ij} \tag{21}$$

$$\text{where } r_{ij} \sim N(0, \sigma^2) \text{ and } u_{0j} \sim N(0, \tau_{00})$$

```
proc mixed data=sasuser.new6 noclprint covtest;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library/solution ddfm=bw;
random intercept/sub=idschool;
run;
```

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z
Intercept	IDSCHOOL	5360.19	978.37	5.48	<.0001
Residual		5617.03	186.43	30.13	<.0001

Fit Statistics

-2 Res Log Likelihood	21825.1
AIC (Smaller is Better)	21829.1
AICC (Smaller is Better)	21829.1
BIC (Smaller is Better)	21833.5

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	377.91	17.2026	66	21.97	<.0001
c_Emphasis_reading_e	11.4052	3.7027	66	3.08	0.0030
library	73.7026	20.2546	66	3.64	0.0005

Figure 15: Output for level two predictors EREG and LIB

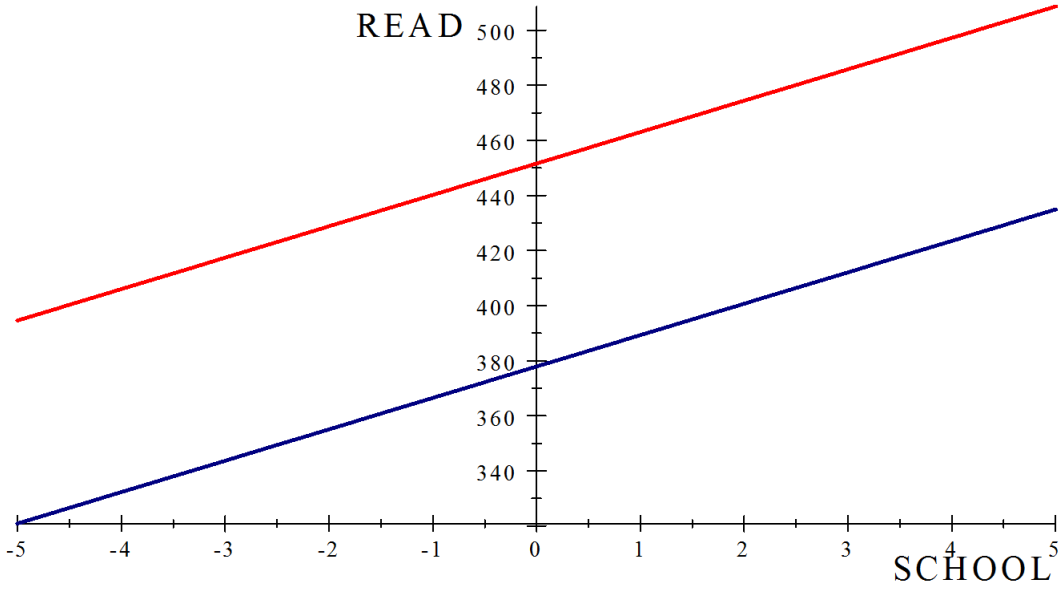


Figure 16: Regression line for model with EREG and LIB predictors

The interpretation of the following terms will be considered:

- γ_{00} is the overall reading score for all schools without an existing library;
- γ_{01} is the relationship between reading score and emphasis on reading in early grades in school j;
- γ_{02} is the overall difference in reading scores between schools with and without an existing library;
- u_{0j} is the random effect of school j's average reading score, given LIB was held constant.[4]

A school with a library will have a intercept that is 73.7 points higher than a school without a library. For all schools, the slope (EREG) will be the same, producing parallel lines. A student at a school, whether the school has a library or not will have a slope of 11.4. All terms in this model are seen to be significant. The variance and covariance components now become conditional variance-covariance components as they denote the variability in β_{0j} and β_{1j} after holding LIB constant [4].

3.5 Modeling including the effects of both level one and level two predictors

3.5.1 Full model including all possible predictors

After studying the models with a student predictor and a school predictor separately, a model with both predictors can be considered. To help with the understanding of the model, it will be expressed as separate models at level one (Equation 22) and level two (Equation 23 and 24) and then will be combined to give the final model (Equation 25). The separate models are as follows:

$$Y_{ij} = \beta_{0j} + \beta_{1j}C\bar{H}RL_{ij} + r_{ij} \tag{22}$$

$$\beta_{0j} = \gamma_{00} + \gamma_{01}C\bar{E}REG_j + \gamma_{02}LIB_j + u_{0j} \tag{23}$$

$$\beta_{1j} = \gamma_{10} + \gamma_{11}C\bar{E}REG_j + \gamma_{12}LIB_j + u_{1j} \tag{24}$$

$$\text{where } r_{ij} \sim N(0, \sigma^2) \text{ and } \begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix}\right]$$

These two separate models are combined to illustrate the full model including all possible level one and level two predictors. However, due to the insignificance of certain terms which can be seen in Figure 17 this is not the final model that will be used in the application. This model is provided for theoretical and consistency purposes only.

$$Y_{ij} = \gamma_{00} + \gamma_{01}C\bar{E}REG_j + \gamma_{02}LIB_j + \gamma_{10}C\bar{H}RL_{ij} + \gamma_{11}C\bar{E}REG_jC\bar{H}RL_{ij}$$

$$+\gamma_{12}LIB_jC\bar{H}RL_{ij} + u_{0j} + u_{1j}C\bar{H}RL_{ij} + r_{ij} \tag{25}$$

```
proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library C_Home_resources_for_learning
c_Emphasis_reading_early_grades*C_Home_resources_for_learning
library*C_Home_resources_for_learning / solution ddfm=bw notest;
random intercept C_Home_resources_for_learning/sub=idschool type=un;
run;
```

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z
UN(1,1)	IDSCHOOL	5369.28	978.20	5.49	<.0001
UN(2,1)	IDSCHOOL	244.54	115.58	2.12	0.0344
UN(2,2)	IDSCHOOL	39.6760	24.3349	1.63	0.0515
Residual		5402.30	182.36	29.62	<.0001

Fit Statistics

-2 Res Log Likelihood	21760.5
AIC (Smaller is Better)	21768.5
AICC (Smaller is Better)	21768.5
BIC (Smaller is Better)	21777.4

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
3	982.30	<.0001

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	377.85	17.2019	66	21.97	<.0001
c_Emphasis_reading_e	11.3747	3.7028	66	3.07	0.0031
library	73.8113	20.2540	66	3.64	0.0005
C_Home_resources_for	4.9778	2.7582	1813	1.80	0.0713
c_Emphasi*C_Home_res	0.9394	0.6081	1813	1.54	0.1226
library*C_Home_resou	3.9063	3.2749	1813	1.19	0.2331

Figure 17: Output of full model.

3.5.2 Model without any interaction effects

A simple model without interaction effects is given as follows:

$$Y_{ij} = \beta_{0j} + \beta_{1j}C\bar{H}RL_{ij} + r_{ij} \tag{26}$$

$$\beta_{0j} = \gamma_{00} + \gamma_{01}C\bar{E}REG_j + \gamma_{02}LIB_j + u_{0j} \tag{27}$$

$$\beta_{1j} = \gamma_{10} + u_{1j} \tag{28}$$

Level 1 model (Equation 26) and level 2 models (Equation 27 and 28) are combined to produce:

$$Y_{ij} = \gamma_{00} + \gamma_{01}C\bar{E}REG_j + \gamma_{02}LIB_j + \gamma_{10}C\bar{H}RL_{ij} + u_{0j} + u_{1j}C\bar{H}RL_{ij} + r_{ij} \tag{29}$$

where $r_{ij} \sim N(0, \sigma^2)$ and $u_{0j} \sim N(0, \tau_{00})$

```
proc mixed data=schools.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library
C_Home_resources_for_learning  /solution ddfm=bw notest;
random intercept Home_resources_for_learning/sub=idschool type=un;
run;
```

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z
UN(1,1)	IDSCHOOL	5383.79	982.80	5.48	<.0001
UN(2,1)	IDSCHOOL	253.46	120.52	2.10	0.0355
UN(2,2)	IDSCHOOL	47.8357	24.9467	1.92	0.0276
Residual		5397.86	182.08	29.65	<.0001

Fit Statistics

-2 Res Log Likelihood	21769.7
AIC (Smaller is Better)	21777.7
AICC (Smaller is Better)	21777.8
BIC (Smaller is Better)	21786.7

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
3	982.75	<.0001

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	382.94	16.7541	66	22.86	<.0001
c_Emphasis_reading_e	9.8926	3.5713	66	2.77	0.0073
library	67.0280	19.5132	66	3.44	0.0010
C_Home_resources_for	7.7435	1.5204	1815	5.09	<.0001

Figure 18: Output for the combined model without interaction effects

The fitted models can be written as follows;
School without an existing library:

$$Readingscore = 382.94 + 9.89C\bar{E}REG + 7.74C\bar{H}RL$$

School with an existing library:

$$Readingscore = 449.97 + 9.89C\bar{E}REG + 7.74C\bar{H}RL$$

A dummy variable has been added to this model. It represents a 1 if there is an existing school library and 0 if there is no existing school library. Through the effect of the variable 'LIB', it can be seen that the intercepts in the two models above differ significantly. Basic interpretations of the estimates can be made as follows:

- The average reading score of a school without an existing library is 382.94.
- The average reading score of a school with an existing library is 449.97.

3.5.3 Final model

A model was fitted with an interaction effect to explain the variation in the trend in EREG and HRL. For the purposes of this report, this interaction effect was removed as it was not significant and did not contribute to a better model. The final model that has been chosen for the purposes of application contains one interaction effect between HRL and LIB. The interaction effect between HRL and EREG was tested and found to be insignificant, hence it was removed from the model. The final model is given as follows:

$$Y_{ij} = \beta_{0j} + \beta_{1j}C\bar{H}RL_{ij} + r_{ij} \tag{30}$$

$$\beta_{0j} = \gamma_{00} + \gamma_{01}C\bar{E}REG_j + \gamma_{02}LIB_j + u_{0j} \tag{31}$$

$$\beta_{1j} = \gamma_{10} + \gamma_{12}LIB_j + u_{1j} \tag{32}$$

Level 2 models (Equation 31 and 32) are substituted into level 1 model (Equation 30) to produce the final combined model in Equation 33:

$$Y_{ij} = \gamma_{00} + \gamma_{01}C\bar{E}REG_j + \gamma_{02}LIB_j + \gamma_{10}CH\bar{R}L_{ij}$$

$$+\gamma_{12}LIB_jCH\bar{R}L_{ij} + u_{0j} + u_{1j}CH\bar{R}L_{ij} + r_{ij} \tag{33}$$

```
proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library
C_Home_resources_for_learning
library*C_Home_resources_for_learning /solution ddfm=bw notest;
random intercept C_Home_resources_for_learning/sub=idschool type=un;
run;
```

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z
UN(1,1)	IDSCHOOL	5377.91	980.82	5.48	<.0001
UN(2,1)	IDSCHOOL	258.80	119.49	2.17	0.0303
UN(2,2)	IDSCHOOL	47.5799	24.7990	1.92	0.0275
Residual		5396.54	182.00	29.65	<.0001

Fit Statistics

-2 Res Log Likelihood	21763.6
AIC (Smaller is Better)	21771.6
AICC (Smaller is Better)	21771.6
BIC (Smaller is Better)	21780.6

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
3	983.40	<.0001

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	377.51	17.2131	66	21.93	<.0001
c_Emphasis_reading_e	9.8932	3.5631	66	2.78	0.0071
library	74.5227	20.2633	66	3.68	0.0005
C_Home_resources_for	4.4992	2.8238	1814	1.59	0.1113
library*C_Home_resou	4.5618	3.3488	1814	1.36	0.1733

Figure 19: Output for final model

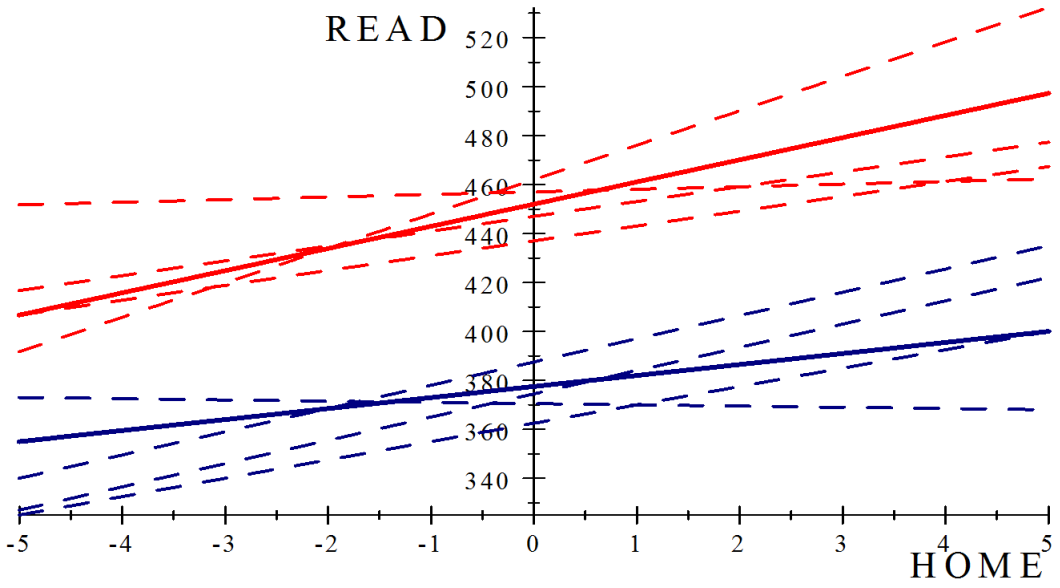


Figure 20: Regression line for HRL and LIB predictors and holding EREG constant

Since the variable LIB is a dummy variable indicating whether a school has an existing library, it can be re-written with a pair of models.The final models can be fitted as follows:

School without an existing library:

$$Readingscore = 377.51 + 9.89C\bar{E}REG + 4.50C\bar{H}RL$$

School with an existing library:

$$Readingscore = 452.03 + 9.89C\bar{E}REG + 9.06C\bar{H}RL$$

All fixed effects are substantially significant for the purposes of this report. The output of the effect "library" indicates that the intercepts of the two models (with or without an existing library) differ significantly. The average reading score for a student at a school with no existing library is 377.51, whereas a student at a school with an existing library has a reading score of 452.03. A library is seen to play a big role within a school when reading is concerned as it alone increases the reading score by 74 points. The interaction effect between library and home resources for learning indicates that there are different slopes for schools with a library and schools without a library. For a school with an existing library, the slope for HRL is much steeper than for a school without an existing library. This demonstrates the importance of a school library. HRL accentuates a student's reading score to a greater extent in a school with a library. This comparison is made by observing the difference between the two coefficients of HRL, namely 4.50 and 9.06. As mentioned previously, the interaction effect between HRL and EREG was found to be insignificant. The variance component of the intercept (τ_{00}) is still significant (p-value $<.0001$) which means that there is still additional variation in average school reading scores that has yet to be explained. This suggests that there are still remaining school variables that would explain the variation in school means. The covariance matrix can be written as follows:

$$\begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix} = \begin{pmatrix} 5377.91 & 258.8 \\ 258.8 & 47.57 \end{pmatrix}$$

There has been a decrease in the variance in intercepts and the covariance between intercepts and slopes but there has been very slight increase in the variance of the slopes if compared to the covariance matrix we observed with level one predictors. It can be stated again that the intercepts are very variable. Schools differ in their average reading score even after controlling for effects of HRL, EREG and LIB. A simpler model

could be implemented, in which intercepts vary across schools but the slopes do not. It would be wise to compare fit statistics to determine if the simpler model fits the data best. The model will be fit as follows:

```
proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library
C_Home_resources_for_learning library*C_Home_resources_for_learning /solution ddfm=bw notest;
random intercept /sub=idschool;
run;
```

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z
UN(1,1)	IDSCHOOL	5366.29	978.48	5.48	<.0001
Residual		5495.50	182.50	30.11	<.0001

Fit Statistics

-2 Res Log Likelihood	21777.3
AIC (Smaller is Better)	21781.3
AICC (Smaller is Better)	21781.3
BIC (Smaller is Better)	21785.7

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
1	969.75	<.0001

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	377.91	17.2037	66	21.97	<.0001
c_Emphasis_reading_e	11.4040	3.7031	66	3.08	0.0030
library	73.7105	20.2561	66	3.64	0.0005
C_Home_resources_for	5.1343	2.2976	1814	2.23	0.0256
library*C_Home_resou	3.6969	2.7162	1814	1.36	0.1737

Figure 21: Model with random intercepts only

The comparison of the fit statistics between a model with random intercepts and slopes and a model with random intercepts only can be made to conclude which model fits the data best. For the ease of comparison, the fits statistics of both models are provided:

	AIC	AICC	BIC
Random intercepts and slopes	21771.6	21771.6	21780.6
Random Intercepts	21781.3	21781.3	21785.7

Table 1: Comparison of Fit Statistics

Recalling that a smaller value for all the methods of fit statistics, it can be concluded that the model with random intercepts and slopes (a less restricted model) provides a better fit.

4 Conclusion

With the increase in data that has a nested hierarchy, there is a need for a technique that accommodates the dependence within hierarchical levels. This study revealed the variables, both on school and student level that had a impact on the reading score of a South African student. Upon examining the Unconditional Means Model, it was concluded that majority (56%) of the variation came from the between schools variation. This highlights the fact that every school has it's own unique intercept and slope. While studying a model with solely student predictors(level one) a high variance in the intercepts was observed. It was concluded that even after controlling for the effects of home resources schools do differ in reading score. The student predictor, "home resources for learning" explained 3.85 % of the explainable variation within schools. In comparison, the school variable, "emphasis on reading in early grades" explained 12% of the school to school variation. With majority of the variation stemming from between schools and the school predictor explaining away a

relatively high percentage of variation, it can be concluded that the school with which a student goes to has a greater impact than the resources available to the student in the home environment. It was noted that despite emphasis on reading in early grades accounted for a significant amount of the school-to-school variation, there is still remaining explainable variation. Upon fitting multiple models including both level one and two variables, it was decided that the model in Figure 11 with random intercepts and slopes would be used. The positive impact on a school having a library was proved. A student at a school with a library is most likely to score 74 points higher than a student at a school without a library. Multi-level modeling is a brilliant technique that accommodates the hierarchical structure and the dependence across all levels of the hierarchy. In the future, further studies could be done to make techniques more approachable for researchers and gain popularity. With the increase in data accessibility, more sophisticated models can be developed and implemented from a theoretical sense to a practical use.

References

[1] B Bell, M Ene, and J Schoeneberger. A multilevel model primer using sas proc mixed. In *SAS Global Forum*, pages 0–19. Citeseer, 2013.

[2] H Goldstein. Multilevel statistical models. *Journal of the American Statistical Association*, 100:354–355, 2005.

[3] S Howie, S Van Staden, M Tshele, C Dowse, and L Zimmerman. Pirls 2011: South african children's reading literacy achievement summary report. *Pretoria: University of Pretoria*, 2012.

[4] S W Raudenbush and A S Bryk. *Hierarchical Linear Models: Applications and Data Analysis Methods*, volume 1. Sage, 2002.

[5] J D Singer. Using sas proc mixed to fit multilevel models, hierarchical models, and individual growth models. *Journal of Educational Psychology and Behavioral Statistics*, 23(4):323–355, 1998.

[6] J D Singer. Fitting individual growth models using sas proc mixed. *Modeling Intraindividual Variability with Repeated Measures Data: Methods and Applications*, pages 135–170, 2002.

Appendix

```
proc contents data=sasuser.student;
run;

proc freq data=sasuser.student;
 tables ASBG01;
 run;

 proc means data=sasuser.student;
 class asbg01;
 var asrrea01;
 run;

 data sasuser.new;
 merge sasuser.student (keep=idschool idstud asbg01 asbg04 asbg05a asbg05b asbg05c asbg05e asrrea01 asbg
           sasuser.school(keep=idschool acbg09 acbg10ag acbg05c acbg11cd acbg12f acbgrss acbgeas)
           ;
 by idschool;
 run;

  data sasuser.new2;
 merge sasuser.new
           sasuser.parent (keep=idstud asbh02b asbh09b asbh17a asbh17b)
           ;
 by idstud;
 run;

 proc contents data=sasuser.new3;
 run;

 data sasuser.final;
 set sasuser.new2;
 Gender=asbg01;
 Books_at_home=asbg04;
 Student_owns_computer=asbg05a;
 Student_owns_desk=asbg05b;
 Student_owns_books=asbg05c;
 Student_access_internet=asbg05e;
 Reading_score=asrrea01;
 Home_resources_for_learning=asbghrl;
 Student_enjoys_reading=asbgslr;
 Existing_school_library=acbg09;
 School_computers_for_instruction=acbg10ag;
 Average_income_level_for_area=acbg05c;
 School_Rules=acbg11cd;
 Parental_involvement_at_school=acbg12f;
 Parents_tell_stories=asbh02b;
 Parents_help_with_hw=asbh09b;
 Level_education_father=asbh17a;
 Level_education_mother=asbh17b;
 Emphasis_reading_early_grades=acbgrss;
 Emphasis_on_academic_sucess=acbgeas;
```

```
 run;

data sasuser.new3;
set sasuser.final;
gender=asbg01;
if acbg09="YES" then library=1;
if acbg09="NO *(IF NO, GO TO #10)*" then library=0;
*if library=. then delete;
*if gender=. then delete;
run;

proc univariate data=sasuser.new3 freq;
var Library;
run;

proc freq data=sasuser.new3;
 tables library;
run;

proc means data=sasuser.new3;
class Emphasis_on_academic_sucess;
var Reading_score;
run;

proc means data=sasuser.new6;
class Parents_help_with_hw;
var Reading_score;
run;

ods graphics on;
proc corr data=sasuser.new3 plots=scatter;
var Reading_score Emphasis_on_academic_sucess ;
run;
ods graphics off;

data sasuser.plots;
set sasuser.new3 (obs=100);
run;

proc reg data=sasuser.new3;
model Reading_score=Home_resources_for_learning;
plot Reading_score*Home_resources_for_learning;
run;

proc reg data=sasuser.new3;
model Reading_score=Emphasis_reading_early_grades;
plot Reading_score*Emphasis_reading_early_grades;
run;

data sasuser.test;
set sasuser.new3;
if idschool=19 then keep idschool;
run;
```

```
proc means data=sasuser.new3;
class idschool;
var Home_resources_for_learning;
output out=center;
run;

proc sort data=center;
by _freq_;
run;

proc univariate data=center;
var idschool Home_resources_for_learning;
run;

proc means data=sasuser.new4;
class idschool;
var Home_resources_for_learning;
output out=try mean=average;
run;

proc corr data=try;
var _freq_ average;
run;

proc univariate data=try freq;
var average;
run;

proc sort data=try;
by _freq_;
run;

data sasuser.new4;
set sasuser.new3;
if  nmiss (Home_resources_for_learning, Emphasis_reading_early_grades)=0;
run;


data sasuser.new5;
merge sasuser.new4
       try
           ;
 by idschool;
 run;

 data sasuser.new6;
 set sasuser.new5;
 C_Home_resources_for_learning=Home_resources_for_learning-average;
 run;

 proc means data=sasuser.new6;
 class library;
```

```
 var Home_resources_for_learning Emphasis_reading_early_grades;
 run;

proc means data=sasuser.new6;
class library;
var Reading_score;
run;

proc means data=sasuser.new6;
var Emphasis_reading_early_grades;
output out=mich;
run;

data sasuser.new6;
set sasuser.new6;
c_Emphasis_reading_early_grades=Emphasis_reading_early_grades-9.1048611;
run;

proc univariate data=sasuser.new6;
var C_Home_resources_for_learning;
run;


/*Unconditional Means*/
proc mixed data=sasuser.new6 noclprint covtest;
class idschool;
model Reading_score= /solution;
random intercept/sub=idschool;
run;

/*Level two predictors*/
proc mixed data=sasuser.new6 noclprint covtest;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades/solution ddfm=bw;
random intercept/sub=idschool;
run;

proc mixed data=sasuser.new6 noclprint covtest;
class idschool;
model Reading_score=library/solution ddfm=bw s;
random intercept/sub=idschool s;
run;

proc mixed data=sasuser.new6 noclprint covtest;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library/solution ddfm=bw;
random intercept/sub=idschool;
run;

/*Level one predictors*/
proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=C_Home_resources_for_learning/solution ddfm=bw notest;
```

```
random intercept C_Home_resources_for_learning/sub=idschool type=un;
run;

/*Level one and two predictors*/
/*useeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee*/
proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library C_Home_resources_for_learning
library*C_Home_resources_for_learning / solution ddfm=bw notest;
random intercept C_Home_resources_for_learning/sub=idschool type=un;
run;

/************************/


proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library C_Home_resources_for_learning / solution ddf
random intercept C_Home_resources_for_learning/sub=idschool type=un;
run;



proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library C_Home_resources_for_learning
c_Emphasis_reading_early_grades*C_Home_resources_for_learning library*C_Home_resources_for_learning / s
random intercept C_Home_resources_for_learning/sub=idschool type=un;
run;

/*simpler model*/
proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library C_Home_resources_for_learning
library*C_Home_resources_for_learning / solution ddfm=bw notest;
random intercept /sub=idschool type=un;
run;


proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library C_Home_resources_for_learning
library*C_Home_resources_for_learning / solution ddfm=bw notest;
random intercept /sub=idschool type=un;
run;
/*final model */
proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library C_Home_resources_for_learning /solution ddfm
random intercept C_Home_resources_for_learning/sub=idschool;
run;

proc mixed data=sasuser.new6 noclprint covtest noitprint;
```

```
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library C_Home_resources_for_learning /solution ddfm
random intercept /sub=idschool;
run;

/*new*/
proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library C_Home_resources_for_learning library*C_Home
random intercept /sub=idschool;
run;

proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=c_Emphasis_reading_early_grades library C_Home_resources_for_learning library*C_Home
random intercept C_Home_resources_for_learning/sub=idschool;
run;

proc reg data=sasuser.new6;
model Reading_score=C_Home_resources_for_learning;
by idschool;
/*where idschool in(33,34);*/
run;

proc freq data=sasuser.new6;
tables idschool;
output out=regression;
run;

proc print data=sasuser.new6;
var Reading_score C_Home_resources_for_learning;
where idschool=50;
run;

proc reg data=plot1;
model Yi1=hrl_i1;
plot yi1*hrl_i1;
Title "School 33 regression line";
run;

proc reg data=plot1;
model Yi2=hrl_i2;
plot yi2*hrl_i2;
Title "School 50 regression line";
run;


proc reg data=sasuser.new6;
model Reading_score=Emphasis_reading_early_grades library C_Home_resources_for_learning;
output out=stats;
run;

proc means data=sasuser.new6;
```

```
class idschool;
run;

ods graphics on;
proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=Emphasis_reading_early_grades library C_Home_resources_for_learning
library*C_Home_resources_for_learning / solution ddfm=bw notest outp=try;
random intercept C_Home_resources_for_learning/sub=idschool type=un;
run;
ods graphics off;

proc reg data=line2;
*model reading_score=home_resources_for_learning;
model reading_score_lib_=home_resources_for_learning_lib_;
plot reading_score_lib_*home_resources_for_learning_lib_;
run;


proc reg data=line2;
model reading_score=home_resources_for_learning;
plot reading_score*home_resources_for_learning;
run;

proc mixed data=sasuser.new6 noclprint covtest noitprint;
class idschool;
model Reading_score=Emphasis_reading_early_grades library C_Home_resources_for_learning
library*C_Home_resources_for_learning / solution ddfm=bw notest;
random intercept C_Home_resources_for_learning/sub=idschool type=un;
run;
```

Topic modeling on short text with emoticons

Mwila Chikonde 13146476

WST795 Research Report

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

Supervisor: Dr Alta de Waal, Co-supervisor: Jocelyn Mazarura

Department of Statistics, University of Pretoria

2 November 2016

Abstract

The vast use of social media in this day and age presents a great opportunity to mine large scale data on public opinions. As people increasingly use emoticons in text on social media sites to express themselves, there has arisen a great need for these sentiments to be analyzed, because this could possibly provide a link between understanding behavior and sentiment analysis. Sentiment analysis is a type of supervised learning involving labeled data. Labeling tends to be expensive and time consuming, added to that labeled copora tend to be highly contextualized. This means a labeled corpus used for politics can't be used for consumer analysis.

The goal of this report is to understand the behavior of topic models on short text with emoticons. A labeled corpus will be analyzed to investigate the coherence of topics for which emoticons with high probabilities have sentiment labels. This should provide an indication on the feasibility of emoticon topic modeling and whether it can be used to enhance sentiment analysis.

Declaration

I, *Mwila Chikonde,* 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.

_____
MWILA CHIKONDE

_____
Dr Alta de Waal

_____
2/11/2016

Acknowledgements

I would like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a post graduate bursary. I would also like to thank my supervisors Dr Alta de Waal and Ms Jocelyn Mazarura 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

1 Introduction **6**

2 Literature Review **6**

3 Background Theory **7**
3.1 Notation and Terminology 7
3.2 Latent Dirichlet Allocation 7
3.3 The Gibbs Sampler 8

4 Data **9**
4.1 Weather Data Set 9

5 Application **9**
5.1 Preprocessing 9
5.2 Experimental Design 10

6 Results **10**
6.1 Weather Data Set 10

7 Conclusion **12**

Appendix **14**

List of Figures

1 Word and Topic Matrix 6

List of Tables

1 Emoticons Table 9
2 Emoticon Statistics 9
3 Preprocessing 10
4 Example of output 10
5 Actual output for weather data set 11

1 Introduction

Topic modeling is a text mining technique, that unearths hidden topics from a large corpus of documents [6]. This paper will explore topic modeling for short text that contains emoticons. Emoticons are a proxy for emotions that people use to express sentiment in a post on social media in the body of a text . This can be classified as an emotional signal [4]. In a natural setting, sentiments can easily be detected by observing the behavior of a person, whether it be a smile or a frown. However, with computer-mediated communication in plain text, ocular prowess counts for nothing. This is where emoticons really come into the fray. These emoticons play the role of visual cues in texts and for this setting, they replace the ordinary physical cues such as a smile or an expression of stress [3]. An emoticon can be read sideways, like :-(, this is considered to be the expression for a sad face [3].

In this paper an alternative approach to supervised sentiment classification is investigated, in incorporating emoticons into topic models. A topic model is deployed on a labeled text file in order to derive topics and usually in doing so special characters and stop words are removed. In this case however emoticons are kept because the idea behind this approach is to investigate the coherence between topics for which emoticons have high probabilities and sentiment labels.

The structure of a topic model is considered to be as follows: A topic model factorizes a word x document matrix into topic x document and word x topic matrices. The matrices represent probabilities and the words associated with high probabilities in a topic vector provide a good description of that topic, as can be seen in figure 1 .

Water
Summer
Beach
Bikini
Party
Blue
Surfing
Sand
Holiday
Drinks

Figure 1: Word and Topic Matrix

In a similar way the doc x topic matrix can be sorted and documents associated with high probabilities in a topic vector should all have similar content. If tweets(documents) associated with high probabilities in this topic overlap significantly with tweets labeled as positive, then it is a good indication that topic models with emoticons can bootstrap sentiment classification. One of the data sets discussed in this paper is labeled with sentiment labels which provides the opportunity to investigate this modeling approach.

2 Literature Review

In a world where more than a billion people use social media, text data mining techniques have widespread applications, in trying to gain a perspective into what knowledge is in texts. This paper focuses on opinion mining, and the aim is to understand the feelings people use in text based on emoticons expressed in them. Lin and He [5] propose using a novel probabilistic modeling framework based on LDA [5] in order to conduct this analysis. To fully understand LDA, Blei's [2] article will be employed. The benefits of mining and quantifying sentiments are huge and an example would be the business sector. In such a setting a possible application of this would be to find out customers possible opinions on a certain product or topic, often expressed in text. The text analyzed is from micro blogging sites such as Twitter and Facebook [1]. In order to garner information from short text, topic modeling has proven to be instrumental in automatic discovery of thematic information. Gaining knowledge on topic modeling for short text will require the article done

by Mazarura [6]. It is possible to gather insights from a large archive of documents. The way LDA works is that documents will be viewed as a mixture of probabilistic topics, where a topic is a probability distribution over words [8]. Within these components, certain structures in the document can be inferred by standard statistical inference.

3 Background Theory

In order to carry out the analysis different topic models can be employed. One popular one that is used is Latent Dirichlet Allocation(LDA), proposed by David Blei et al [2]. Latent Dirichlet allocation is a generative probabilistic model for collections of discrete data such as text corpora.

One of the main topic modeling techniques that will be implemented for this paper will be the LDA topic modeling technique. This is a probabilistic model that forms collections of data that are discrete, such as text corpora. It is a three-level Bayesian model in which an item is modeled over a finite mixture of underlying topics. These topics are modeled over an infinite mixture underlying a set of topic probabilities, which provide a very good representation of a document. The general idea is that a document could possibly be represented as a random mixture over latent topics, where a topic would be characterized by a distribution over words [2].

3.1 Notation and Terminology

Some notation and terminology that will be used in this report is listed as follows:

- **Emoticon:** used to express emotion on social media and text. Combination of special characters used to express emotion e.g:-), :-(.
- **Word:** In this setting is a unit of discrete data.
- **Stop Words:** These are words such as the, and, but, for, because, since etc.
- **Document:** Is a sequence of N words, and a document is denoted by $\mathbf{a}$= **{** a_1**,** a_2**,.....,**a_N**},** a_n is the nth word.
- **Special Characters:** symbols such as #,*,& would fall into this category.
- **Corpus:** Is a collection of M Documents, its is denoted as D=**{**$\mathbf{a}_1$**,....,**$\mathbf{a}_M$**}.**

3.2 Latent Dirichlet Allocation

Topic modeling is a text mining technique, that unearths hidden topics from a large corpus of documents [6]. The method of topic modeling that will be used in this report is Latent Dirichlet Allocation(LDA) . In LDA the basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words[2]. Each word belongs to a certain topic drawn from a specific distribution [7].

With LDA the following assumptions are made for each document $\mathbf{a}$ in a corpus D**[2]**.

1. Choose N~Poi(δ)
2. Choose Θ~Dir(α)
3. For each of the N words in a_n:

- Choose a topic y_n~Multinomial(Θ)
- Choose a word a_n from p(a_n|y_n,β) Multinomial probability conditioned on the topic.

As proposed by David Blei et al [2] in order for the basic model to work there are several simplifying assumptions made. Firstly the N is independent of θ and $\mathbf{y}$. Secondly we assume the dimensionality of k for the Dirichlet distribution is fixed and known. Lastly β is a k×V matrix that Parameterizes the word probabilities. Where β_{ij}= p(a^j=1| y^i=1) and in this case we treat it as a fixed quantity.

For a k dimensional vector Dirichlet random variable Θ, the k vector lies within a (k-1) simplex if $\Theta_i \geq 0$, $\sum_{i=1}^{k} \Theta_i = 1$ and has the following probability mass function.

$$p(\Theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \Theta_1^{\alpha_i - 1} \ldots\ldots\ldots \Theta^{\alpha_k^{\alpha_{k-1}}} \tag{1}$$

Where α is a k vector, and are $\alpha_i' s$ are positive. $\Gamma(x)$ is a gamma distribution. This model with its accompanying properties facilitates the development of the parameter estimation for the LDA algorithm.

Furthermore the joint distribution over a topic mixture Θ given $\alpha$and β, a list of N topics y and also N words a, the following is obtained:

$$p(\Theta, y, a|\alpha, \beta) = p(\Theta|\alpha) \prod_{i=1}^{N} p(y_n|\Theta) p(a_n|y_n, \beta) \tag{2}$$

If we integrate over Θ and sum over topic y, the marginal distribution of a document is:

$$p(a|\alpha, \beta) = p(\Theta|\alpha)(\prod_{n=1}^{N} \sum_{y_n} p(z_n|\Theta) p(a_n|y_n, \beta) d\Theta \tag{3}$$

Taking the product over M documents for the equation stated above we get the probability of the corpus as follows:

$$p(D|\alpha, \beta) = \prod_{d=1}^{M} p(\Theta_d|\alpha)(\prod_{n=1}^{N_d} \sum_{y_n} p(y_{d_n}|\Theta_d) p(w_{d_n}|y_{d_n}, \beta)) d\Theta_d \tag{4}$$

This forms the basis of LDA topic modeling.

3.3 The Gibbs Sampler

Murphy [7] states that the Gibbs Sampler is an algorithm based on the Markov Chain Monte Carlo or MCMC. The MCMC is an important algorithm and the basic idea behind it, is to construct a Markov Chain on a state space whose distribution is the target density of interest.

With Gibbs sampling, each variable sampled is conditioned on the values of all the other variables in the distribution. This means that given a joint sample of $y^s$of all the variables, a new sample is generated y^{s+1} by sampling each component in turn, based on the most recent values of the other variables. For example if we had three variables, y_1,y_2 and y_3, then to demonstrate what goes on with Gibbs sampling we would have:

- $y_1^{s+1} \sim$ p($y_1|y_2^s, y_3^s$)
- $y_2^{s+1} \sim$ p($y_2|y_1^{s+1}, y_3^s$)
- $y_3^{s+1} \sim$p($y_3|y_1^{s+1}, y_2^{s+1}$)

This does generalize for D variables. If x_i is known we do not need to sample it.

The expression p($y_i|y_{i+1}$) is called the full conditional for variable i where y_i depends on other variables in general.

4 Data

The three general sentiments that will be considered for this experiment will go into three categories. These categories are positive sentiments, negative sentiments and neutral sentiments. Table 1will illustrate which group of sentiments the emoticons considered will go.

POSITIVE	NEGATIVE	NEUTRAL
:-)	: (	:-/
;)	: - (	:/
:-D	: '(	:-o
: D	: '-(	:o
;)		: - |

Table 1: Emoticons Table

4.1 Weather Data Set

The weather data set contains tweets with regards to weather. The data set is labeled, so it can be used it to test the correlation between labeled documents and documents associated with positive topics. The data set was obtained from crowd flower and can be accessed via this link https://www.crowdflower.com/data/weather-sentiment/. It contains 19611 tweets.

Table 2 contains statistics on the emoticons in the weather data set, the weighted percentages is weighted on the total number of emoticons considered for this experiment.

Emoticon	Weighted Percentage	Unweighted Percentage
:-)	21.76%	16.4%
;)	4.29%	3.24%
:-D	0.89%	0.68%
: D	23.82%	17.96%
;)	4.29%	3.24%
: (	9.99%	7.53%
: - (	4.99%	3.77%
: '(	1.41%	1.06%
: '-(	0.64%	0.48%
:-/	2.75%	2.08%
:/	22.73%	17.14%
:-o	0.13%	0.09%
:o	2.11%	1.59%
: - |	0.19%	0.14%

Table 2: Emoticon Statistics

5 Application

5.1 Preprocessing

The preprocessing stage entails removing words and special characters that will corrupt the output. Examples of the things that will be removed are words that appear once, or words like "the", "and" during preprocessing. Special characters are also taken out during the preprocssing stage, because of this the emoticons that will be considered for this analysis will be assigned special codes that will be illustrated in table 3. The benefit of doing it this way, is that one can easily keep track of the emoticons even after the preprocessing. The gensim library in python is used to do the topic modeling. Table 3 also shows the percentages of positive, negative and neutral sentiment in the weather data set.

	POSITIVE	NEGATIVE	NEUTRAL
	Hapnfa	Sadnfa	Sernfa
	:-)	: (	:-/
	;)	: - (	:/
	:-D	: '(	:-o
	: D	: '-(	:o
	;)		: - |
Weighted Percentage	55.06%	17.03%	27.91%
Unweighted Precentage	41.53%	14.92%	18.98%

Table 3: Preprocessing

5.2 Experimental Design

Each topic model will have ten slots and these slots will be made up of emoticons and words. The reason for having the ten slots is because in my python code I set up a loop that will ensure each topic model has ten slots. This is so that inference of what the thematic information has is simplified. The expected output in this stage will be in a specific topic model. There will be at least one emoticon conveying the statements and a bunch of words that have a latent meaning. So once the meaning of the latent collection of words has been deduced, coupled with the emoticons, we will be able to know the sentiments attached to the topic for a specific topic model. As stated in the background theory, the LDA method will be used to do the topic modeling. Table 4 will serve as an example of how the output for the official experiments will look.

Hapnfa	Sadnfa	Sernfa
Summer	Windy	Bloomberg
Beach	Gust	Financials
Bikini	Chill	Markets
Party	Drizzles	Currencies
Hapnfa	cold	Forex
Surfing	**Sadnfa**	Trading
Sand	Stuck	Opening
Holiday	Home	**Sernfa**
Drinks	Winter	Closing

Table 4: Example of output

From the first topic in the table above, it can be deduced that its summer holiday time and there are a lot of fun activities going on at the beach. The general sentiments with regards to it are positive, with the happy emoticons being expressed a lot. In reference to table 3, the "Hapnfa" coverts to happy emoticons.

From the second topic in the table above it can be deduced that it is a day with winter rains and the general sentiments with regards to that are negative, In reference to table 3, the "Sadnfa" coverts to sad emoticons.

From the third topic in the table above it can be inferred that its talk about financial markets provided by Bloomberg, the general sentiments associated with this are neutral.

The general idea for the actual application with regards to this is that ten topics will be obtained. Some of the topic will possible have more than one category of sentiments in it.

6 Results

6.1 Weather Data Set

This serves as the official application of topic modeling on the weather data set.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
Condtions	Day	Weather	Friday	Humidity	Year	Time	Week
Lake	Rainy	National	Raining	Mph	Went	Weather	Weather
High	**Hapnfa**	Service	Check	Wind	Weather	Supposed	Wtf
Tho	Early	Tornado	School	Weather	Tired	Tuesday	**Sadnfa**
Low	Humid	Shelter	Joplin	Feels	Fall	Seattle	Course
Hapnfa	Today	Indiana	**Hapnfa**	Noh	Girl	Family	Games
Current	Times	**Sadnfa**	Weather	East	**Sernfa**	**Hapnfa**	Appreciate
Guys	Pleasant	Delayed	Play	Southeast	Missing	Cancelled	Eating
Forecast	Breakfast	Loves	Break	Southwest	Flooding	Today	Showing
Thinking	Flip	Children	Okay	Bipolar	Trip	Talking	Canceled

Topic 9	Topic 10
Weather	Better
Central	Weather
Dallas	End
Bbq	World
Type	Red
Austin	Miles
Living	Expected
Coverage	Keeps
Storm	Phone
Worst	Wake

Table 5: Actual output for weather data set

As can be seen from table 5, there are some topic models without any emoticons in them and some do have. Each of the topics with emoticons in them will have their meaning deciphered and the topics with emoticons in them will be classified into one of the three sentiment group classifications. This will show us whether or not there is a correlation between the topics and the sentiment expressed. If there is a correlation then topic modeling on short text with emoticons can be used as a quick and easy sentiment analysis tool.

Topic one refers to a weather forecast predicting favorable conditions out on a lake. This could be good information for fisherman, hence the positive emotion expressed in the model via the positive emoticon. Topic one can be classified in the positive sentiment group.

Topic two refers to a day with early morning rains and humidity later on, and this has a positive signal because sleeping during rain or waking up to it is generally a nice feeling. Topic two will be classified into the positive sentiment group.

Topic three refers to a tornado in Indiana and thus it has a negative emotional signal, possibly due to damages in property and the danger posed to peoples lives. Topic three falls into the negative sentiment group.

From topic four it could be inferred that its a rainy Friday in Joplin, a city in the USA. This has been met with a positive reaction and thus topic four will be grouped into positive sentiments group.

Topic six refers to flooding and possible missing persons. This has the structure of a news report because the sentiment classification is neutral thus it will be placed in the neutral sentiments group.

Topic seven refers to an event being canceled on account of weather in Seattle and this leaves room for family bonding. This topic is classified into the positive sentiment group.

Topic eight also refers to the cancellation of an event, possibly a gaming convention and this leads to topic eight be classified in the negative sentiments group.

7 Conclusion

The research problem is to see whether there is a correlation between the sentiments expressed in a topic via emtoicons and the actual topic. In the application section of this paper it could be seen there was a correlation. An example would be topic three where a tornado was on rampage, it was met with negative sentiments. Weather reports predicting favorable weather conditions are met with positive sentiments and this does make sense, as everyone typically likes good weather.

All in all, topic modeling with emoticons can be used as a quick and easy form of sentiment analysis. Possible shortfalls of this research is that firstly the data set should be rich in emoticons, if it isn't then trying to conduct sentiment analysis this way is pointless. In the application section it could be seen that some topics were not classified into one of the three sentiment classification groups. This could stem from the fact that the weather data set is not rich in emoticons. Secondly the topic modeling was deployed on a labeled data set which did simplify things, a future recommendation would be to do the same on an unlabeled data set. Lastly due to the nature of topic modeling, inferring the meaning of the topic model could vary from user to user.

References

[1] Francesco Barbieri and Horacio Saggion. Modelling irony in twitter: Feature analysis and evaluation. In *The International Conference on Language Resources and Evaluation*, pages 4258–4264, 2014.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] Alexander Hogenboom, Daniella Bal, Flavius Frasincar, Malissa Bal, Franciska de Jong, and Uzay Kaymak. Exploiting emoticons in sentiment analysis. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 703–710. ACM, 2013.

[4] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Unsupervised sentiment analysis with emotional signals. In *Proceedings of the 22nd international conference on World Wide Web*, pages 607–618. International World Wide Web Conferences Steering Committee, 2013.

[5] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th Association for Computing Machinery Conference on Information and Knowledge Management*, pages 375–384. ACM, 2009.

[6] Jocelyn Mazarura, Alta de Waal, Frans Kanfer, and Sollie Millard. Topic modelling for short text. *Proceedings of the Pattern Recognisation Association of South Africa (PRASA 2014)*, 2014.

[7] Kevin P.Murphy. *Machine Learning: A Probablistic Perspective*. The MIT Press, 2002.

[8] Daniel Ramage, Susan T Dumais, and Daniel J Liebling. Characterizing microblogs with topic models. *The International Conference on Weblogs and Social Media*, 10:1–1, 2010.

Appendix

```
# -*- coding: utf-8 -*-
"""
Created on Wed Aug 24 11:56:48 2016

@author: Mwila Chikonde
"""

from gensim.corpora import Dictionary
from gensim.models import ldamodel
from gensim.parsing.preprocessing import STOPWORDS
from gensim import utils
import numpy
import re

def wordlist(text,stopwords=STOPWORDS):
    #remove stopwords and words <= 2 letters
    return [w
            for w in utils.tokenize(text, lower=True)
            if w not in stopwords and len(w) > 2]

#path to text file:
f_path = "modweather.txt"
#open text file:
fopen = open(f_path)
#read text file
tweet_corpus = fopen.readlines()
texts = [[w for w in wordlist(tweet)] for tweet in tweet_corpus]

dictionary = Dictionary(texts)
dictionary.filter_extremes(no_below=5, no_above=0.85)
corpus = [dictionary.doc2bow(text) for text in texts]

numpy.random.seed(1) # setting random seed to get the same results each time.
model = ldamodel.LdaModel(corpus, id2word=dictionary, num_topics=100)

for i in model.show_topics():
    print i[1]
    print '\n'

#model.get_term_topics('water')
```

Gaussian local level model and the use of a Kalman filter

Natasha Chirwa 13159136

WST795 Research Report

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

Supervisor: Dr J. Kleyn

Department of Statistics, University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

2 November 2016

Abstract

In this paper, the Gaussian local level model will be considered. The model is the simplest form of the state space model. We will also examine the theory of the Kalman filter and illustrate the application of the Kalman filter to obtain the smoothing and filtering distributions. In a simulation study, the Kalman filter was applied to a data set that was randomly generated from a normal distribution and it was found that the filter calculated the mean and variance of the unobserved state given the observations obtained from the data set. From the simulation study, the signal to noise ratio had an impact on the results obtained by applying the Kalman filter. For a signal to noise ratio of less than 1, the signal extraction of the process has some random variation, meaning that the noise in the process is large relative to the signal. On the other hand, a signal to noise ratio of more than 1 shows that the signal extraction of the process is easier and more reliable. In other words, the process has a larger signal relative to the noise. The case when the signal to noise ratio is 0 is known as a special case. The estimation results show that when the signal to noise ratio is 0, the process has approached a steady state. The technique was also applied to the Nile data set to analyse the effect that the Ashwan high dam has had on the Nile River from 1871 to 1970. It was found that the Ashwan high dam has caused various level shifts in the time series depicted by the Nile data, meaning that the construction of the Ashwan high dam has played a major role on the flow of the Nile River.

Declaration

I, *Natasha Chirwa,* 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.

_____
Natasha Chirwa

_____
Judy Kleyn

_____
02 November 2016

Acknowledgments

First and foremost, I would like to thank God for his favour, guidance and unconditional love as I journeyed through my honours year. He gave me direction and strength. I'll forever be grateful to God for making all this possible.

Secondly, I would like to thank my supervisor, Dr Judy Kleyn, for cheering me on even when I felt like giving up. Her encouragement kept me going when I felt like the work load was just beyond me.

Thirdly, I am also thankful to my parents, family and friends. Thank you for letting me act sane when things around seemed insane. Thank you for being my pillars.

Finally, I would like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a post graduate bursary.

Contents

1 Introduction and Literature Review **6**

2 Model **7**
2.1 Gaussian local level model 7
2.2 Kalman filter 8
2.2.1 Filtering 8
2.2.2 Smoothing 9

3 Application **10**
3.1 Simulated local level model 10
3.2 Nile River 16

4 Conclusion **17**

Appendix **21**

List of Figures

1 Simulated data and output: (i) simulated data, signal, filtered state and its 95% confidence intervals; (ii) simulated data, signal, smoothed state and its 95% confidence interval; (iii) Filtered and smoothed mean; (iv) Filtered and smoothed standard error. 11
2 Simulated data and output: (i) simulated data, signal, filtered state and its 95% confidence intervals; (ii) simulated data, signal, smoothed state and its 95% confidence interval; (iii) Filtered and smoothed mean; (iv) Filtered and smoothed standard error. 12
3 Simulated data and output: (i) simulated data, signal, filtered state and its 95% confidence intervals; (ii) simulated data, signal, smoothed state and its 95% confidence interval; (iii) Filtered and smoothed mean; (iv) Filtered and smoothed standard error. 13
4 Simulated data and output: (i) simulated data, signal, filtered state and its 95% confidence intervals; (ii) simulated data, signal, smoothed state and its 95% confidence interval; (iii) Filtered and smoothed mean; (iv) Filtered and smoothed standard error. 14
5 Simulated data and output: (i) simulated data, signal, filtered state and its 95% confidence intervals; (ii) simulated data, signal, smoothed state and its 95% confidence interval; (iii) Filtered and smoothed mean; (iv) Filtered and smoothed standard error. 15
6 Nile data and output of Kalman filter: (i) simulated data, signal, filtered state and its 95% confidence intervals; (ii) simulated data, signal, smoothed state and its 95% confidence interval. 16
7 Nile data and output of Kalman filter: (i) Filtered and smoothed mean; (ii) Filtered and smoothed standard error. 17

1 Introduction and Literature Review

The phrase "state space" originated from the article first published in 1960 by Rudolph Kalman [16]. Koller and Friedman [19] defined the state space model as a class of probabilistic graphical models. The state space model provides a general framework for analyzing deterministic and stochastic models. This model allows us to make a link between the observed variables and the state variables in order for us to make statistical inference about the unobserved states.

The state space model is used widely in various technical and quantitative fields such as economics, finance, engineering and genetics. Considering its vast usage, different fields have different names all referring to the state space model. In engineering, Roumeliotis and Bekey [25] used the state space model in robotics. They defined a robot as a device that carries sensors that are able to monitor its motion and also allows it to estimate its path as it moves away from its current location. The state space model was used specifically to track the pose displacements of the robot from one place to another, thus the robot was able to navigate from place to place given that its current position and orientation at any given time was known. Strang and Borre [29] applied the state space model in navigation. The model was used in positioning problems for a Global Positioning System (GPS), which watches the movement of the earth's crust. Furthermore, the state space model has also been used in areas of tracking [21] and computer vision [24]. From an engineering standpoint, the state space model is generally referred to as a dynamic linear model.

In economics and finance, Wu and Zeng [32] made use of the state space model to model financial data for prediction of interest rates, Stock and Watson [28] used these models to measure the sensitivity of the business cycles and Hamilton [3] used the model in the estimation of future expected inflation. The state space model is also referred to as the latent process model [20] in some applications or as a hidden Markov model [32, 22] where as some fundamental statistical treatments of its classes are discussed in Cappé, Mouline and Rydén [4].

The simplest form of the state space model is called the local level model. This paper will focus on the Gaussian local level model by assuming normality and thus making computations of the recursive equations easier [8].

In 1960, Rudolph Kalman [16] published his now widely used article on the Kalman filter. His article described a recursive algorithm that can be used to find the solution to a linear filtering problem based on a discrete data set.

The Kalman filter is a set of mathematical equations that provide a recursive algorithm to enable one to estimate the state and the error covariance of a given process. In terms of estimation, it is a very powerful tool because it takes into account the past, present and future of the states in the process.The main aim of the filter is to minimize the mean square error of the estimated parameters [27]

Lipton, Fujiyoshi and Patil [33] used the Kalman filter to explain human motion tracking. Due to the fact that the tracking of the human body is a problem, the Kalman filter was used in estimation of the parameters of the human body where as moving targets were extracted from a real time video stream. The targets were then classified into categories based on image-based properties and thus made tracking possible [33]. Human motion tracking was further illustrated by Welch [31]. From recent research by The Council for Scientific and Industrial Research (CSIR) [1], human language technology makes it easier for humans to communicate with machines. In order for the machine to recognize a human's voice with precision, the Kalman filter has been used to filter out noise signals. Other applications of the Kalman filter in voice recognition in devices have been presented in [15]. The Kalman filter has also played a vital role in video stabilization [2] and in econometrics and finance [12].

In section 2, a discussion on the Gaussian local level model and the Kalman will be presented with all necessary equations. By making use of the Kalman filter and its recursive equations, filtering and smooth-

ing distributions will be obtained. In section 3, two examples (Simulation study and Nile data set) will be included on how to use a Kalman filter for linear filtering and prediction in order to obtain the smoothing and filtering distributions using R as the programming language. Particular attention will be given on the graphs obtained for the original distribution, filtered distribution and the smoothed distribution. Lastly, in section 4, comments on the findings of the examples will be provided. A brief discussion on the shortfalls and recommendations of the research report will also be included.

2 Model

2.1 Gaussian local level model

In this section, the Gaussian local level model will be discussed by looking at the three parts that make up the model, namely: the measurement equation, the transition equation and the initial state [8].

The first part is the observation equation which is also referred to as the measurement equation and it is given by:

$$Y_t \quad = \quad \mu_t + \varepsilon_t \, , \, \varepsilon_t \overset{iid}{\sim} N(0, \sigma_\varepsilon^2) \, , \, t = 1, 2, 3....T \tag{1}$$

The second part is the state equation which is also referred to as the transition equation and it is given by:

$$\mu_{t+1} \quad = \quad \mu_t + \eta_t \, , \, \eta_t \overset{iid}{\sim} N(0, \sigma_\eta^2) \, , \, t = 1, 2, 3.... \tag{2}$$

Lastly, the third part is the initial state given by:

$$\mu_1 \mid \mathcal{F}_1 \sim N(a_1, P_1) \tag{3}$$

From (1), the observation y_t consists of a state μ_t which measures the stochastic trend and the noise/error term ε_t. The state μ_t is a slowly varying component also known as a random walk. Since a random walk is non-stationary, the Gaussian local level model is thus non-stationary meaning that the distributions of the variables Y_t and μ_t depend on time t . Since Y_t is a messed up version of the true state μ_t of the process, the state is allowed to move over time. In addition, the assumption that ε_t is independent and identically normally distributed with mean 0 and variance σ_ε^2 has been made. By using equation (1), the main aim is to extract the true state μ_t from the observed measurement y_t. [8]

From (2), the unobserved state μ_{t+1} consists of the state μ_t which measures the stochastic trend such that μ_t is a random walk and the noise/error term η_t. The state is a Markovian process since it only depends on the previous period and the transition equation is a random walk process. It is assumed that η_t is independent and identically normally distributed with mean 0 and variance σ_η^2. [8]

From (3), the initial state μ_1 given the observations (y_1, y_0) is normally distributed with $E(\mu_1 \mid \mathcal{F}_1) = a_1$ and $var(\mu_1 \mid \mathcal{F}_1) = P_1$. This is prior information regarding the state of the process. [8]

The signal to noise ratio is defined as: the measure of the strength of the signal relative to the strength of the noise. Taking into account the fact that the behaviour of the process Y_t is influenced greatly by the signal to noise ratio, signal extraction will be considered by making use of the signal to noise ratio (SNR) which is defined as:

$$q \quad = \quad \frac{\sigma_\eta^2}{\sigma_\varepsilon^2} \tag{4}$$

The signal to noise ratio is used to determine whether the process has a signal (not just a random variation) or just noise (a random variation). From (4), the $SNR = q = \frac{\sigma_\eta^2}{\sigma_\varepsilon^2} = 0$ if and only if $\sigma_\eta^2 = 0$, this means that the unobserved state μ_{t+1} is equal to the state μ_t thus the process has a steady state. On the other hand, a larger signal to noise ratio ($\sigma_\eta^2 > \sigma_\varepsilon^2$) , means that the strength of the signal is large relative to the strength of the noise thus the signal is significant. A large signal to noise ratio also means that extraction of information for the process Y_t is easier and the results thereof are more reliable. [8, 27, 14]

2.2 Kalman filter

In this section, the Kalman filter will be considered by firstly focusing on the distributional properties of different components of the process and then to specifically focus on the smoothing and filtering distributions.

The Kalman filter provides us with a recursive algorithm such that it calculates the mean and variance of the unobserved state μ_t of Y_{t+1}, given a set of observations. The recursive algorithm makes computations easier in such a way that as soon as a new observation y_t becomes available, the process can be updated and the new best estimate can be recalculated. [8]

The local level model considers the observation variable Y_t over time where the latent state μ_t is unobserved. Since the Gaussian local level model is under consideration, this means that all distributions are normal hence the conditional joint distributions of observations will again be normal. In essence, the conditional joint distribution of a set of observations given another set of observations will be normally distributed. [27, 8] [See Appendix]

2.2.1 Filtering

Filtering is done to extract the state μ_t given a set of observations. Observations from time period 1 are used up to a specified time period t such that the observations $\mathcal{F}_t = \{y_1, y_2, ..., y_t\}$ are used to estimate the state μ_t. Filtering is done to remove measurement errors from the given data.

Using the initial condition $\mu_1 \mid \mathcal{F}_1 \sim N(a_1, P_1)$ and the fact that the error terms ϵ_t and η_t are Gaussian, we have that the model $Y_1, Y_2..., Y_T, \mu_1, ...\mu_T$ is a big multivariate normal due to its linear structure. Thus the conditional distribution of $\mu_1, ...\mu_T \mid Y_1, Y_2, ...Y_T$ is normal hence we can assume that the model will have marginal densities that are also normal.

Assume $\mu_t \mid Y_1, Y_2, ...Y_{t-1} \sim N(a_t, P_t)$ where a_t and P_t (State variance) are known. Also assume that the conditional distribution of $\mu_{t+1} \mid Y_1, Y_2, ...Y_t \sim N(a_{t+1}, P_{t+1})$. In addition, the conditional distribution of $\mu_t \mid Y_1, Y_2, ...Y_t \sim N(a_{t|t}, P_{t|t})$, which is known as the filtering distribution. As soon as a new observation y_t becomes available, $a_{t|t}$, $P_{t|t}$, a_{t+1} *and* P_{t+1} need to be recalculated in order to make inferences about the parameters of the new best estimate. The filtered estimate of the state μ_t is defined as $a_{t|t}$ such that $E(\mu_t \mid Y_1, Y_2, ...Y_t) = a_{t|t}$ with a corresponding variance of $P_{t|t}$. On the other hand, a_{t+1} is called the one step ahead predictor of the unobserved state μ_{t+1} such that $E(\mu_{t+1} \mid Y_1, Y_2, ...Y_t) = a_{t+1}$ with a corresponding variance of P_{t+1}.

Starting with the initial condition $\mu_1 \mid \mathcal{F}_1 \sim N(a_1, P_1)$, a_{t+1}, P_{t+1} and F_t can be calculated by making use of recursive equations.

v_t, the one step ahead prediction error of y_t is defined as $v_t = y_t - a_t$ for $t = 1, 2....T$ such that $E(v_t) = E(E(v_t \mid \mathcal{F}_{t-1})) = 0$ since $E(v_t \mid \mathcal{F}_{t-1}) = 0$ and $var(v_t) = var(y_t - a_t) = var(y_t) + var(a_t) = P_t + \sigma_\varepsilon^2 = F_t$ thus $v_t \sim N(0, F_t)$.
Recursive equations for a_{t+1}, P_{t+1} and F_t are defined as follows:

$$a_{t+1} = a_t + K_t v_t \; where \; K_t = \frac{P_t}{F_t} = \frac{P_t}{P_t + \sigma_\varepsilon^2} \tag{5}$$

$$P_{t+1} = P_t(1 - K_t) + \sigma_\eta^2 \tag{6}$$

$$F_t = P_t + \sigma_\varepsilon^2 \tag{7}$$

where F_t is referred to as the variance of the prediction error v_t and K_t is known as the Kalman gain such that $K_t \in [0, 1]$.

As soon as a new observation y_t is available, parameters of the new state μ_{t+1} can be estimated using the prior information a_{t+1}, P_{t+1}, F_t and an updated v_t.

$$a_{t|t} = E(\mu_t \mid \mathcal{F}_t) = a_t + K_t v_t = a_t + (\frac{P_t}{F_t})v_t = a_t + (\frac{P_t}{P_t + \sigma_\varepsilon^2})(y_t - a_t) = a_{t+1} \tag{8}$$

$$P_{t|t} = var(\mu_t \mid \mathcal{F}_t) = var(\mu_t \mid \mathcal{F}_{t-1}, v_t) = \frac{P_t \sigma_\varepsilon^2}{P_t + \sigma_\varepsilon^2} = P_{t+1} - \sigma_\eta^2 \tag{9}$$

thus the distribution of μ_{t+1} becomes known, in other words $\mu_{t+1} \sim N(a_{t|t}, P_{t|t})$.

The filtering distribution can thus be defined as : $\mu_t \mid Y_1, Y_2, ...Y_t \sim N(a_{t|t}, P_{t|t})$. [27, 8]

The process starts with the observation y_1 and an initial condition of (a_1, P_1) , it is then possible to compute (a_2, P_2, F_2). When a new observation y_2 becomes available, v_2 is updated using the given equations above. The new value v_2 is used to then update the conditional mean $a_{2|2}$ and the conditional variance $P_{2|2}$ thus the distribution of the new state with its estimated parameters is obtained. This process is repeated until time period t. [27, 8]

If P_t, the state variance, converges to a positive value, this means that the process has reached a steady state hence $P_{t+1} = P_t$. As soon as the process converges, the computation of F_t and K_t can be stopped because $a_{t+1} = a_t + K_t v_t$ from (1). [8]

As the process approaches a steady state ($t \to \infty$), the Kalman gain $K_t = \frac{P_t}{F_t} \to K = \frac{\frac{\sqrt{q^2+4q}-q}{2}+q}{\frac{\sqrt{q^2+4q}-q}{2}+q+1} \in [0, 1]$.
So the update $a_{t|t} = a_t + K_t v_t \cong a_t + K v_t$ whereby as t increases, the approximation becomes more precise and accurate. This is known as the EWMA (Exponentially weighted moving average) forecast meaning that filtering converges to the EWMA forecast where K is defined in terms of the signal to noise ratio (SNR). In other words, the local level model is a model representation for EWMA forecasting.

2.2.2 Smoothing

Smoothing is done to extract the state μ_t given a set of observations whereby observations from time period 1 to the end of the process at time period T are used ($\mathcal{F}_T = \{y_1, y_2, ...y_t..., y_T\}$). The mean and variance is calculated conditional on the set of observations$\mathcal{F}_T$.

Once again, using the initial condition $\mu_1 \mid \mathcal{F}_1 \sim N(a_1, P_1)$ and the fact that the error terms ϵ_t and η_t are Gaussian, we have that the model $Y_1, Y_2..., Y_T, \mu_1, ...\mu_T$ is a big multivariate normal due to its linear structure. Thus the conditional distribution of $\mu_1, ...\mu_T \mid Y_1, Y_2, ...Y_T$ is normal hence we can assume that the model will have marginal densities that are also normal.

The smoothing distribution can be defined as: $\mu_t \mid Y_1, Y_2, ...Y_T \sim N(a_{t|T}, P_{t|T})$ for $t = 1, 2, ..., T$ where $a_{t|T}$ is called the smoothed state and $P_{t|T}$ is called the smoothed state variance. [23]

By using joint distribution properties and the Bayes theorem, it can be shown that $\mu_t \mid \mathcal{F}_t \sim N(a_{t|t}, P_{t|t})$ and $\mu_{t+1} \mid \mu_t, F_t \sim N(\mu_t, \sigma_\eta^2)$.
Hence the backward result can be defined as follows:

$$\mu_t \mid \mu_{t+1}, \mathcal{F}_T \sim N\left(a_{t|t} + \frac{P_{t|t}}{P_{t+1|t}}\left(\mu_{t+1} - a_{t|t}\right),\ P_{t|t} - \frac{P_{t|t}^2}{P_{t+1|t}}\right) \tag{10}$$

Since $\mu_{t+1} \mid \mathcal{F}_T \sim N\left(a_{t+1|T}, P_{t+1|T}\right)$ then $\mu_t \mid \mathcal{F}_T \sim N\left(a_{t|T}, P_{t|T}\right)$ where

$$a_{t|T} = E(\mu_t \mid \mathcal{F}_T) = a_{t|t} + \frac{P_{t|t}}{P_{t+1|t}}(a_{t+1|T} - a_{t|t}) = a_{t|t} + H_t(a_{t+1|T} - a_{t|t}),\ H_t = \frac{P_{t|t}}{P_{t+1|t}} \tag{11}$$

$$P_{t|T} = var(\mu_t \mid \mathcal{F}_T) = P_{t|t} - \frac{P_{t|t}^2}{P_{t+1|t}} + \left(\frac{P_{t|t}}{P_{t+1|t}}\right)^2 P_{t+1|T} = P_{t|t} + H_t(P_{t+1|T} - P_{t+1|t})H_t^{'} \tag{12}$$

If a simulation from $\mu_T \mid \mathcal{F}_T \sim N(a_{T|T}, P_{T|T})$ is done, then it is possible to simulate backwards through the result given by (10) to $t = T, T-1, T-2, ..., 1$ meaning that (10) can be run backwards in time. This process is known as the smoother in time series. It is also referred to as the fixed lag smoother or as the Kalman smoother. [23]

The simulation smoother is the path drawn from $\mu_1, ...\mu_T \mid \mathcal{F}_T$ whereby classical references based on the simulation smoother have been discussed in [6], [10] and [5].

3 Application

In this section, the Kalman filter will be illustrated to obtain the smoothing and filtering distributions by making use of two different applications namely: a simulated local level model and a set of observations from the Nile river.

3.1 Simulated local level model

The local level model was simulated in R (Appendix) as defined in (1), (2) and (3). The random normal deviates $\varepsilon_t^* \sim N(0, \sigma_\varepsilon^2)$ and $\eta_t^* \sim N(0, \sigma_\eta^2)$ for $t = 1, ..., T$ are drawn. Using the local level recursion, the observations are generated as follows [8] :

$$y_t^* = \mu_t^* + \varepsilon_t^* \ for\ t = 1, ..., T$$
$$\mu_{t+1}^* = \mu_t^* + \eta_t^* \ for\ t = 1, ..., T$$
$$for\ some\ initial\ values\ a_1^*\ and\ P_1^*$$

The application of filtering and smoothing on the simulated local level model is shown in the R code (Appendix) as defined in equations (5) - (9) for filtering and (11) - (12) for smoothing.

The simulated data will be analyzed using the local level model with $a_1 = 0$, $P_1 = 1,000,000$ and different pairs of $(\sigma_\eta^2, \sigma_\varepsilon^2)$. The different components of filtering and smoothing for $t = 1, 2....T$ given by the Kalman filter is shown graphically in Figure 1 - Figure 5 where the role of the signal to noise ratio is also illustrated.

Case 1: $\sigma_\varepsilon^2 > \sigma_\eta^2$ ($\sigma_\varepsilon^2 = 400,\ \sigma_\eta^2 = 25,\ q = 0.0625$)

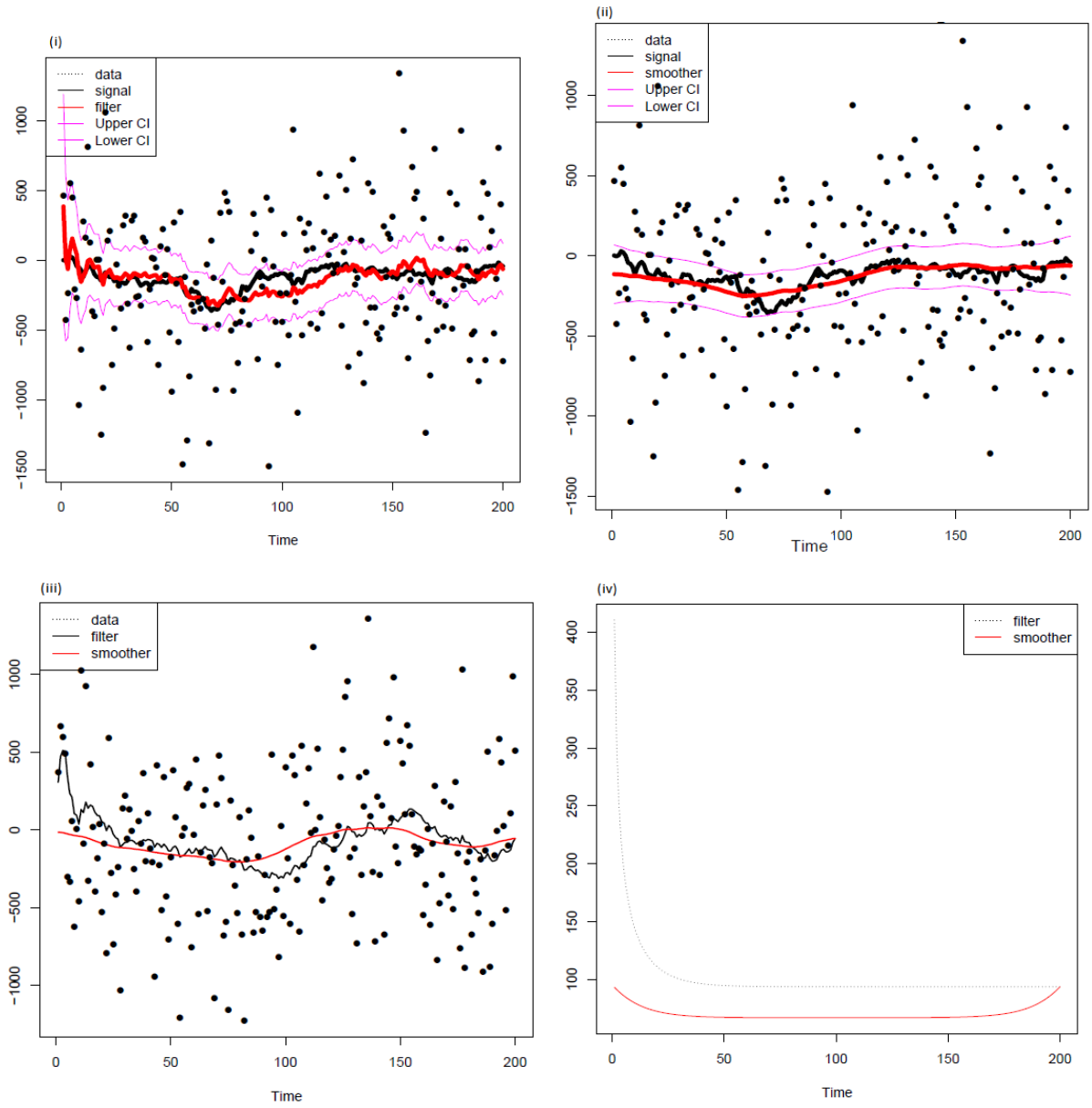


Figure 1: Simulated data and output: (i) simulated data, signal, filtered state and its 95% confidence intervals; (ii) simulated data, signal, smoothed state and its 95% confidence interval; (iii) Filtered and smoothed mean; (iv) Filtered and smoothed standard error.

From the panel of sketches above, the most obvious feature is that there is a large confidence interval due to the uncertainty associated with the noise being more significant than the signal in the process. Another feature observed is that the smoother has a more even surface than the filter. It is also visibly clear that the standard error for the smoother is less than the standard error of the filter for the entire process. However, at time period T, the standard error for the both the smoother and the filter converges to a constant value.

Case 2: $\sigma_\varepsilon^2 > \sigma_\eta^2$ $(\sigma_\varepsilon^2 = 1,\ \sigma_\eta^2 = 0.1,\ q = 0.1)$

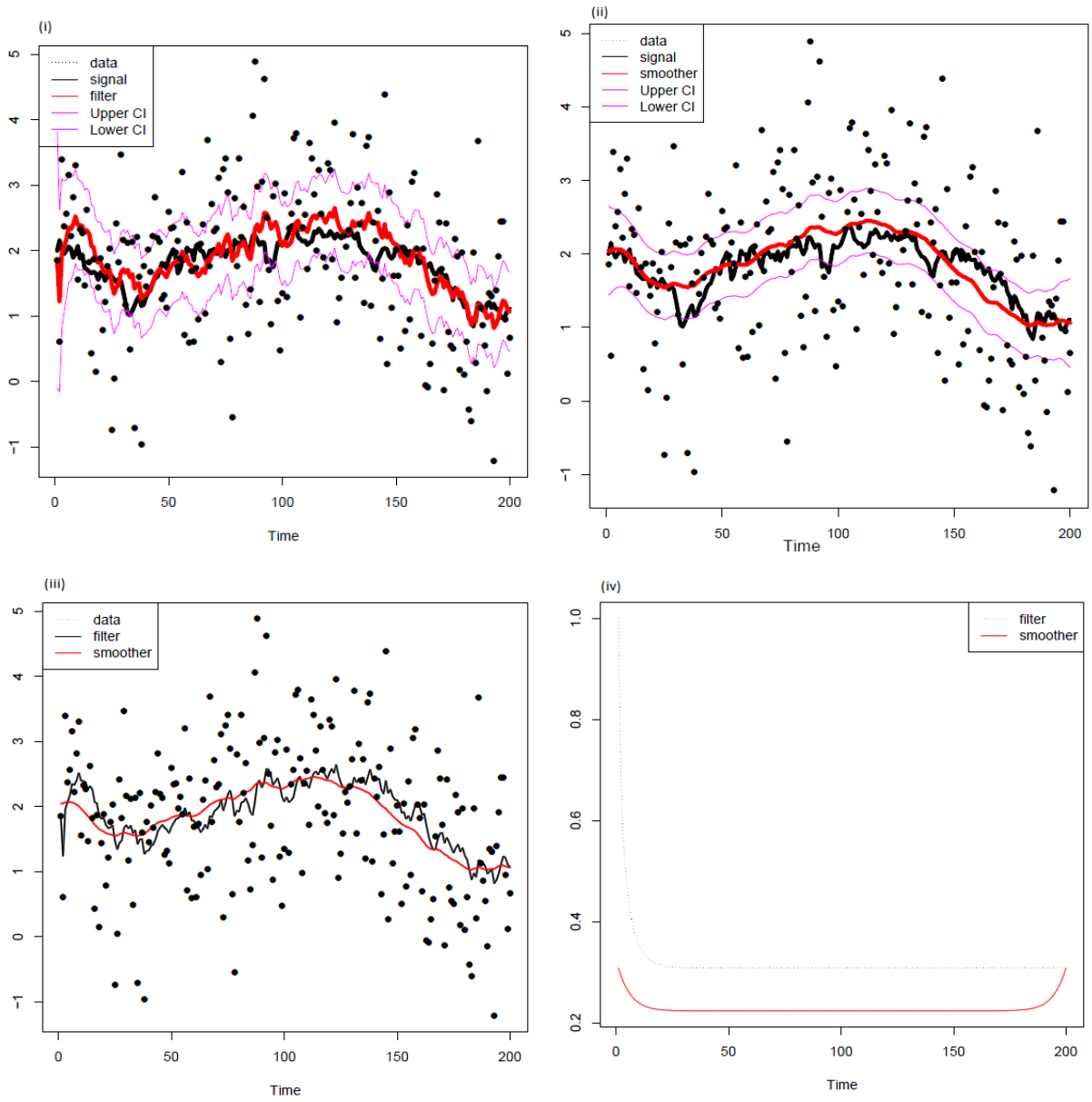


Figure 2: Simulated data and output: (i) simulated data, signal, filtered state and its 95% confidence intervals; (ii) simulated data, signal, smoothed state and its 95% confidence interval; (iii) Filtered and smoothed mean; (iv) Filtered and smoothed standard error.

Similarly to case 1, the panels of sketches above show that the smoother has a more even surface than the filter. Looking at the filter and smoother standard error, the smoother standard error is less than the filter standard error. The only difference between case 1 and case 2 is the interval on the vertical axis. Case 1 has a larger interval compared to case 2 due to the fact that the chosen values for σ_ε^2 and σ_η^2 in case 1 are bigger than those in case 2.

Case 3: $\sigma_\varepsilon^2 < \sigma_\eta^2$ ($\sigma_\varepsilon^2 = 25,\ \sigma_\eta^2 = 400,\ q = 16$)

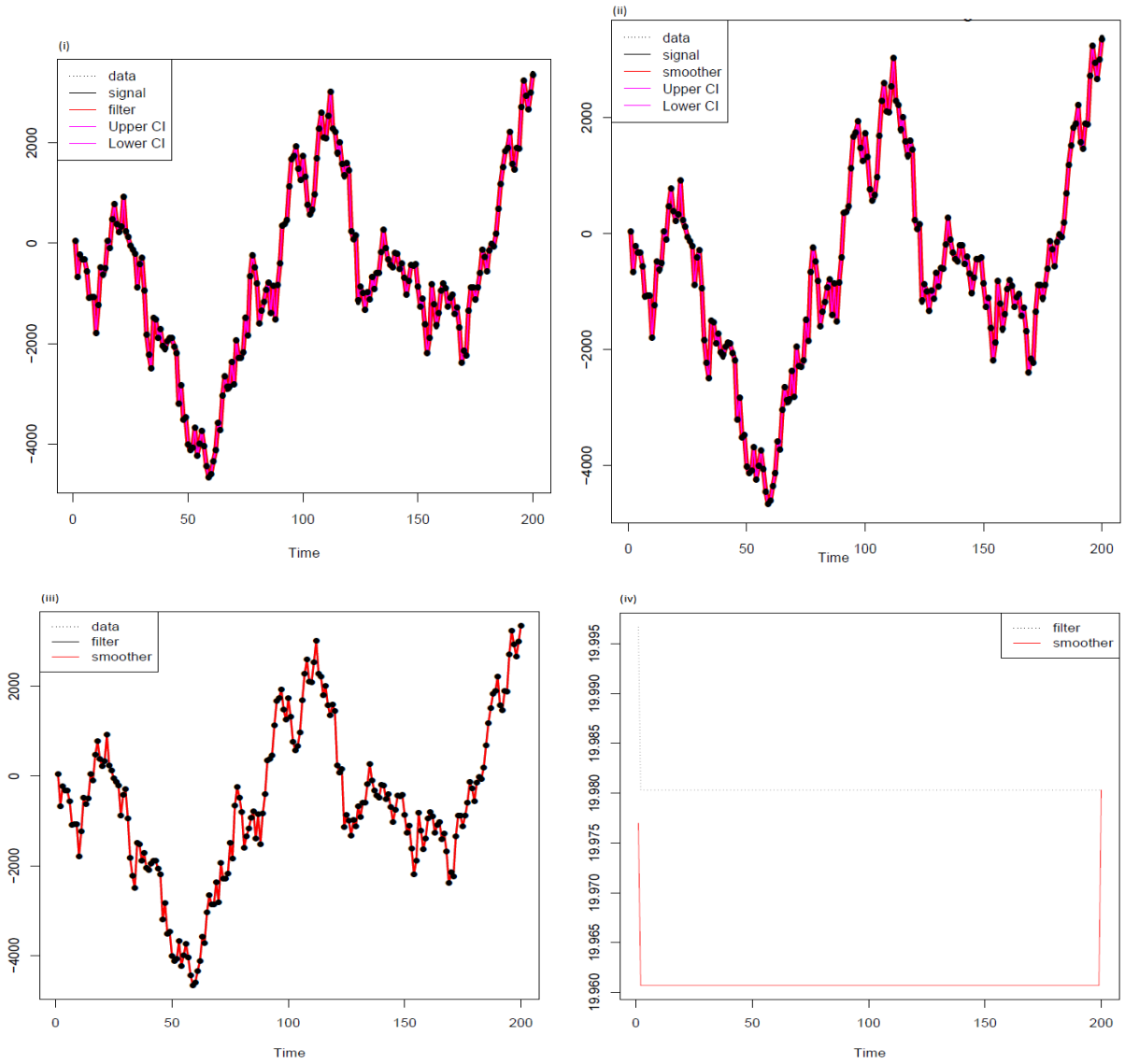


Figure 3: Simulated data and output: (i) simulated data, signal, filtered state and its 95% confidence intervals; (ii) simulated data, signal, smoothed state and its 95% confidence interval; (iii) Filtered and smoothed mean; (iv) Filtered and smoothed standard error.

From the panel of sketches above, it can be observed that the confidence interval is narrower due to the fact that the strength of the signal is large relative to the strength of the noise in the process. This means that the noise in the process has been filtered/smoothed out in such a way that there is greater confidence in the process that the state obtained is the unobserved state since the state is more significant. From graph (iii), there is no significant difference between the filter and smoother mean. Similarly to case 1, the standard error of the smoother is less than that of the filter although the gradient is higher. At time period T, the standard error converges to the same constant value.

Case 4: $\sigma_\varepsilon^2 < \sigma_\eta^2$ ($\sigma_\varepsilon^2 = 2,\ \sigma_\eta^2 = 3,\ q = 1.5$)

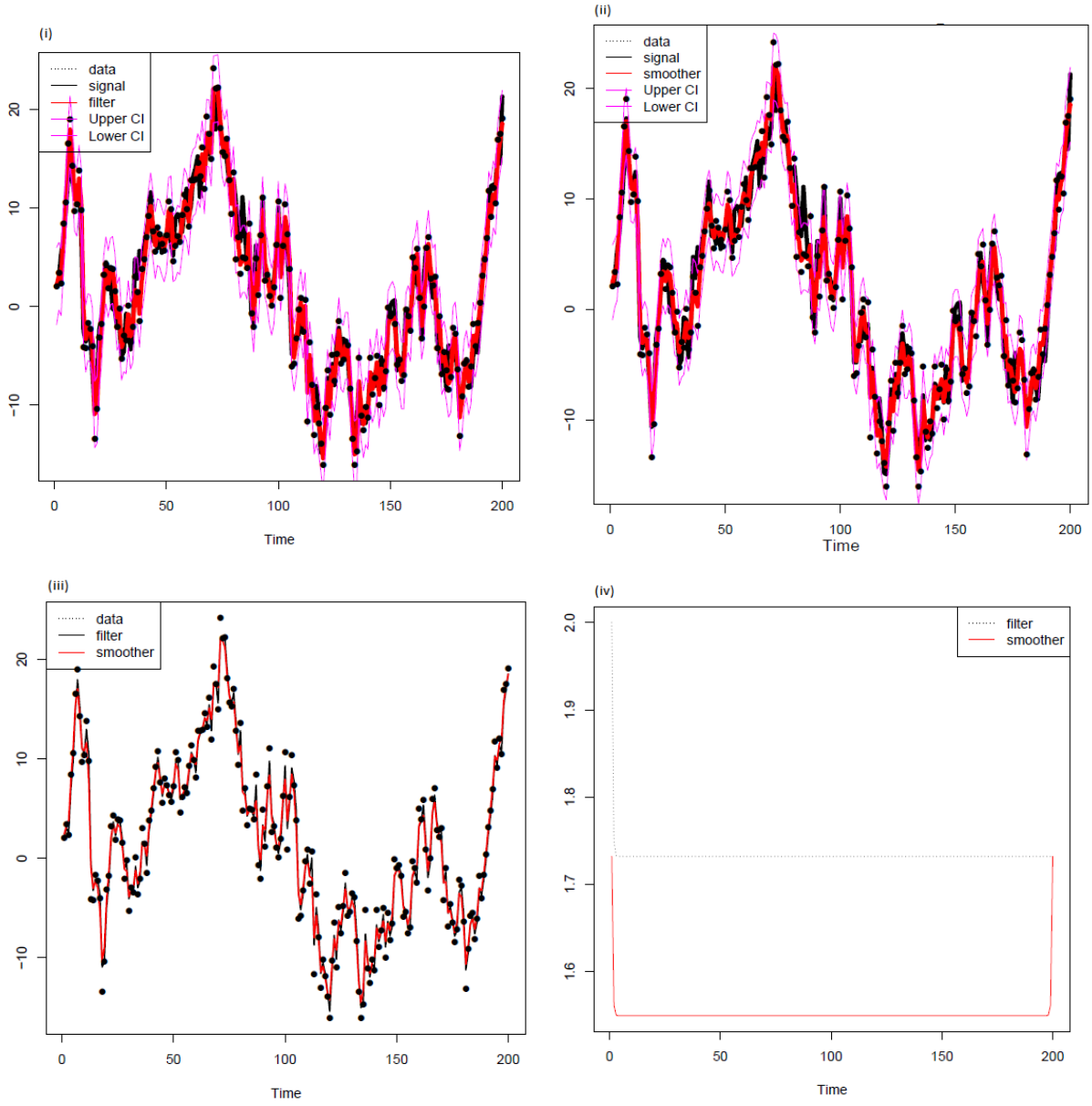


Figure 4: Simulated data and output: (i) simulated data, signal, filtered state and its 95% confidence intervals; (ii) simulated data, signal, smoothed state and its 95% confidence interval; (iii) Filtered and smoothed mean; (iv) Filtered and smoothed standard error.

Similarly to case 3, the signal in the process is large relative to that of the noise. It can also be observed that both the filter and smoother do not differ significantly from the signal hence there is almost a perfect fit. The interval in case 4 is smaller than case 2 because of the chosen values for σ_ε^2 and σ_η^2 for the different cases. Considering the fact there is not much of a difference between the filter/smoother and the signal, the extraction of information for the process Y_t is more reliable.

Case 5: $\sigma_\eta^2 = 0$ ($\sigma_\varepsilon^2 = 400,\ \sigma_\eta^2 = 0,\ q = 0$)

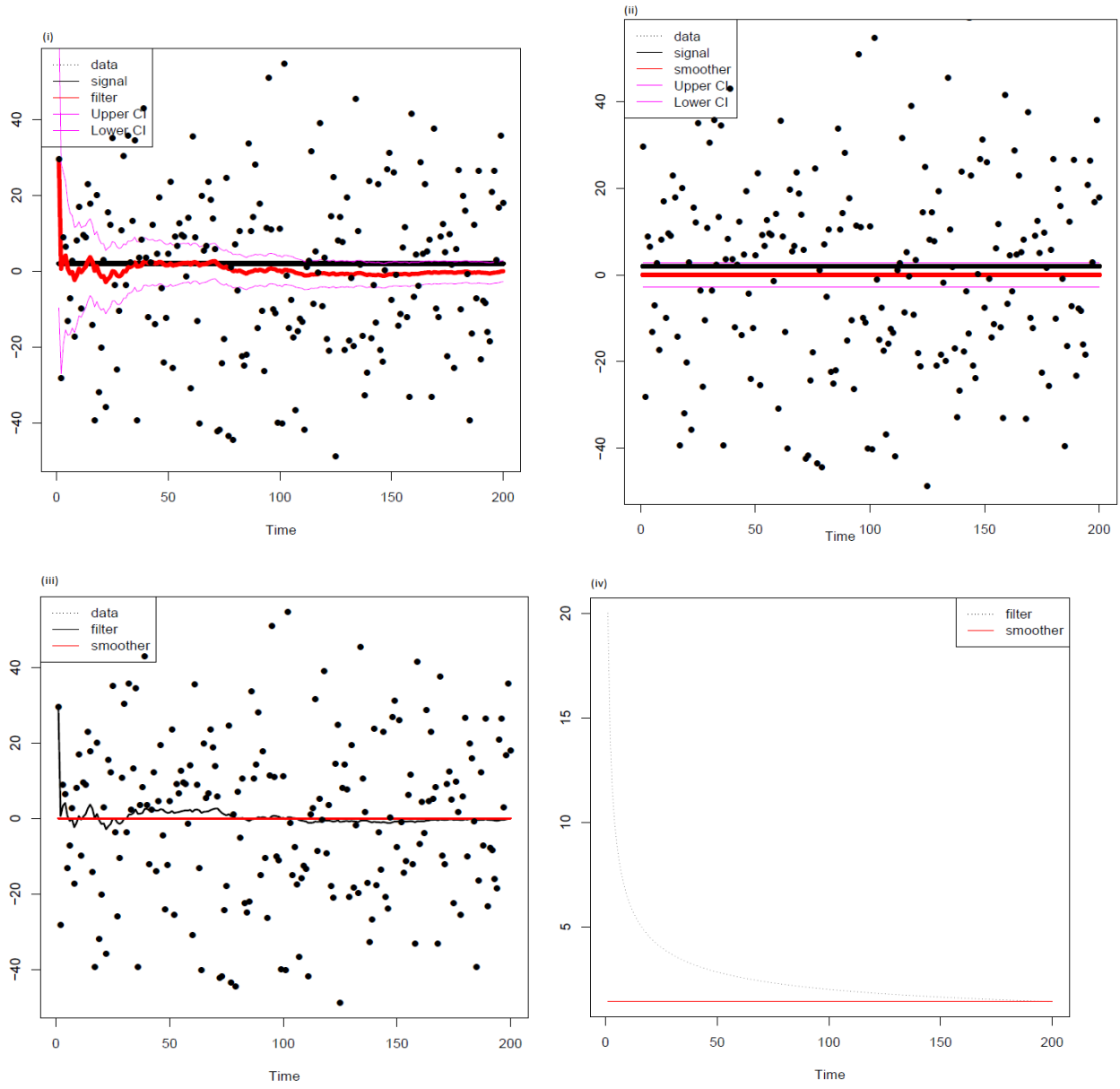


Figure 5: Simulated data and output: (i) simulated data, signal, filtered state and its 95% confidence intervals; (ii) simulated data, signal, smoothed state and its 95% confidence interval; (iii) Filtered and smoothed mean; (iv) Filtered and smoothed standard error.

From the panel of sketches above, it is very evident that when $\sigma_\eta^2 = 0$, the signal is also 0 and thus this implies that the state will stay the same for the entire process. Similarly to the four cases above, the smoother is has an even surface compared to the filter. Since $\sigma_\eta^2 = 0$ and using the local level model (2), it justifies the fact that the mean of the smoother is 0 and the standard error is approximately 0. The standard error for the smoother has a gradient of 0 and its value has not changed since the start of the process. It can also be observed that the standard error for the filter decreases exponentially and at time period T, the value of the standard error is the same as that of the smoother.

3.2 Nile River

The Nile data set consists of observations that are a series of readings of the annual flow volume at Aswan from time period 1871 to 1970. The local level model [(1), (2), (3)] will be illustrated by selecting arbitrary values for a_1 and P_1 and the values for σ_η^2 and σ_ε^2 will be chosen such that these values are the maximum likelihood estimates.

The data will be analyzed using the local level model with $a_1 = 0$, $P_1 = 10,000,000$, $\sigma_\eta^2 = 24.51$ and $\sigma_\varepsilon^2 = 129.92$. After running the R code (Appendix), the values of the filtered state, the smoothed state together with the other components such as P_t and the confidence intervals $for\ t = 1, ..., T$, given by the Kalman filter can shown graphically in Figure 6 and 7 .

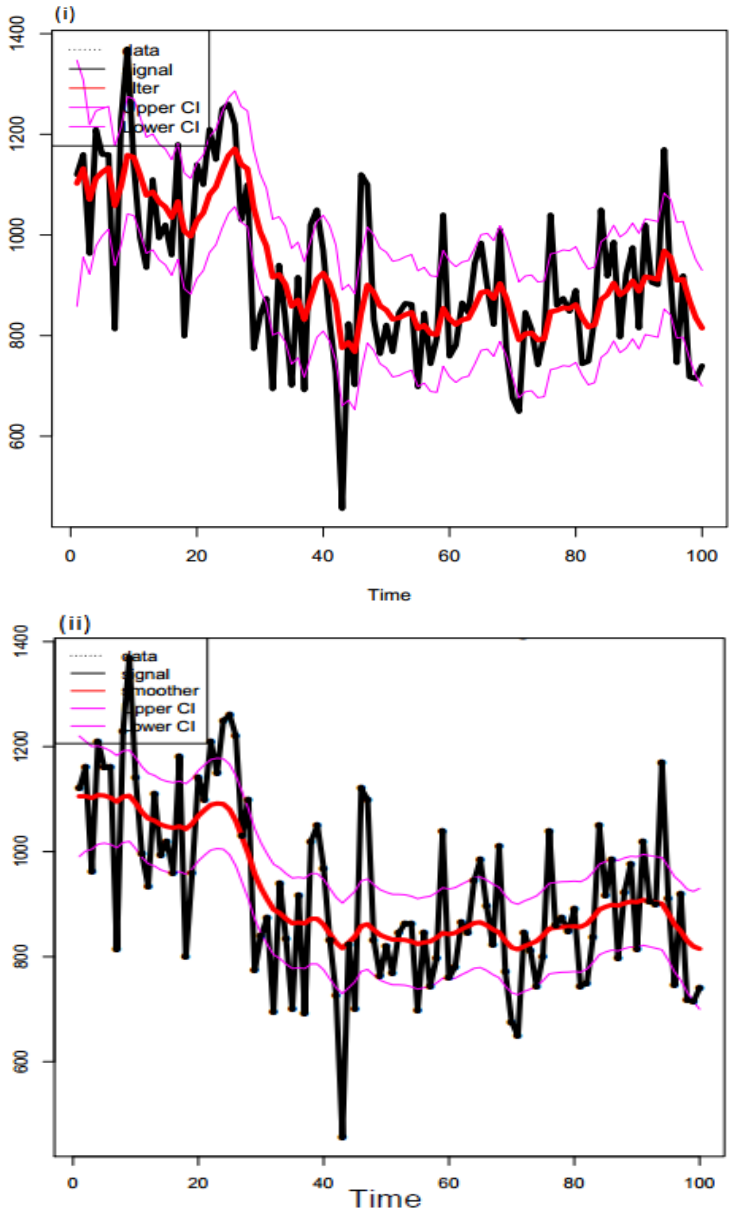


Figure 6: Nile data and output of Kalman filter: (i) simulated data, signal, filtered state and its 95% confidence intervals; (ii) simulated data, signal, smoothed state and its 95% confidence interval.

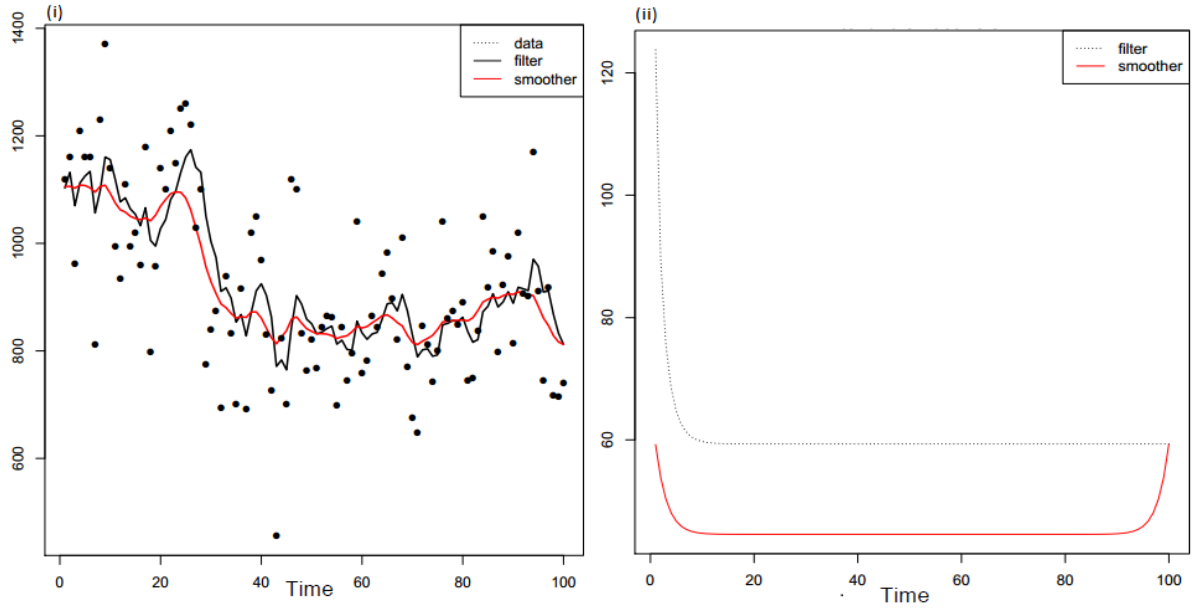


Figure 7: Nile data and output of Kalman filter: (i) Filtered and smoothed mean; (ii) Filtered and smoothed standard error.

Comparing the graphs obtained in (i) and (ii) of figure 6, the confidence interval associated with the smoother is narrower than the confidence interval for the filter and the smoother does indeed have a more even surface compared to the filter.

From figure 7, graph (i), it can be observed that the smoother mean is generally less than the filter mean for the process. Using the values for σ_η^2 and σ_ε^2 , a signal to noise ratio of 0.189 is calculated hence it is justifiable to say that the filtered values will not necessarily tend closely to the Nile data.

Looking at figure 7, graph (ii), it is very evident that the variance of the prediction error F_t and the state variance P_t converge rapidly to constant values meaning that the process has since stabilized and thus the local level model has a steady state solution.In addition, notice that the standard error at $t = 100$ which is the time that the last observation was measured is the same for the filter and the smoother. However from time $t = 0, 1, 2, ..., 99$, the smoother has a lower standard error than that of the filter. Paying specific attention to time $t = 50$ (halfway through the process) and using the fact that $(standard\,error)^2 = variance$, it can be observed that the filtering variance is greater than the smoothing variance. This is due to the fact that the filtering distribution only took into account observations from time $t = 0, ...50$ while the smoothing distribution has taken into account observations from time $t = 0, ...50, ...100$.

4 Conclusion

In this essay, we considered the Gaussian local level model and the use of the Kalman filter to obtain filtering and smoothing distributions. The Kalman filter was applied to a simulated data set and a Nile data set. By examining the results obtained from the simulated study, it is clear that the role of the signal to noise ratio is to determine whether the process has a signal (not just a random variation) or just noise (random variation). If the signal to noise ratio is less than 1, it means that the noise is relatively larger than the signal and thus

there is some variation. Alternatively, if the signal to noise ratio is greater than 1, the signal is relatively larger than the noise meaning that it is easier to extract information for the process and the information is more reliable. From the results obtained by applying the Kalman filter to the Nile data, it is evident that the Nile data shows various level shifts and this is due to the fact that the Ashwan high dam was constructed and therefore it impacted the annual flow of the Nile River.

In this paper, the Gaussian local level model was considered, however it is possible to consider other complex models of the state space model such as a non-linear non-Gaussian state space model as discussed by Kitagawa in [18, 17]. In addition, methods such as unscented filtering [30], Bayesian filtering [9, 26] and the extended Kalman filter [7, 11] can all be used as alternatives of the Kalman filter.

References

[1] E. Barnard, M.H. Davel, C. van Heerden, F. de Wet, and J. Badenhorst. The NCHLT speech corpus of the South African languages. *4th International Workshop on Spoken Languages Technologies for Under-resourced Languages*, pages 194–200, 2014.

[2] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato. SIFT features tracking for video stabilization. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 825–830. IEEE, 2007.

[3] E. Burmeister, K.D. Wall, and J.D. Hamilton. Estimation of unobserved expected monthly inflation using Kalman filtering. *Journal of Business & Economic Statistics*, 4(2):147–160, 1986.

[4] O. Cappé, E. Moulines, and T. Rydén. An information-theoretic perspective on order estimation. *Inference in Hidden Markov Models*, pages 565–601, 2005.

[5] C.K. Carter and R. Kohn. On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553, 1994.

[6] P. De Jong and N. Shephard. The simulation smoother for time series models. *Biometrika*, 82(2):339–350, 1995.

[7] R. Dhaouadi, N. Mohan, and L. Norum. Design and implementation of an extended Kalman filter for the state estimation of a permanent magnet synchronous motor. *Institute of Electrical and Electronics Engineers Transactions on Power Electronics*, 6(3):491–497, 1991.

[8] J. Durbin and S.J. Koopman. *Time Series Analysis by State Space Methods*. Number 38. Oxford University Press, second edition, 2012.

[9] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *Institute of Electrical and Electronics Engineers Pervasive Computing*, 2(3):24–33, 2003.

[10] S. Frühwirth-Schnatter. Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, 15(2):183–202, 1994.

[11] K. Fujii. Extended Kalman Filter. *Reference Manual*, 2013.

[12] A.C. Harvey. Applications of the Kalman filter in Econometrics. *Advances in Econometrics*, pages 285–313, 1987.

[13] R.V Hogg and A.T. Craig. *Introduction to Mathematical Statistics*. Macmillan, fourth edition, 1978.

[14] D.S Holmes and A.E. Mergen. Signal to noise ratio–what is the right size. *Quality Magazine*, pages 1–6, 2007.

[15] B.H. Juang, S.E. Levinson, L.R. Rabiner, and M.M. Sondhi. Hidden Markov model speech recognition arrangement, November 8 1988. US Patent 4,783,804.

[16] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[17] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[18] G. Kitagawa. A self-organizing state-space model. *Journal of the American Statistical Association*, pages 1203–1215, 1998.

[19] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

[20] N.D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in Neural Information Processing Systems*, 16(3):329–336, 2004.

[21] A.J. Lipton, H. Fujiyoshi, and R.S. Patil. Moving target classification and tracking from real-time video. In *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, pages 8–14. IEEE, 1998.

[22] H. Nielsen and A. Krogh. Prediction of Signal Peptides and Signal Anchors by a Hidden Markov Model. In *Intelligent Systems for Molecular Biology*, volume 6, pages 122–130, 1998.

[23] H.E. Rauch, C.T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *American Institute of Aeronautics and Astronautics journal*, 3(8):1445–1450, 1965.

[24] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive Background Estimation and Foreground Detection using Kalman-Filtering. In *Proceedings of International Conference on recent Advances in Mechatronics*, pages 193–199. Citeseer, 1995.

[25] S.I. Roumeliotis and G.A. Bekey. Bayesian estimation and Kalman filtering: A unified framework for mobile robot localization. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 3, pages 2985–2992. IEEE, 2000.

[26] Simo Särkkä. *Bayesian Filtering and Smoothing*, volume 3. Cambridge University Press, 2013.

[27] R.H. Shumway and D.S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer Science & Business Media, second edition, 2011.

[28] J.H. Stock and M.W. Watson. A simple estimator of cointegrating vectors in higher order integrated systems. *Econometrica: Journal of the Econometric Society*, pages 783–820, 1993.

[29] G. Strang and K. Borre. *Linear Algebra, Geodesy, and GPS*. Society of Industrial and Applied Mathematics, 1997.

[30] E.A Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee, 2000.

[31] G.F. Welch. History: The use of the Kalman filter for human motion tracking in virtual reality. *Presence: Teleoperators and Virtual Environments*, 18(1):72–91, 2009.

[32] S. Wu and Y. Zeng. Affine regime-switching models for interest rate term. In *Mathematics of Finance: Proceedings of an AMS-IMS-SIAM Joint Summer Research Conference on Mathematics of Finance, June 22-26, 2003, Snowbird, Utah*, volume 351, page 375. American Mathematical Soc., 2004.

[33] X. Yun and E.R. Bachmann. Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking. *Robotics, IEEE Transactions on*, 22(6):1216–1227, 2006.

Appendix

Multivariate Normal Distributions

Given random variables $X_1, X_2 ..... X_p$ such that $\underline{X}$: $p \times 1$ matrix distributed $\sim N_p(\underline{\mu}, \Sigma)$, then the p-dimensional normal density has the form:

$$f_{\underline{X}}(\underline{x}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}[(\underline{x}-\underline{\mu})' \Sigma^{-1} (\underline{x}-\underline{\mu})]}$$

where $-\infty < x_i < \infty, i = 1, 2...., p$. [13]

Univariate Normal

Given that $X \sim N(\mu, \sigma^2)$, the density function of X is defined as:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

where $-\infty < x < \infty$. [13]

Bivariate Normal (Joint density function)

Given that $\underline{X}$: 2×1, $\underline{X} \sim N(\underline{\mu}, \Sigma)$ *such that* $\underline{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ *and* $\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$, the joint density function is defined as:

$$f_{X_1,X_2}(x_1, x_2) = \frac{1}{(2\pi)\sigma_1\sigma_2\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 - 2\rho\left(\frac{x_1-\mu_1}{\sigma_1}\right)\left(\frac{x_2-\mu_2}{\sigma_2}\right) + \left(\frac{x_2-\mu_2}{\sigma_2}\right)^2\right]}$$

where $\rho = \frac{\sigma_{12}}{\sqrt{\sigma_1^2}\sqrt{\sigma_2^2}}$. [13]

Bivariate Normal (Conditional)

Given that $\underline{X}$: 2×1, $\underline{X} \sim N(\underline{\mu}, \Sigma)$ *such that* $\underline{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ *and* $\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix}$

We have that $X_1 \mid X_2 = x_2 \sim N(\mu_1 + \frac{\sigma_{12}}{\sigma_{22}}(x_2 - \mu_2), \sigma_{11} - \frac{\sigma_{12}^2}{\sigma_{22}})$ thus the conditional normal density of X_1 given that $X_2 = x_2$ is defined by:

$$f_{X_1|X_2}(x_1) = \frac{1}{\sqrt{2\pi}\sqrt{\sigma_{11} - \frac{\sigma_{12}^2}{\sigma_{22}}}} e^{-\frac{1}{2}\left[\frac{x_1 - (\mu_1 + \frac{\sigma_{12}}{\sigma_{22}}(x_2-\mu_2))}{\sqrt{\sigma_{11} - \frac{\sigma_{12}^2}{\sigma_{22}}}}\right]^2}$$

. [13]

Simulated local level model and the Nile data set (Code)

Simulation of local level model

```
# simulator of local level model
LLsim <- function(T,sigma_epsilon,sigma_eta)
{
    mY <-matrix(0,T,1) # store data
    mState <-matrix(0,T,1) # store state
    mState[1,1] = 2.0 # initial value
    for (i in 1:T ) {
        if (i>1) mState[i,1] = mState[i-1,1] + rnorm(1,0.0,sigma_eta)
        mY[i,1] = mState[i,1] + rnorm(1,0.0,sigma_epsilon)
    }
    Model.results = list(Y = mY, mu = mState) # return results
    return(Model.results)
}
```

Filtering

```
KF <- function(X,sigma_epsilon,sigma_eta)
{
    T = length(X) # length of data
    s_eps2 = sigma_epsilon^2; s_eta2 = sigma_eta^2;
    #setup some storage
    mMstore <-matrix(0,T,1)
    mM_Lstore <-matrix(0,T,1)
    mPstore <-matrix(0,T,1)
    mP_Lstore <-matrix(0,T,1)
    mVstore <-matrix(0,T,1)
    mFstore <-matrix(0,T,1)
    # initial values
    m = 0.0
    P = 1000000.0
    for (i in 1:T) { # KF recursions
        m_L = m # m_t|t-1
        P_L = P + s_eta2 # P_t|t-1
        aF = P_L + s_eps2
        v = X[i,1] - m_L
        K = P_L/aF
        P = P_L*(1.0-K) #P_t|t
        m = m_L + K*v # m_t|t
        mMstore[i,1] = m
        mM_Lstore[i,1] = m_L
        mPstore[i,1] = P
        mP_Lstore[i,1] = P_L
        mVstore[i,1] = v
        mFstore[i,1] = aF
    }
    Filter.output = list(mFilter = mMstore, Predict = mM_Lstore,
                         PFilter = mPstore, PPredict = mP_Lstore,
                         mErrors = mVstore, mErrVar = mFstore)
    return(Filter.output)
}
```

Smoothing

```
Smoother <- function(Filter.output){
    T = length(Filter.output$mFilter)
    mMsmostore <-matrix(0,T,1)
    mPsmostore <-matrix(0,T,1)
    mSmooth = Filter.output$mFilter[T,1]
    PSmooth = Filter.output$PFilter[T,1]
    mMsmostore[T,1] = mSmooth; mPsmostore[T,1] = PSmooth;
    for (i in (T-1):1) { # KF recursions
        H = Filter.output$PFilter[i,1]/Filter.output$PPredict[i+1,1]
        mSmooth = Filter.output$mFilter[i,1]
        + H*(mSmooth - Filter.output$mFilter[i+1,1])
        PSmooth = Filter.output$PFilter[i,1]
        + H*(PSmooth - Filter.output$PPredict[i+1,1])*H
        mMsmostore[i,1] = mSmooth
        mPsmostore[i,1] = PSmooth
    }
    Smooth.output = list(mMsmoo = mMsmostore, mPsmoo = mPsmostore)
    return(Smooth.output)
}
```

Simulation Smoother

```
SimSmoother <- function(Filter.output){
 n = length(Filter.output$mFilter)

 mSimStore   <-matrix(0,n,1)

 mSim = Filter.output$mFilter[n,1]+
 rnorm(1,0.0,sqrt(Filter.output$PFilter[n,1]))
 mSimStore[n,1] = mSim

 for (i in (n-1):1) { # KF recursions
  H = Filter.output$PFilter[i,1]/Filter.output$PPredict[i+1,1]

  StateSE = sqrt((1.0-H)*Filter.output$PFilter[i,1])

  mSim = Filter.output$mFilter[i,1]+H*(mSim - Filter.output$Predict[i+1,1])
  +rnorm(1,0.0,StateSE)

      mSimStore[i,1] = mSim
    }

    SimSmooth.output = list(SimSmooth = mSimStore)

    return(SimSmooth.output)
}
```

Maximum likelihood estimates for σ_η^2 and σ_ε^2 (Nile data set)

```
optim(c(0.5,0.2),logL_LL,Y=Y)
```

Log-likelihood function

```
logL <- function(mErr,mVar){
    logL1 = sum(-0.5*log(mVar) -0.5*(mErr*mErr/mVar))

    logL1
}

logL_LL <- function(mTheta,Y){

 sigma_epsilon = abs(mTheta[1]);   sigma_eta = abs(mTheta[2]);
 Kalman.output = KF(Y,sigma_epsilon,sigma_eta) # KF
 logL1 = logL(Kalman.output$mErrors,Kalman.output$mErrVar)
 # Gaussian logL

print(cbind(t(mTheta),logL1))

  return(-logL1)
}
```

Gibbs sampling

```
Gibbs_LL <- function(Y,iRep,S_epsilon,S_eta){
    n = length(Y)
    aprior = S_epsilon[1]; bprior = S_epsilon[2];
    aprior1= S_eta[1]; bprior1 = S_eta[2];

    mStore = matrix(0,iRep,2) # store results

    sigma_epsilon = sd(diff(Y,1));
    sigma_eta = sd(diff(Y,1)); # initial values for simulation

for (i in 1:iRep){
Kalman.output = KF(Y,sigma_epsilon,sigma_eta)
mSim = SimSmoother(Kalman.output)$SimSmooth #run simulation smoother
mEps = Y - mSim # measurement error
mEta = diff(mSim,1) #transition error

  # draw from posterior
  ap = (aprior+(0.5*n)); bp = bprior+(0.5*sum(mEps*mEps))
  ap1 = (aprior1+(0.5*(n-1))); bp1 = bprior1+(0.5*sum(mEta*mEta))

   sigma_epsilon = 1.0/sqrt(rgamma(1,shape=ap,scale=(1.0/bp)))
   sigma_eta = 1.0/sqrt(rgamma(1,shape=ap1,scale=(1.0/bp1)))

    #store results
    mStore[i,1] = sigma_epsilon
    mStore[i,2] = sigma_eta
    }

Gibbs_LL.output = list(SEepsilon = mStore[,1], SEeta = mStore[,2])

return(Gibbs_LL.output)
}
```

Education statistics: Blended/hybrid learning

Marené Cronje 13103599

STK795 Research Report

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

Supervisors: Dr L Fletcher and Ms LE Bodenstein

Department of Statistics, University of Pretoria

2 November 2016

Abstract

Online education is undeniably popular and accepted worldwide. Universities are concerned about measurable performance of student learning activities in web-based courses. Will student performances be equal or better when implementing more and more blended learning techniques? This report will discuss blended learning and the benefits of applying it. It will test if positive outcomes are attainable when introducing different blended learning tools. The specific statistical method which will be used is called a two-way analysis of covariance (ANCOVA) where we will compare the averages of continuous variables of independent groups while covariates are controlled. A hypothesis test will be conducted in which we will evaluate the F-test statistic and its p-value to determine whether the result is significant. In case a significant outcome is obtained, post hoc tests are applied to analyse the pattern of differences between the averages.

Declaration

I, *Marené Cronje,* 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.

_____
Marené Cronje

_____
Lizelle Fletcher and Loina Bodenstein

_____
Date

Acknowledgements

First I would like to articulate my appreciation to Dr L Fletcher and Ms LE Bodenstein for all their valuable insight, guidance and suggestions. I am also grateful to Dr. I Fabris-Rotelli for offering generous help and support as chairperson of the postgraduate committee. I would like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a post graduate bursary.

Contents

1 Introduction **6**

2 Background theory **7**
2.1 Methodology 7
2.2 Test procedure 8
2.2.1 The hypothesis test 8
2.2.2 The two-way ANCOVA table 9
2.2.3 The general linear model (GLM) 9
2.2.4 Effect size 10
2.2.5 Post hoc procedures 10

3 Application **11**
3.1 Problem description 11
3.2 Results 12

4 Conclusion **14**

References **15**

Appendix **16**

List of Tables

1 Two-way theoretical ANCOVA table 9
2 Descriptive statistics 12
3 Two-way ANCOVA table 13

1 Introduction

Graham challenged the width and ambiguity of the term "blended learning" and redefined "blended learning systems" as a combination of the traditional face-to-face learning environment and the distributed learning environment, using to some extent computer-mediated technologies [10]. The 21st century can be regarded as a transformation era in terms of exposure to technological communication and information. A new world with countless possibilities unfolded and it is nearly impossible for any educational institution to grow without including some form of a blended system. It is inevitable that new developments will alter the way we communicate and learn and therefore also the way we think. [10] is of the opinion that blended learning has become such a popular tool in education that the term "blended" will disappear and blended learning will be regarded as just learning.

Blended learning offers an enormous transformative potential, especially in higher education. Ultimately it is important for a university to recognise and explore how to best utilise both face-to-face learning (synchronous) and text-based internet learning (asynchronous) activities [8]. Information and communication technology tools provide flexibility in terms of time, place and diversity. This is beneficial for students as it offers some kind of control over the way they interact and learn [19]. Integrating the strengths of both these activities are challenging and requires reconceptualisation and re-organising of existing dynamics to be constructive and meaningful [1]. Different instructional methods can be incorporated into traditional university programmes. Universities need to cater for a diverse section of the population entering the blended learning environment. Adjustments for growing expectations and demands for better quantity and quality of learning experiences and outcomes are required.

Blended learning has been part of society ever since older communication tools such as radio, television and telephones have been available. Since the internet became more available and accessible, blended learning practices affected higher education substantially [17]. Current practices in blended learning include simulations, visualizations, communication and interactive technology e.g. immediate feedback on online assignments. Mobile learning is one of the largest practices currently applied, using social networking sites such as Facebook, MySpace, Flickr and Twitter as a tool to stay connected and improve communication between students as well as lecturers.[6] According to the Executive Committee of the National Council of Teachers of English in [10], the social impact of learning is increasingly related to the ability to use and combine digital tools in learning environments. Group effort on a social platform promotes problem solving as well as independent thought. Students are exposed to multiple amounts of information and learn how to process, analyze and conceptualize information. They interact and develop skills to master the technology behind different language of instruction tools whilst practising ethical behaviour.

According to the report of Staker and Horn [19] combinations of blended learning resulted in four suggested models. First, he mentioned the "Rotation" model which is applied when instruction is presented online but rooted in and combined with, constant supervision of a teacher who is available for face-to-face consultation and support. Instruction occurs in a cyclical manner. The suggested "Flex" model applies when multiple students are primarily engaged online, but with supervisory teacher capacity. The "Self-Blending" model is applicable to those students who choose to supplement their learning through participating in additional online courses offered by institutions. They do so in a setting where a supervising teacher and other students are co-present, or completely by themselves off-site. The "Enriched-Virtual" model is used when learning occurs on an online platform, with optional teacher check-ins, face-to-face.[19] Graham suggested that new pathways will evolve from scratch from initially blended environments to meet the emerging needs in education and training. The choice of the blend is important to universities and requires presentation tools to be reliable and consumer friendly. It is essential to have high quality technical support as backup, as well as trained instructors.[2] Overlapping of models often occurs depending on individual or institutional needs. The University of Pretoria makes use of a combination of all four models in some way or another.

It is the opinion of Graham that blended learning occurs at many levels e.g. institutional-, programme-, course-, and activity levels. Participation at course and activity levels has instructor stakeholders and is primarily focussed on learning effectiveness and productivity. Blended learning at programme and institutional levels have administrator stakeholders who are mainly interested in issues of cost effectiveness and expanding learning access.[10] Universities will strive to adopt the instructor stakeholder level where the focus is on the students and to promote their learning abilities, but at the same time they necessarily need to be aware of the cost implications and expansion of learning access with new technological accessories and tools.

Graham mentioned three core benefits of blended learning. Firstly he spoke of improved learning effectiveness and enhanced skills in problem solving. Secondly mentioned is increased access to up-to-date resources, time flexibility and freedom over time, place and pace. One can also expect the convenience offered by an online environment. Lastly he mentioned increased cost effectiveness and reduction in teaching costs as a major advantage.[10] Arfield described other advantages such as positive attitude adjustments of students towards learning. Overall communication improvements and instant measures of academic progress of students are also possible due to the availability of student performance data.[2] Alfred et al. mentioned a benefit of blended learning as being an increased perception of belonging to a study community which were experienced when mutual participation took place [18]. Hughes is of the opinion that an increased student support via e-learning led to improved student retention rates [13].

We live in a blended era and advantages of blended learning are countless. Models and the integration thereof are here to stay. New initiatives will be tested and applied, not necessarily to only create new models, but also to identify and make use of the strengths of each environment [16].

Blended learning at the University of Pretoria will be investigated using a first year module (STK110) when various interventions were introduced over a period of 5 years, i.e. 2011 to 2015.

2 Background theory

2.1 Methodology

An analysis of variance (ANOVA) model is appropriate when a continuous dependent variable is predicted by at least one categorical explanatory variable with two or more levels (treatments). ANOVA refers to a case where independent samples are drawn from several populations and their averages compared. A two-way ANOVA model refers to a cross classification of two or more factors where averages of the dependent variable between two groups are compared with the main purpose to determine whether interaction between the two independent variables on the dependent variable exist. A two-way analysis of covariance (ANCOVA) model is applicable when an independent continuous variable, called the covariate, is also added to the model. This covariate has a relationship with the dependent variable but is not influenced by the factors. Adding the covariate and also using more than one factor, is expected to increase the goodness of fit of the model and thereby also increase the accuracy of the estimates.[11]

Using a two-way ANOVA involves a randomized block design where the treatments are referred to factor A and the blocks to factor B. Blocks are the levels at which we hold an extraneous factor fixed so that we can measure its contribution to the total variation of the data. The treatments are distributed at random within each of the blocks and each treatment appears only once in each block. The levels of the first factor as well as the levels of the second factor are compared. We are primarily interested in testing the treatment averages per block. Testing for a difference in block averages is equivalent to test whether the blocking was effective in removing the extraneous source of variation. If there are no differences among the block averages, we conclude that blocking is not effective in reducing the variability. In ANCOVA the focus is on the analysis of the effect of the factor levels, holding the covariate constant. Adding a covariate to the model can significantly affect the final results and improve the accuracy of the model.[11]

2.2 Test procedure

2.2.1 The hypothesis test

A hypothesis test is conducted to test on a 5% level of significance whether differences between the averages of the dependent variables for the different levels are significant, and also whether blocking on students was effective in reducing variation in their average marks. The data is collected according to a randomized block design in which the treatments are distributed at random within each of the blocks and each treatment appears once in each block. The null hypothesis, $H_0 : \mu_1 = \mu_2 = \mu_3 = \ldots = \mu_k$ (equal population averages for the treatments) is tested against the alternative hypothesis, $H_1 : \mu_i \neq \mu_j$ for at least one pair $i \neq j; i, j = 1, 2, \ldots, k$ are not equal, that is at least two population averages differ from each other.

We assume that the averages of the populations are normally distributed with a common unknown variance. We have to estimate the variance from the sample information. The total sum of squares (SST) for the randomized block design is partitioned into three parts, namely

$$SST = SSA + SSB + SSE$$

where,

$$SSA = \text{Treatment sum of squares}$$

$$SSB = \text{Block sum of squares}$$

$$SSE = \text{Error sum of squares}$$

The first source of variation is the variation between samples and is measured by the variation of the sample averages about the overall sample mean, namely

$$\begin{aligned} MSA &= \frac{n_1(\bar{Y}_1 - \bar{Y})^2 + n_2(\bar{Y}_2 - \bar{Y})^2 + n_3(\bar{Y}_3 - \bar{Y})^2 + \ldots + n_k(\bar{Y}_k - \bar{Y})^2}{n-1} \\ &= \frac{SSA}{n-1} \end{aligned}$$

$$\begin{aligned} MSB &= \frac{n_1(\bar{Y}_1 - \bar{Y})^2 + n_2(\bar{Y}_2 - \bar{Y})^2 + n_3(\bar{Y}_3 - \bar{Y})^2 + \ldots + n_k(\bar{Y}_k - \bar{Y})^2}{n-1} \\ &= \frac{SSB}{n-1} \end{aligned}$$

These quantities are called the between-sample variations. The quantity in the numerator is denoted by SSA, the treatment sum of squares, and the quantity in the denominator is the number of samples minus one. We can say MSA or MSB is based on $(n-1)$ degrees of freedom. The total number of treatments is denoted by n and we subtract one from it for the estimation for the overall mean.

The second source of variation is the variation within samples and it is measured by a pooled estimator of the variance based on individual variances of all the samples, namely

$$\begin{aligned} MSE &= \frac{\sum_{i=1}^{n_1}(Y_{i1} - \bar{Y}_1)^2 + \sum_{i=2}^{n_2}(Y_{i2} - \bar{Y}_2)^2 + \sum_{i=3}^{n_3}(Y_{i3} - \bar{Y}_3)^3 + \ldots + \sum_{i=j}^{n_k}(Y_{ij} - \bar{Y}_j)^k}{n_1 + n_2 + n_3 + \ldots + n_k - k} \\ &= \frac{(n_1-1)S_1^2 + (n_2-1)S_2^2 + (n_3-1)S_3^2 + \ldots + (n_k-1)S_k^2}{n_1 + n_2 + n_3 + \ldots + n_k - k} \\ &= \frac{SSE}{n-k} \end{aligned}$$

This quantity is called the within-sample variation. The numerator is denoted by SSE, the error sum of squares. MSE measures unexplained variation, in this case variation unexplained by the differences between sample averages. MSE is based on $n-k$ degrees of freedom. The total number of treatments is denoted by n. We subtract k for each of the treatment averages being estimated.[11]

2.2.2 The two-way ANCOVA table

Assume that a is the number of treatments and b is the number of blocks in the following table for a two-way ANCOVA:

Source of variation	Df	Sum of squares	Mean sum of squares	F-test statistic
Factor A (Treatments)	$a-1$	SSA	MSA	$\frac{MSA}{MSE}$
Factor B (Blocks)	$b-1$	SSB	MSB	$\frac{MSB}{MSE}$
Covariate	1	SS_{Cov}	MS_{Cov}	$\frac{MS_{Cov}}{MSE}$
Error	$N-a-b$	SSE	MSE	
Total	$N-1$	SST		

Table 1: Two-way theoretical ANCOVA table

The test statistic for testing the effect of factor A is

$$f = \frac{MSA}{MSE}$$

The test statistic for testing the effect of factor B is

$$f = \frac{MSB}{MSE}$$

The test statistic for testing the effect of the covariate is

$$f = \frac{MSC}{MSE}$$

2.2.3 The general linear model (GLM)

A GLM for an ANCOVA is appropriate when the explanatory variables of interest are categorical, but a continuous, observed value which is the covariate is controlled.[12] The relationship between a continuous dependent variable y and explanatory variables $x_1, x_2, \ldots, x_k$ can be expressed in terms of a linear regression model,

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k \tag{1}$$

where the regression coefficients $\beta_0, \beta_1, \ldots, \beta_k$ are unknown constants which have to be estimated from the data. The explanatory variables $x_1, x_2, \ldots, x_k$ may be categorical or a mixture of categorical and continuous variables. It is assumed that the distribution of y within each subpopulation is normal with equal variances.

2.2.4 Effect size

Effect size allow us to move beyond the simplistic "significant or not" statement to a more sophisticated "how well and with what magnitude does it work?" statement. It is valuable for quantifying the size and effectiveness of an intervention and the difference between two groups relative to some comparison. The size of the effect has practical importance in reporting and interpreting effectiveness. It places emphasis on the most important aspect of an intervention, the size of the effect, rather than its statistical significance. Effect size is thus the magnitude of the difference between groups.[4]

2.2.5 Post hoc procedures

When the null hypothesis is rejected, ANOVA gives a significant result thus that not all means are equal which indicates that at least one group differs on average from the other groups. Post hoc tests are conducted as follow-up analyses when additional exploration of the differences among the groups are needed to provide specific information on exactly which means for the specific groups are significantly different from each other, by using pairwise comparisons, multiple pairwise comparisons or planned comparisons.[7] Various post hoc tests can be used to compare averages. Fisher's Least Significant Difference (LSD) test, Tukey's Honest Significant Difference (HSD) test, the Dunnett test, the Sidak test, the Bonferroni test and the Sheffé test are some of the tests that will be explained and considered to do post hoc analyses with.

Fisher's LSD test

Fisher's LSD test evaluates the averages of the groups by using t-tests. It compares all possible pairwise comparisons by calculating a set of individual t-tests. The LSD test shows increasing power in that it computes the pooled standard deviation from all groups. The LSD is the smallest significant difference between two averages and they are declared significant if their values are larger than the LSD.[12] The equation for the LSD is,

$$LSD = t \times \sqrt{\frac{2 \times MSE}{n}}$$

where t is the critical value with the associated degrees of freedom and n is the number of groups compared and used to calculate the averages of interest.

Tukey's HSD test

Tukey's HSD test calculates a new critical value which involves the average difference that has to be exceeded to achieve significant results. It can be used to calculate the difference between all possible pairwise averages so that each difference can be compared to the critical HSD value to test if the comparison is significant. All possible comparisons are required before the HSD test can be a powerful test and can be used when sample sizes are unequal or confidence intervals are lacking. It is conservative and therefore more likely to detect differences if they exist.[15] The HSD equation can be written as

$$HSD = q \times \sqrt{\frac{MSE}{n}}$$

where MSE is the mean square error and n is the number of q is the studentised range statistic equated as

$$q = \frac{\bar{y}_1 - \bar{y}_2}{MSE}$$

and where μ_1 is for example the largest mean of the two. MSE is the mean square error of the F-test and n is the sample size of each group.

The Dunnett test

This test is used for pairwise multiple comparisons and it's based on a t-test that compares a set of treatments against a single control average. It compares a single group average against all other group averages.

$$D_{Dunnett} = t_{Dunnett} \times \sqrt{\frac{2MSE}{n}}$$

where $D_{Dunnett}$ is the difference which will be used for comparison and $t_{Dunnett}$ is the new t-test statistic. If the two averages are greater than $D_{Dunnett}$, a significant difference between the two averages exist in which the control average is used.[14]

The Bonferroni test

The overall error rate across statistical tests conducted on the same experimental data is known as the familywise error (FWE) rate and it gives the chance of having made at least one Type 1 error. The FWE rate can be calculated using the equation $\alpha_{FWE} = 1-(1-0.05)^n$ where n is the number of tests carried out on the data. This test uses t-tests to perform pairwise comparisons between group averages. The Bonferroni correction controls the FWE by calculating a new pairwise alpha value by dividing the FWE rate by the number of comparisons to ensure that the cumulative Type 1 error is below 0.05. Controlling the FWE rate causes the Bonferroni test to lack power. By being more conservative in the Type 1 error rate for each comparison, the probability of rejecting an effect that does actually exist (Type 2 error) is increased.[7]

The Šidák test

The Šidák test is also a method to control the FWE rate only when the comparisons are independent. This test uses an adjusted p-value calculated as $1-(1-unadjusted\ p-value) \times k$, where k is the number of comparisons in the family of comparisons.[14]

The Scheffé test

The Scheffé test computes a new critical value, $(n-1)F_{0.05}$ after the number of groups that are compared are taken into account [20]. This new value represents the critical value for the maximum possible FWE rates. Hair et al. described the Scheffé test as conservative in cases when pairwise comparisons are the only comparisons of interest and it also results in a higher than desired Type II error rate [12]. According to the Sheffé method of multiple comparisons, two population averages are considered to be significant if,

$$|\bar{y}_1 - \bar{y}_2| > \sqrt{(n-1)F_{0.05}MSE(\frac{1}{n_1}+\frac{1}{n_2})}.$$

3 Application

3.1 Problem description

In this report, the final marks of first year level Statistics at the University of Pretoria were analysed to determine if the marks varies according to the years at which different blended learning techniques were implemented and therefore to evaluate whether blended learning has had any effect on student performance. A hypothesis test will be conducted on a 5% level of significance by comparing the averages of the first year first semester (STK110) final marks for the different years (treatments), using an ANCOVA test. The continuous variable of interest is therefore STK110 final marks and it may be influenced by at least two factors which are potential sources of variation. These factors are known s the categorical independent variables in the model. The first factor in the model was the different years (2011 to 2015) in which the techniques were implemented and the block factor was language. Language was taken into consideration since the language in which students speak at home and the language in which they receive teaching has an impact on their marks if these two differ from each other.[9] A new dummy variable was created called "language" which was equal to one if the language of preference of the students is equal to their home language and zero otherwise. The

language factor helped to try and reduce the existing unexplained variation and to test whether blocking on students was effective in reducing variation in the average STK110 marks. The first year students' grade 12 mathematics results were used to represent the covariate, mathematics. According to the article of Chimka et al. students with better mathematics marks are more likely to pass at tertiary level.[3] Mathematics is therefore expected to have an influence on their STK110 final marks. Data sets will be taken from consecutive years where a new add-on blended learning technique was implemented in each year.

In 2011 no blended learning techniques were implemented and no interventions existed. This year will be regarded as the baseline year. In 2012 the Department of Statistics started intervening by implementing an online platform called Aplia. Aplia is an educational technology company which can be used by students mainly to do the homework assignments or tests after each STK110 lecture. This was a means for lecturers to evaluate students knowledge gained during lectures. In 2013 a flipped classroom was introduced for the first time in this module. This technique required students to do online assignments in Aplia before a STK110 lecture commenced. It was employed in order for students to be well prepared and to have a better understanding of the content of the work beforehand. In 2014 the flipped classroom technique was also used, but in a more refined way than the previous year. Students were required to do extensive self preparation before class, using Aplia. During the last year, 2015, more blended learning techniques were implemented. In this year, in addition to Aplia and the flipped classroom, clickers and Mindtap were also introduced. Clicker is an interactive "keypad" response software tool students used to answer test questions in class where the lecturer could immediately collect and view the responses of the entire class. Mindtap is a cloud-based and personal learning program provided by Cengage Learning when students could access their course materials, e-textbook, homework and quizzes, etc.

The data set received from the University of Pretoria's Bureau for Institutional Research and Planning (BIRAP) had to be filtered in order for the data to be usable and comparable before it could be used to do analyses on. In order to work with STK110 mainstream date we decided to exclude any anti-semester and winter school students. Module repeaters had to be filtered out since we worked only with the students who were exposed to the module for the first time. Students were allowed to take the module if they had a mark of 60% or more for their grade 12 mathematics subject. Some students still managed to enter the system with a lower mark than 60% in maths though. These students were removed from the data set as well.

To test if there is a significant relationship between blended learning and students' average final marks for the module, the null hypothesis, $H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5$, of equal population averages is tested against the alternative hypothesis, $H_1 : \mu_i \neq \mu_j$ for at least one pair $i \neq j; i, j = 1, 2, 3, 4, 5$ i.e. not all population averages are equal or at least two differ from each other. We assumed that the marks for every year are normally distributed with a common unknown variance σ^2 that has to be estimated from sample information. The variation in the marks consists of two components namely the variation explained by the differences between the five years as well as the variation within each of the five years' marks.

3.2 Results

Term	N	Mean	Standard deviation
2011	1617	55.5708	14.1122
2012	1206	58.4784	14.0467
2013	1351	64.6462	14.774
2014	1394	61.0897	15.93
2015	1415	64.617	15.6223

Table 2: Descriptive statistics

The data analysis for this essay was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA. The means procedure is used in SAS to obtain descriptive statistics in which we can evaluate the equality of variances. Techniques are robust if they maintain their statistical properties under violations of assumptions. From the output in Table 2 it is evident that the variances are very close to each other and since the sample sizes are fairly equal, i.e. $n_1 \approx n_2 \approx n_3 \approx n_4 \approx n_5$,we relied on the robustness of the test. Assume homogeneity of variances. [5]

Source of variation	Df	Sum of squares	Mean sum of squares	F-test statistic	p-value
Term	4	88088.8189	220022.2047	99.33	$<$0.0001
Language	1	1730.1304	1730.1304	7.8	$<$0.0001
Maths	1	4340.44123	4340.44123	19.58	0.0052
Error	6976	1546556.058	221.943		
Total	6982	1644270.652			

Table 3: Two-way ANCOVA table

A two-way ANCOVA model with five treatments, a blocking factor and a covariate is written, using Equation 1 as follows:

$$E(y) = \beta_0 + \beta^A x^A + \beta^B x^B + \beta_1 x_1$$

where,

$\mu = \beta_0$, i.e. the intercept term in the model

x^A = Factor A

(treatment i.e terms: 2011, 2012, 2013, 2014, 2015)

x^B = Factor B

(language: 1 when home language=language of preference, 0 otherwise)

x_1 = The covariate, i.e. grade 12 mathematics marks of students.

The F-test for the model (f=73.46) is highly significant (p-value$<$0.0001) of Output 4 in the Appendix. From Output 5 in the Appendix it follows that the treatment- and block factors as well as the covariate are significant. It means that the average marks of the 5 years differ significantly (f=99.33 and p-value$<$0.0001) as well as the average marks for language (f=7.8 and p-value=0.0052) while controlling for the covariate i.e. mathematics marks of students. The effect of the covariate is also significant (f=19.58 and p-value$<$0.0001).

It is important to note that we are working with a large sample. Large samples imply high statistical power, i.e. even small differences between groups may be statistically significant. This begs the question whether these differences are of any practical importance. The "effect size" statement in SAS is used to evaluate this problem. Cohen suggested guidelines for interpreting specific effect size measures. These suggested levels at which we'd describe an effect as either small, medium or large are 0.01, 0,06 and 0.14 respectively for partial η^2.[4] Partial η^2 is an appropriate measure of effect size for ANCOVA to describe the practical significance of differences observed between each year. The partial η^2 for term is 0.0558, which is a medium effect according to the Cohen's guidelines. The partial $\eta^2$for maths are 0.0028 and for language 0.0011. Both these variables have small effects according to Cohen's guidelines. However, the University of Pretoria is a large educational institution with a large number of students enrolled for the STK110 module. Even small increases from year to year will still have a relatively large impact and outcomes will still remain significant.

We used the Tukey statement in SAS to determine which groups differ significantly from one another. From SAS Output 6 in the Appendix, and from Table 1 it can be seen that there was a year-on-year increase in the final mark of students, with the exception of 2013 to 2014. In this year no new intervention was introduced. A significant difference was found from 2011 and 2012, 2012 and 2013, and 2014 and 2015.

4 Conclusion

It can be assumed that the techniques that were implemented added substantial value since all average marks were significantly different from one another. Each year's average marks increased compared to the average marks in 2011 (the baseline year). In 2012 when Aplia was implemeted, the average marks increased. In 2013 when the flipped classroom effect was introduced, the average marks increased from 2012. But in 2014, when the effect of the flipped classroom was more refined, the average marks decreased from 2013. The last year, 2015, when the flipped classroom, clickers and Mindtap were implemented, the highest average marks had been achieved comparative to all the other years.

Since we have observed significant results when applying an increasing number of blended learning tools, we can conclude that these kind of applications will enhance student performance and learning. The statistical method, a two-way ANCOVA, was successful in demonstrating that when more and more blended learning techniques are implemented, student performances will be enhanced. Positive outcomes have been attained after different blended learning tools were introduced and when various interventions took place.

It can be concluded that Factor B (language) as well as the covariate (mathematics marks of students) are useful and significant predictors. Blocking was effective in reducing the extraneous source of variation and thus decreasing the error term in the model. The covariate that was added to the model, improved the accuracy of the model. To further refine the model, more factors can be added, for instance, a student's gender or the area in which students live, so that the goodness of fit of the model can be increased.

References

[1] Sife Alfred, Edda Lwoga, and Camillius Sanga. New technologies for teaching and learning: Challenges for higher learning institutions in developing countries. *International Journal of Education and Development using ICT*, 3(2), 2007.

[2] John Arfield, Keith Hodgkinson, Alison Smith, and Winnie Wade. *Flexible Learning in Higher Education*. Routledge, 2013.

[3] Justin R Chimka, Teri Reed-Rhoads, and Kash Barker. Proportional hazards models of graduation. *Journal of College Student Retention: Research, Theory & Practice*, 9(2):221–232, 2007.

[4] Jacob Cohen. Statistical power analysis. *Current Directions in Psychological Science*, 1(3):98–101, 1992.

[5] Ralph B D'Agostino, Lisa M Sullivan, and Alexa S Beiser. *Introductory Applied Biostatistics*. Thomson Brooks/Cole, 2006.

[6] Meredith DeCosta, Jennifer Clifton, and Duane Roen. Collaboration and social interaction in english classrooms. *The English Journal*, 99(5):14–21, 2010.

[7] Andy Field. *Discovering Statistics Using IBM SPSS Statistics*. Sage, 2013.

[8] D Randy Garrison and Heather Kanuka. Blended learning: Uncovering its transformative potential in higher education. *The Internet and Higher Education*, 7(2):95–105, 2004.

[9] Ans Gerber, Johann Engelbrecht, Ansie Harding, and John Rogan. The influence of second language teaching on undergraduate mathematics performance. *Mathematics Education Research Journal*, 17(3):3–21, 2005.

[10] Charles R Graham. Blended learning models. *Encyclopedia of Information Science and Technology*, pages 375–382, 2009.

[11] Damoder N Gujarati. *Basic Econometrics*. Tata McGraw-Hill Education, 2009.

[12] Joseph F Hair, William C Black, Babin Barry J, and Rolph E Anderson. *Multivariate Data Analysis*. Pearson, 2009.

[13] Gwyneth Hughes. Using blended learning to increase learner support and improve retention. *Teaching in Higher Education*, 12(3):349–363, 2007.

[14] Douglas C Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 2008.

[15] R O'Neill and GB Wetherill. The present state of multiple comparison methods. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 218–250, 1971.

[16] Russell T Osguthorpe and Charles R Graham. Blended learning environments: Definitions and directions. *Quarterly Review of Distance Education*, 4(3):227–33, 2003.

[17] David Parsons. *Refining Current Practices in Mobile and Blended Learning: New Applications*. IGI Global, 2012.

[18] Alfred P Rovai and Hope Jordan. Blended learning and sense of community: A comparative analysis with traditional and fully online graduate courses. *The International Review of Research in Open and Distributed Learning*, 5(2), 2004.

[19] Heather Staker and Michael B Horn. Classifying k-12 blended learning. *Innosight Institute*, 2012.

[20] Abraham Gert Willem Steyn, Christian Frederick Smit, and Stephen Henry Charles Du Toit. *Moderne Statistiek vir die Praktyk*. Van Schaik, 1982.

Appendix

Output 1: SAS code used to build a two-way ANCOVA model:

```
/*'Differences among the average marks for five years';
'Two-way analysis of covariance';
'Use PROC GLM to test for differences among five years averages';*/

data STK110;
set use;
/*Creating a dummy variable for language*/
Language=0;
if Home_language='English' and Language_Of_Preference='English' then Language=1;
if Home_language='Afrikaans' and Language_Of_Preference='Afrikaans' then Language=1;
if Home_language='Afrikaans/English' and Language_Of_Preference='English' then
Language=1;
if Home_language='Afrikaans/English' and Language_Of_Preference='Afrikaans' then
Language=1;
run;

title1 'Descriptive statistics';
proc sort data=STK110;
by Term;

proc means data=STK110 n mean std;
by Term;
var Final_mark;
run;

title;
title2 'Tukeys HSD as a post hoc test and effect size';

proc glm;
class Term;
model Final_mark = Term Maths Language /effectsize;
means Term / tukey;
run;
```

Output 2: Descriptive statistics

Descriptive statistics

Term=2011

The MEANS Procedure

Analysis Variable : Final_Mark

N	Mean	Std Dev
1617	55.5708101	14.1122213

Term=2012

Analysis Variable : Final_Mark

N	Mean	Std Dev
1206	58.4784411	14.0466602

Term=2013

Analysis Variable : Final_Mark

N	Mean	Std Dev
1351	64.6461880	14.7739731

Output 2 continued: Descriptive statistics

Term=2014

Analysis Variable : Final_Mark

N	Mean	Std Dev
1394	61.0896700	15.9300290

Term=2015

The MEANS Procedure

Analysis Variable : Final_Mark

N	Mean	Std Dev
1415	64.6169611	15.6222739

Output 3: The GLM procedure

Tukeys HSD as a post hoc test and effect size

The GLM Procedure

Class Level Information

Class	Levels	Values
Term	5	2011 2012 2013 2014 2015

Number of Observations Read 6983
Number of Observations Used 6983

Tukeys HSD as a post hoc test and effect size

The GLM Procedure

Dependent Variable: Final_Mark

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	97714.594	16285.766	73.46	<.0001
Error	6976	1546556.058	221.697		
Corrected Total	6982	1644270.652			

Output 3 continued: The relevant F- and p-values

R-Square	Coeff Var	Root MSE	Final_Mark Mean
0.059427	24.50396	14.88948	60.76357

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Term	4	88088.81887	22022.20472	99.33	<.0001
Maths	1	4340.44123	4340.44123	19.58	<.0001
Language	1	1730.13037	1730.13037	7.80	0.0052

Total Variation Accounted For

Source	Semipartial Eta-Square	Semipartial Omega-Square	Conservative 95% Confidence Limits	
Term	0.0536	0.0530	0.0434	0.0637
Maths	0.0026	0.0025	0.0008	0.0056
Language	0.0011	0.0009	0.0001	0.0031

Output 4: Tukey's HSD edited test results

Critical Value of Studentized Range 3.85867

Comparisons significant at the 0.05 level are indicated by ***.

Term Comparison	Difference Between Means	Simultaneous 95% Confidence Limits		
2013 - 2014	3.5565	2.0055	5.1075	***
2014 - 2015	-3.5273	-5.0604	-1.9942	***
2012 - 2013	-6.1677	-7.7772	-4.5583	***
2011 - 2012	-2.9076	-4.4533	-1.3619	***

Interrater agreement

Cairstine de Kock 13018656

STK795 Research report

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

Supervisors: Dr L Fletcher and Dr EM Louw

Department of Statistics, University of Pretoria

2 November 2016

Abstract

The kappa statistic (κ) was introduced by Cohen in 1960 as a measure of interrater agreement between two independent raters who allocate a fixed amount of subjects into the same number of categories on a nominal scale.[2, 11] Kappa has different uses, advantages and disadvantages as well as important assumptions associated with it, which will be discussed in this essay.

One of the extensions of kappa that will be discussed is weighted kappa (κ_w), which was developed by Jacob Cohen in 1968 [3] as a generalisation of the kappa statistic. Weighted kappa is used in situations where data, classified into ordinal categories, are being analysed and weights need to be assigned to the categories according to the importance of each category [3].

Another measure of interrater agreement that will be explained in this essay is Fleiss' kappa statistic, which is an improvement upon Cohen's unweighted kappa statistic. Fleiss developed this statistic in 1971 [6] to allow for measurement of chance-corrected agreement among any constant number of raters.

Formulae to calculate these statistics will be explained in this essay. Interpretation of the results using a practical dermatology example as well as a practical example based on data used in a doctoral thesis, will be done.

Declaration

I, *Cairstine de Kock,* 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.

_____
Cairstine de Kock

_____
Dr L Fletcher and Dr EM Louw

_____
Date

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisors, Dr L Fletcher and Dr EM Louw, for their continual support, guidance, patience and immense knowledge. I would also like to thank Dr PM Joubert for the use of some of his PhD thesis data in my research. My sincere thanks go to the Centre for Artificial Intelligence Research (CAIR) and STATOMET for their financial support in the form of postgraduate bursaries, making it possible for me to complete my Honours Degree in Statistics.

Contents

1 Introduction **6**

2 Background theory **6**

3 Application **9**
3.1 Evaluating two raters . . . 9
3.1.1 Problem setting for two raters . . . 9
3.1.2 Results . . . 10
3.1.3 Interpretation and discussion . . . 11
3.2 Evaluating more than two raters . . . 12
3.2.1 Problem setting for more than two raters . . . 12
3.2.2 Results . . . 12
3.2.3 Interpretation and discussion . . . 13

4 Conclusion **13**

Appendix **16**

List of Figures

1 Agreement plot for four categories of skin conditions . . . 11

List of Tables

1 Kappa Guidelines . . . 7
2 Four categories of skin conditions . . . 9
3 Three categories of skin conditions . . . 10
4 Two categories of skin conditions . . . 10
5 Cohen's kappa and weighted kappa values . . . 10
6 Cicchetti-Allison agreement weights for the four categories of skin conditions in Table 2 . . . 10
7 Fleiss-Cohen agreement weights for the four categories of skin conditions in Table 2 . . . 11
8 Rater classification for patient A . . . 12
9 Rater classification for patient B . . . 13
10 Fleiss' kappa value for each patient . . . 13

1 Introduction

There are two main types of agreement; interrater agreement and intrarater agreement, which differ from each other as follows. Intrarater agreement is the agreement that originates from the same person evaluating the same group of subjects at different points in time and the McNemar chi-square test for comparison of paired proportions is used in this case. With intrarater agreement the k x k contingency table of joint categorical assignment frequencies is used to calculate a contingency coefficient, C, based on the chi-square statistic, χ^2, to measure agreement in the k categories [3]. Interrater agreement is the agreement that originates from two or more independent raters evaluating the same group of subjects [4]. The kappa statistic is used as a measure of this type of agreement.

In this essay, the kappa statistic (κ), introduced by Cohen in 1960 [2, 11] and the weighted kappa statistic (κ_w), introduced by Cohen in 1968 [3, 11], will be reviewed. Kappa is a statistical measure used to quantify and test for the degree of interrater agreement between two independent raters who allocate a fixed amount of subjects into categories on a nominal scale [6].

Chance plays a prominent role in this process of categorisation and influences the degree of interrater agreement between the raters. Cohen's kappa attempts to remove the element of chance by taking into account the chance-expected agreement when calculating kappa, thus only measuring the interrater agreement beyond chance [11, 4, 2]. The formulae to illustrate this will be explained as well as the advantages and disadvantages when using kappa, different uses of kappa and the interpretation of the statistic.

Weighted kappa (κ_w) was developed by Cohen in 1968 [3] as a generalisation of the kappa statistic. This statistic is used in situations where data, classified into ordinal categories, are analysed and weights need to be assigned to the categories according to the importance of each category.

Fleiss addressed one of the restrictions of kappa in [6], by improving on Cohen's unweighted kappa statistic to allow for measurement of chance-corrected agreement among any constant number of raters, instead of just two raters. The formulae to illustrate this will also be explained.

The objective of this essay is to investigate the use and restrictions of Cohen's kappa, weighted kappa and Fleiss's improvement of Cohen's unweighted kappa statistic. This will be done through the interpretation of the results of a practical dermatology example and a practical example which is based on data used in a doctoral thesis that investigated emotionally triggered involuntary violent behaviour (ETIVB).

2 Background theory

There are two uses for the kappa statistic: to test rater independence and to quantify the level of agreement[1]. The kappa statistic was developed by Cohen in 1960 to measure agreement between two independent raters, who rate a fixed amount of subjects on a nominal scale of k categories [2, 11]. Kappa is defined as the agreement beyond chance divided by the total amount of possible agreement beyond chance [4]. Cohen's kappa is limited to the case of only two raters and where the same two raters rate the same group of subjects [6].

Cohen proposed the following assumptions for the kappa statistic [2]: all the measured units are independent of one another, the k categories of the nominal scale are independent, mutually exclusive and exhaustive and the investigators or raters work independently. The results can be displayed in a k x k contingency table.

The formula for calculating kappa is [10, 2, 3] :

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{1}$$

where P_o is the observed probability of agreement, i.e. the relative frequencies in the diagonal cells of the square k x k table and P_e is the probability of chance-expected agreement calculated by using the appropriate marginal totals of the k x k table.

[1] Taken from www.john-uebersax.com, accessed on 28 February 2016.

In the literature there are various guidelines for the interpretation of the kappa statistic. Dawson, for example, provides the following guidelines in [4]:

Kappa Statistic	Strength of Agreement
$\leq$0.00	No agreement
0.01-0.20	Poor agreement
0.21-0.40	Slight agreement
0.41-0.60	Fair agreement
0.61-0.80	Good agreement
0.81-0.92	Very good agreement
0.93-1.00	Excellent agreement

Table 1: Kappa Guidelines

A similar table in [10] contains fewer categories. Kappa treats all disagreements equally and the magnitude of the kappa statistic is dependent on how the categories were defined [3, 12]. The value of the kappa statistic will be higher when there are fewer categories, thus influencing the conclusion made about the degree of agreement of the situation being evaluated [12].

There are both advantages and disadvantages when using the kappa statistic. The advantages are that it is relatively easy to calculate and it is suitable for measuring agreement beyond chance. One disadvantage is that it is not often comparable among studies, because it depends on the distribution of the phenomenon being studied[2]. In Cohen's case the fact that kappa is limited to only two raters can be considered as a disadvantage since most studies include more than two raters. Another inconvenience is the restriction of identical categories that must be used by raters when rating subjects [6].

In 1968 Cohen developed a weighted kappa statistic (κ_w) which is applicable in situations where ordinal data are used. When some categories are considered more important than others, as could be argued for ordinal data, weights are assigned to reflect the importance among the rated conditions [3, 11]. Hence weighted kappa is a chance-corrected proportion of agreement.

The formula to calculate weighted kappa can be derived in the following way [3] :

Starting from the formula for kappa, as defined in Equation 1 above,

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

we define a weighted proportion of observed agreement, p'_o and a weighted proportion of chance-expected agreement, p'_e as follows:

$$p'_o = \frac{\sum w_{ij} P_{oij}}{w_{max}}$$

and

$$p'_e = \frac{\sum w_{ij} P_{eij}}{w_{max}}$$

where w_{ij} are cell weights that reflect agreement, w_{max} is a maximum weight that represents complete agreement, P_{oij} is the proportion of the joint judgements observed in cell ij and P_{eij} is the proportion in cell ij expected by chance.

There are two types of agreement weights that can be used to calculate weighted kappa, namely Cicchetti-Allison (CA) weights,

[2]Taken from www.john-uebersax.com, accessed on 28 February 2016.

$$w_{ij} = 1 - \frac{|C_i - C_j|}{C_k - C_1}$$

and Fleiss-Cohen (FC) weights,

$$w_{ij} = 1 - \frac{(C_i - C_j)^2}{(C_k - C_1)^2}.$$

CA and FC weights are defined for two categories, i and j, where C_i is the score attached to category i, C_j is the score attached to category j, C_k is the score attached to the last category and C_1 is the score attached to the first category. C_i, C_j, C_k and C_1 are table scores, that represent either the numeric value of the category labels or the category numbers if the categories are of character type.[8]

The only difference between the CA and the FC weights is that the CA weights use the absolute value of the difference between the scores of the two categories, i and j, and the FC weights use the squared value of the difference between the scores of the two categories, i and j.

Replacing the unweighted proportions in the basic formula for kappa, Equation 1, with the above proportions of weighted agreement yields

$$\kappa_w = \frac{p'_o - p'_e}{1 - p'_e} \tag{2}$$

The agreement weight, w_{ij}, is a positive weight determined by a panel of experts or the investigator's own judgement. This weight should be determined before the collection of data and should yield a ratio scale.

An example of such a ratio weight is to assign a maximum weight of one for perfect agreement on the diagonal values of a 6 x 6 table and a minimum weight of 0 representing no agreement. [3]

Fleiss addressed the restriction of two raters in [6] by improving on Cohen's unweighted kappa statistic to allow for the measurement of chance-corrected agreement among any constant number of raters. Fleiss' kappa statistic is an extension of Scott's π-statistic for two raters and not, as is generally believed, a generalisation of Cohen's kappa statistic. The difference between Cohen's kappa and Scott's π is in the way P_e is calculated: Cohen's kappa calculates the expected frequency as the sum of the products of the row and column categories while Scott's π measures the likelihood of agreement by chance, i.e. the sum of the squared averages of the column and row totals [8]. For this Fleiss kappa, the constant number of raters need not necessarily be the same group of raters for each subject, but may be randomly chosen from a group of raters.[6]

The formula that Fleiss developed is :

$$\kappa_f = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e}$$

where $\bar{P}$ is the overall extent of agreement over all the subjects and $\bar{P}_e$ is the mean proportion of chance-expected agreement.[6]

The overall extent of agreement, $\bar{P}$, is calculated with the following formula:

$$\bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i$$

and the mean proportion of chance-expected agreement, $\bar{P}_e$, is calculated as follows:

$$\bar{P}_e = \sum_{j=1}^{k} p_j^2$$

p_j is the proportion of all assignments to the jth category and is calculated with:

$$p_j = \frac{1}{nN} \sum_{i=1}^{N} n_{ij}$$

where N is the total number of subjects, n is the number of ratings per subject and n_{ij} is the number of raters who assigned the ith subject to the jth category.

A number of different indices to quantify interrater agreement on binary data exists, as discussed by Fleiss in [7]; for example three indices developed by Goodman and Kruskal, two indices developed by Rogot and Goldberg, an index by Armitage et al. as well as indices developed by Cohen and Fleiss. Among these authors there was no consensus that chance-expected agreement should be incorporated into the assessment of interrater agreement. The value expected by chance of the majority of these indices yield kappa as the chance-corrected method of agreement.

In 1977 Landis and Koch realised that the rater himself could also be a source of measurement error that could cause interobserver bias. The interested reader is referred to [7].

3 Application

3.1 Evaluating two raters

In this section a practical dermatology example, from SAS/STAT® 13.2 User's Guide[3] in *The FREQ Procedure Agreement Study* example, will be discussed to illustrate the use of Cohen's kappa and weighted kappa statistics.

3.1.1 Problem setting for two raters

Medical researchers want to evaluate a new treatment for a skin condition. Two dermatologists (raters) examined the same group of 88 patients who used this new treatment and evaluated their skin condition by classifying the patients into one of four categories.

The results are shown in Table 2:

Dermatologist 1	Dermatologist 2				
	Clear	Marginal	Poor	Terrible	Total
Clear	13	6	2	0	21
Marginal	5	12	4	2	23
Poor	2	12	10	5	29
Terrible	0	1	4	10	15
Total	20	31	20	17	88

Table 2: Four categories of skin conditions

To test the effect of fewer categories on the values of kappa and weighted kappa, the 'poor' and 'terrible' categories of Table 2 were merged into one category labelled 'bad', resulting in a table with three categories (see Table 3).

[3]The data analysis for this essay was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA.

Dermatologist 1	Dermatologist 2			
	Clear	Marginal	Bad	Total
Clear	13	6	2	21
Marginal	5	12	6	23
Bad	2	13	29	44
Total	20	31	37	88

Table 3: Three categories of skin conditions

The 'clear' and 'marginal' categories were then merged into one category labelled 'normal', in order to have a table with two categories (see Table 4).

Dermatologist 1	Dermatologist 2		
	Normal	Bad	Total
Normal	36	8	44
Bad	15	29	44
Total	51	37	88

Table 4: Two categories of skin conditions

3.1.2 Results

Interrater measures like kappa and weighted kappa were calculated, using SAS software, for each of the three tables above (Tables 2 to 4). The weighted kappa statistics were calculated using the Fleiss-Cohen agreement weights from Table 7. The code of the SAS programme appears in the Appendix.

The interrater measures, based on Tables 2 to 4, are summarised in Table 5:

Statistic	Number of categories		
	Four	Three	Two
Kappa	0.3449	0.3996	0.4773
Weighted kappa	0.6607	0.3113	0.4773

Table 5: Cohen's kappa and weighted kappa values

The two different ways to calculate the agreement weights w_{ij}, used in the formula to calculate weighted kappa for interrater agreement in Tables 2 and 3, were also investigated and the results are shown in Tables 6 and 7:

Dermatologist 1	Dermatologist 2			
	Clear	Marginal	Poor	Terrible
Clear	1	0.6667	0.3333	0
Marginal	0.6667	1	0.6667	0.3333
Poor	0.3333	0.6667	1	0.6667
Terrible	0	0.3333	0.6667	1

Table 6: Cicchetti-Allison agreement weights for the four categories of skin conditions in Table 2

Dermatologist 1	Dermatologist 2			
	Clear	Marginal	Poor	Terrible
Clear	1	0.8889	0.5556	0
Marginal	0.8889	1	0.8889	0.5556
Poor	0.5556	0.8889	1	0.8889
Terrible	0	0.5556	0.8889	1

Table 7: Fleiss-Cohen agreement weights for the four categories of skin conditions in Table 2

3.1.3 Interpretation and discussion

From Table 2 the exact agreement can be calculated as $\frac{45}{88}$, i.e. 0.5114, by using the counts on the diagonal. The exact agreement value was calculated without taking the element of chance into account, whereas Cohen's kappa statistic represents agreement beyond chance and is equal to 0.3449 for four categories of skin conditions (From Table 5). This value of Cohen's kappa would lead to the conclusion that there is slight agreement (see Table 1) between the ratings of the two dermatologists.

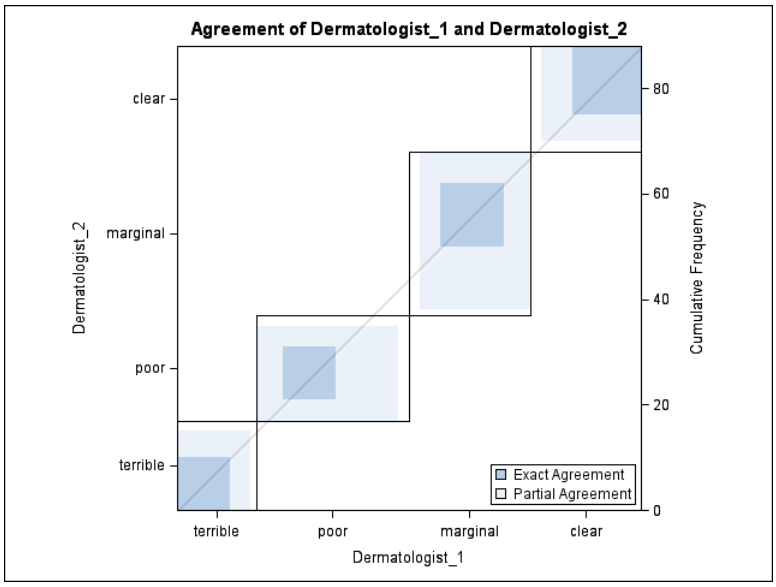


Figure 1: Agreement plot for four categories of skin conditions

The agreement between the two dermatologists is shown visually in Figure 1 (from the SAS output). The dark blue blocks represent the exact agreement values as seen on the diagonal of Table 2 and the light blue blocks represent the off-diagonal partial agreement values in Table 2.

From the results in Table 5 it can be seen that the value of the kappa statistic increases when reducing the number of categories. For four categories of skin conditions the value of Cohen's kappa is 0.3449. When the categories are combined into three categories of skin conditions, it can be seen that the value of Cohen's kappa is 0.3996, indicating an increase in the kappa value and slight agreement between the ratings of the two

dermatologists. In the last case of two categories of skin conditions, it can be seen that the value of Cohen's kappa is 0.4773, which is again higher than the two previous cases, indicating fair agreement (see Table 1) between the ratings of the two dermatologists. Using two categories of skin conditions instead of three or four categories, resulted in a higher value of the kappa statistic and thus a more favourable conclusion of a higher degree of agreement between the ratings of the two dermatologists, as mentioned in Section 2.

From Table 5 it is also clear that the value of Cohen's weighted kappa statistic decreases when reducing the number of categories. Weights are assigned to the categories according to the importance of each category. For four categories of skin conditions the value of weighted kappa is 0.6607 and for three categories of skin conditions the value of weighted kappa is 0.3113. For two categories of skin conditions the value of kappa and weighted kappa are the same and equal to 0.4773. Researchers should thus be cautious when deciding on the amount of categories for the interpretation of results, due to the fact that it can influence the conclusion made about the degree of agreement present.

The Cicchetti-Allison and Fleiss-Cohen agreement weights are shown in Tables 6 and 7. It is clear from Table 7 that the Fleiss-Cohen weights are bigger than the Cicchetti-Allison weights in Table 6, thus the decision of agreement weights will also influence the value of the weighted kappa statistic.

3.2 Evaluating more than two raters

In this section a practical example, based on a portion of data used in a doctoral thesis that investigated emotionally triggered involuntary violent behaviour (ETIVB), will be discussed to illustrate the use of Fleiss' kappa.

3.2.1 Problem setting for more than two raters

An ETIVB-instrument, comprising several sets of criteria as an assessment tool, was developed by Joubert in [9]. In his thesis Joubert investigated ETIVB with this instrument. Part of the study also consisted of evaluating the instrument's reliability. In his study 25 psychiatrists (raters) used the ETIVB list of criteria to determine whether a patient's behaviour can be classified as ETIVB. Several patients were evaluated in the study; in this section I will be considering only two of these patients (patient A and patient B). Furthermore, only one set of the criteria (F1 to H1) was considered in order to investigate Fleiss' kappa and the agreement of the 25 psychiatrists with regards to the classification of the criteria into one of the four categories: *met*, *not met*, *uncertain* or *not applicable*.

3.2.2 Results

The classification of the criteria, by the 25 raters, into one of the four categories, is as follows for each of the two patients:

Criteria	Categories for classification of criteria			
	Met	Not Met	Uncertain	Not Applicable
F1	25	0	0	0
F2	17	6	1	1
F3	21	4	0	0
F4	24	0	1	0
F5	22	1	2	0
G1	2	22	0	1
H1	11	14	0	0
Total	122	47	4	2
	175			

Table 8: Rater classification for patient A

Criteria	Categories for classification of criteria			
	Met	Not Met	Uncertain	Not Applicable
F1	0	22	0	3
F2	0	22	0	3
F3	0	22	0	3
F4	0	21	2	2
F5	0	21	1	3
G1	1	23	0	1
H1	6	17	1	1
Total	7	148	4	16
	175			

Table 9: Rater classification for patient B

Tables 8 and 9 were used to calculate Fleiss' kappa for both patients, using Fleiss' formula in Section 2, and summarised in Table 10:

Statistic	Patient A	Patient B
Kappa	0.4096	0.0122

Table 10: Fleiss' kappa value for each patient

3.2.3 Interpretation and discussion

The results from the SAS/IML code, refer to the Appendix, are presented in Table 10. The results in Table 10 indicate that for patient A there was fair agreement among the 25 raters when they classified the criteria for patient A, since the value of Fleiss' kappa is 0.4096. For patient B the value of Fleiss' kappa is 0.0122, which means there was poor agreement among the 25 raters when they classified the criteria for patient B. This result for patient B is contradicting the results from Table 9, which show that 85% (148/175) of all the ratings "agreed" that the criteria were not met and thus also that the majority of raters agreed that the criteria were not met. A better value for the measure of agreement than that of the result of Fleiss' kappa (0.0122) was thus expected. As seen from the results of patient B, Fleiss' kappa can be flawed. This shortcoming is not unique to this study and dataset, Cicchetti and Feinstein have also found this shortcoming and discussed it in [5] and [1]. When evaluating the results of any study it is therefore very important to evaluate the value of the statistic, together with the data itself, in order to detect any shortcomings as seen in the case of patient B.

4 Conclusion

This essay sought to provide a basic overview of Cohen's weighted and unweighted kappa as well as Fleiss' kappa, which are all measures of interrater agreement. The background, use, advantages and disadvantages of these statistics were discussed. Cohen's weighted and unweighted kappa statistics are limited to the case of only two raters and where the same two raters rate the same group of subjects [6]. Fleiss addressed the restriction of two raters in [6] by improving on Cohen's unweighted kappa statistic to allow for the measurement of chance-corrected agreement among any constant number of raters.

As seen in the essay, both of these statistics have shortcomings that need to be kept in mind when performing a study that involves measuring agreement. In the case of Cohen's kappa and weighted kappa the number of categories should be carefully considered since they can influence the value of the kappa statistic and thus the conclusion made about the degree of agreement between the raters. In the case of Fleiss' kappa it is possible to obtain a low value for the kappa statistic even though it is clear from the data that there is a high level of agreement among the raters.

Various other measures for evaluating not only interrater but also intrarater agreement are available. This essay only focussed on three measures of agreement, but as mentioned in Section 1 and 2, other measures that can be used are the McNemar chi-square statistic or Scott's π statistic.

The three statistics discussed in this essay are therefore not the only measures of agreement but the most common ones used to evaluate agreement, especially in medical practice. It would not be wise to compare these statistics among different studies as the number of categories for Cohen's kappa or raters for Fleiss' kappa may differ among the studies. In any study it is very important to consider the data together with the results when interpreting them and coming to a conclusion. Any discrepancies that might be found should be stated.

References

[1] Domenic V Cicchetti and Alvan R Feinstein. High agreement but low kappa: II. Resolving the paradoxes. *Journal of Clinical Epidemiology*, 43(6):551–558, 1990.

[2] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychosocial Measurement*, 20(1):37–46, 1960.

[3] Jacob Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220, 1968.

[4] Beth Dawson and Robert G Trapp. *Basic & Clinical Biostatistics*. Lange Medical Books; McGraw-Hill, fourth edition, 2004.

[5] Alvan R Feinstein and Domenic V Cicchetti. High agreement but low kappa: I. The problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549, 1990.

[6] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[7] Joseph L Fleiss. Measuring agreement between two judges on the presence or absence of a trait. *Biometrics*, 31(3):651–659, 1975.

[8] Kilem Gwet. Computing inter-rater reliability with the SAS system. *Statistical Methods For Inter-Rater Reliability Assessment*, 3:1–16, 2002.

[9] Pierre Mauritz Joubert. *Emotionally Triggered Involuntary Violent Behaviour not Attributed to a Mental Disorder: Conceptual Criteria and Their Reliability*. PhD thesis, University of Pretoria, 2014.

[10] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

[11] Lawrence Lin, AS Hedayat, and Wenting Wu. *Statistical Tools for Measuring Agreement*. Springer Science & Business Media, 2012.

[12] Malcolm Maclure and Walter C Willett. Misinterpretation and misuse of the kappa statistic. *American Journal of Epidemiology*, 126(2):161–169, 1987.

Appendix

See the SAS code used for the examples, discussed in the practical application section, below.

Evaluating two raters

```
/****** Calculating kappa and weighted kappa for more than two categories *******/
/*********************** Four Categories (Table 2) ****************************/
data Ex1;
input Dermatologist_1$ Dermatologist_2$ Count @@;
datalines;
terrible     terrible 10
terrible     poor 4
terrible     marginal 1
terrible     clear 0
poor         terrible 5
poor         poor 10
poor         marginal 12
poor         clear 2
marginal     terrible 2
marginal     poor 4
marginal     marginal 12
marginal     clear 5
clear        terrible 0
clear        poor 2
clear        marginal 6
clear        clear 13
;
run;

ods graphics on;
proc freq data=Ex1 order=data;
   tables dermatologist_1*dermatologist_2 /Agree(WT=FC) printkwt plots=agreeplot;
   weight Count;
run;
ods graphics off;

*To calculate weighted kappa using CA weights the (WT=FC) option in the TABLES
statement was left out;
*The plots option generates the agreement plot and the printkwt option prints
the weights used to calculate weighted kappa;

/************************** Three Categories (Table 3) ************************/
data Ex2;
input Dermatologist_1$ Dermatologist_2$ Count @@;
datalines;
clear    clear    13
clear    marginal 6
clear    bad      2
marginal clear    5
marginal marginal 12
marginal bad      6
bad      clear    2
```

```
bad      marginal 13
bad      bad      29
;
run;

proc freq data=Ex2;
tables dermatologist_1*dermatologist_2 / Agree score=table printkwt;
weight Count;
run;

/************************* Two Categories (Table 4) ***************************/
data Ex3;
input Dermatologist_1$ Dermatologist_2$ Count @@;
datalines;
normal normal 36
normal bad    8
bad    normal 15
bad    bad    29
;
run;

proc freq data=Ex3;
tables dermatologist_1*dermatologist_2 / Agree(WT=FC) score=table printkwt;
weight Count;
run;
```

Evaluating more than two raters

```
/******************************** Fleiss kappa ********************************/
PROC IMPORT OUT= WORK.fleiss
            DATAFILE= "C:\Users\Cairstine\Documents\University\2016\Re
searchReport\PracApplication\Joubert_Data\CdK Joubert_1_Edited_7 Dec 201
2_1.csv"
            DBMS=CSV REPLACE;
            GETNAMES=YES;
            DATAROW=2;
RUN;

proc format;
value aa 1= Met
         2= Not_met
         3= Uncertain
         4= Not_applicable;
run;
proc print data=fleiss;
format F1 F2 F3 F4 F5 G1 H1 aa.;
run;

proc freq data=fleiss;
tables patient*(F1 F2 F3 F4 F5 G1 H1)/nopercent norow nocol;
format F1 F2 F3 F4 F5 G1 H1 aa.;
```

```
run;

proc iml;
n=25;
NN=7;

*Frequency Table --> Patient A;
A={25 . . .,17 6 1 1,21 4 . .,24 . 1 .,22 1 2 .,2 22 . 1,11 14 . .};
Total_A=A[+,];
Sum_A=Total_A[,+];
Print 'Classification for Patient A',, A, Total_A, Sum_A;

A2=A##2;
Sum_A2=A2[,+];
print A2, Sum_A2;

Pi_A=(1/(n*(n-1)))*(Sum_A2-J(nrow(Sum_A2),1,n));
print Pi_A;
P_bar_A=(1/NN)*(Pi_A[+,]);
print P_bar_A;
Pj_A=Total_A/Sum_A;
Pj_A2=Pj_A##2;
print Pj_A Pj_A2;
Pbar_e_A=Pj_A2[,+];
print Pbar_e_A;

Kappa_A=(P_bar_A-Pbar_e_A)/(1-Pbar_e_A);
print "Fleiss's kappa for patient A", Kappa_A;

*Frequency Table --> Patient B;
B={. 22 . 3, . 22  . 3, . 22 . 3, . 21 2 2, . 21 1 3, 1 23 . 1, 6 17 1 1};
Total_B=B[+,];
Sum_B=Total_B[,+];
Print 'Classification for Patient B',, B, Total_B, Sum_B;

B2=B##2;
Sum_B2=B2[,+];
print B2, Sum_B2;

Pi_B=(1/(n*(n-1)))*(Sum_B2-J(nrow(Sum_B2),1,n));
print Pi_B;
P_bar_B=(1/NN)*(Pi_B[+,]);
print P_bar_B;
Pj_B=Total_B/Sum_B;
Pj_B2=Pj_B##2;
print Pj_B Pj_B2;
Pbar_e_B=Pj_B2[,+];
print Pbar_e_B;

Kappa_B=(P_bar_B-Pbar_e_B)/(1-Pbar_e_B);
print "Fleiss's kappa for patient B", Kappa_B;

quit;
```

Methods for semi-supervised text classification

Gabriël de Wet 11304643

STK795 Research Report

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

Supervisor: Dr A de Waal, Co-supervisor: J Mazarura

Department of Statistics, University of Pretoria

2 November 2016

Abstract

In the real world, the effectiveness of text classification models -like support vector machines and logistic regression- are highly dependent on the quality and structure of the training data set. In practice unlabelled data are in abundance, as opposed to labelled data, which are the usual standard when it comes to building a classifier. Semi-supervised methods for classification do exist, although some of them do not perform as well in practice as the theory behind them would suggest.

This paper evaluates specific methods for classification of unlabelled text data by determining how effective they are when compared to a standard supervised learning method such as the support vector machine. Two independent, fully labelled data sets will be used in order to evaluate these methods and determine if it is at all possible to apply them in the real world.

Declaration

I, *Gabriël Botha de Wet,* 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.

_____
Gabriël Botha de Wet

_____
Jocelyn Mazarura

_____
Alta de Waal

_____
Date

Acknowledgements

The author would like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a postgraduate bursary.

Contents

1 Introduction **6**

2 Background Theory **8**
2.1 Document classification 9
2.2 Rocchio classifier 10
2.3 Support vector machines 11
2.4 Theoretical implementation 14

3 Application **17**

4 Conclusion **20**

Appendix **23**
Notation 23
Code 24
Results 35

List of Figures

1 Illustrations of Rocchio and SVM 11
2 Accuracy and F1 scores for 20 newsgroups data 19
3 Accuracy and F1 scores for GOP data 20
4 ROC curves for GOP data 21

List of Tables

1 Sample GOP data 17
2 F1 and accuracy scores for supervised models 18
3 Number of assumed negative observations before and after implementation of semi-supervised methods. 19
4 Results: 20 Newsgroups data 35
5 Results: GOP data (positive tweets as labelled) 36
6 Results: GOP data (negative tweets as labelled) 37

List of Algorithms

1 Rocchio 14
2 Rocchio with clustering 15
3 Train final SVM 16

1 Introduction

Binary text classification using semi-supervised methods will form the main focus of this report. Large amounts of data are available thanks to modern technology but the challenge with the available data is that the data are typically unlabelled; in other words, there is no corresponding label or class on raw data (e.g. stating that the sentiment of a tweet is positive). Usually, when supervised classification is taking place, it means that a human being had to manually label the training data set. This can be tedious and expensive in terms of time and money. Naturally the solution would be to use as little labelled data as possible without compromising on the performance of the model. This is exactly what will be attempted in this report. The experimental application will consist of applying several semi-supervised learning algorithms on a data set containing tweet text along with sentiment labels. The algorithms are applied using varying amounts of labelled data and the results are compared to a logistic regression performed on the same data, as well as two supervised models.

The topic of the tweets to be used in the experimental application will be similar to that of Charalampakis et al. [2], where they classified political tweets in Greece as ironic or not. Their hypothesis was that humorous political tweets could be used to predict election results and their approach was semi-supervised since they did not have a lot of labelled data at their disposal. They used collective learning algorithms (collective classification) in order to take both labelled and unlabelled data into account and in the end, they compared it to previous research they conducted using supervised classification. The same methods will not be implemented in this report. The concept of using semi-supervised techniques is very similar, however this report will classify sentiment in tweets rather than irony.

In order to discuss some of the options available to us in the realms of semi-supervised classification using support vector machines (SVMs), it is important to start at the beginning. In 1995, the concept of SVM was born thanks to Cortes & Vapnik [3]. Conceptually, SVMs non-linearly map input vectors onto a high-dimensional space. In our case, these vectors will represent the text documents (tweet text). When vectors are mapped, a linear decision surface is constructed, onto which a separating hyperplane is mapped as well. In the case of two-way classification, this hyperplane will separate the vectors of the two classes. In order for the hyperplane to be optimal and generalised, it needs to be a linear decision function that maximises the distance between the closest vectors from the different classes. Their algorithm for support vector networks proved to work quite well, even relative to other classical algorithms. This, along with other properties like capacity control and the ease of changing the implemented decision surface make SVMs quite capable as general machine learning models.

Building on what Cortes & Vapnik [3] created, Joachims [6] went a step further in an attempt to improve what was already a very solid and reliable binary-classifier. Joachims introduced Transductive Support Vector Machines (TSVMs) specifically for text classification as a solution to the problem of manually labelling data. An SVM can perform transduction by finding the optimal hyperplane with respect to both labelled and unlabelled data [16]. This makes it an ideal method for semi-supervised classification. The inductive approach (standard SVM) attempts to induce a decision function with a small error onto the entire collection of examples for a particular task, which means it can generalise to independent data sets; while the goal of the transductive approach is simply to classify a set of examples while generating the fewest errors possible, not regarding the decision function at all. TSVMs are especially effective on short text (like tweets), and in some cases, according to Joachims [6], reduces the amount of labelled data required by a factor of 20. TSVMs excel at using the specific statistical properties of text and use the margins of separating hyperplanes to encode prior knowledge into the model. It should be noted that TSVMs cannot be generalised to be used on independent data due to its design.

Acquiring unlabelled data in very large quantities is not very difficult, the issue is how to use it. Thus, determining the value of unlabelled data under certain classification models is vital. Zhang & Oles [17] approached their analysis of unlabelled data from a statistical point of view, given the assumption that the correct model of the underlying statistical distribution is given. They used Fisher information matrices to

evaluate the asymptotic value of the unlabelled data and applied this methodology to active learning as well as passive partially supervised learning. According to their paper, SVMs are not quite suited to passive partially supervised learning, however, that is not the case with active learning, where their experiments showed that SVMs are definitely suitable for active learning. With active learning, an algorithm selects unlabelled observations and requires the user to input the label so that it can train further. With that in mind, Tong & Koller [16] introduced an active learning SVM model in their paper *Support Vector Machine Active Learning with Applications to Text Classification*. They described pool-based active learning as a procedure in which the learner/model has a pool of unlabelled instances available to it, from which it can request labels. Thus, the algorithm they proposed is useful for choosing which instance the model should request next. The hope was that the additional flexibility added by a learner that can actively choose the training data would lessen the thirst for large amounts of labelled data. The next issue was to ensure that the learner would request good labels or queries from the pool. For this, they [16] suggested TSVMs, where the performance of the model is evaluated on the original test set rather than a new, independent one. Tong & Koller[16] used three algorithms and all three of them led to improvements for both SVMs and TSVMs.

In their paper, Li & Lui [10] discussed a special case of semi-supervised text classification, where there exists two sets of documents. Assuming a binary classifier, one document set contains unlabelled data, and the other one contains data which is labelled as only one of the two classes (i.e. semi-supervised due to the lack of a second class). Let the unlabelled document set be U, and the labelled document set be P (positive). Note that while P only contains documents of class P, document set U contains unlabelled documents that may belong to either of the possible classes. Now, given U and P, the objective is to build a model that can classify documents in U as either P or "not P" (let it be N, for negative). The proposed solution by Li & Lui [10] was to use the Rocchio[1] classifier to extract some reliable negative documents (RN) from U and then use an SVM iteratively for building a classifier. Two main methods that were used:

1. **Standard Rocchio:** Is treated U as only negative documents and then used P and U as training sets to train the Rocchio classifier. The classifier was then used to classify U and the negative documents were denoted by RN (reliable negative documents).

2. **Rocchio with clustering:** It was used when the decision boundary was non-linear which might have caused Rocchio to extract some positive documents from U and put them into RN. The point of the clustering was to further purify RN. To summarise, the k-means algorithm [4] was used to cluster RN into k clusters. Then Rocchio was used to build a classifier using each cluster and P. The idea was that the classifier will find positive documents in the clusters and remove them.

These methods derive the final negative sets RN and RN' respectively. An SVM was used to build the final classifier, using P as the positive training set and RN or RN' as the negative training set. Implementing and evaluating these methods used by Li & Lui [10] form the main focus of this paper.

[1] Note: Rocchio method is in essence Nearest Centroid classification using TF-IDF vectors[13].

2 Background Theory

Basic definitions:

Hyperplane: A hyperplane is a flat subspace of dimension $n-1$ in an n-dimensional vector space (e.g. in a 2-dimensional space, a hyperplane is a line)[5].

Margin: In the context of support vector machines, a margin is the distance between a hyperplane and the closest examples / observations [15].

Kernel: A function that can quantify the similarity / relationship between two data points or vectors. Kernels can have multiple uses. An example would be a kernel of the form

$$\kappa\left(\mathbf{x}_i, \mathbf{x'}_i\right) = (1+\sum_{j=1}^{P} x_{ij}x'_{ij})^d$$

which is a polynomial kernel used for enlarging feature space [4]**.**
Confusion matrix: A summary of classification performance with respect to some test data. Matrix contains four cells, *true positive* (TP), *false positives* (FP), *true negatives* (TN) and *false negatives* (FN) [15].

Precision (*Positive Predictive Value*): The proportion of correct positive predictions relative to total positive predictions [15]:

$$Precision = TP/(TP+FP)$$

Sensitivity (*Recall / True Positive Rate*): Proportion of correct positive predictions to the total actual positive values [15]:

$$Recall = TP/(TP+FN)$$

Specificity: (*True Negative Rate*): Proportion of correct negative predictions to the total actual negative values [15]:

$$Specificity = TN/(TN+FP)$$

False Positive Rate: (1-Specificity): Proportion of incorrect negative predictions to the total actual negative values [15]:

$$FPR = 1 - TNR = 1 - Specificity$$

F1-score: Used in evaluating the prediction accuracy in binary classification. Calculated as the harmonic mean between recall and precision (also tends toward the lower of the two values) [15].

$$F1 = \frac{2TP}{2TP+FP+FN}$$

AUC: *Area under curve.* Empirical evaluation of classifier performance. AUC is the area under an ROC curve [15].

ROC curves: A graph plotting the trade-off between *Sensitivity (True Positive Rate, TPR)* and the *False Positive Rate* (FPR) for different threshold values for a model's decision function [15].

Cross-validation: Cross-validation is a resampling method used to evaluate a model's performance by estimating the test error associated with the model [5]. K-fold cross-validation partitions data into k samples ($S_1 \rightarrow S_k$), called folds. Then the learning method is applied k times ($i = 1$ to k), using $S_i$as the test set and the union of all other folds as the training set [15].

2.1 Document classification

In order to classify text, it should be possible to mathematically compare two documents. To do this, one can use kernels. In this report, the kernel that will be used to compare documents is the cosine similarity. For comparing two documents, $\mathbf{x_i}$ and $\mathbf{x_{i'}}$, the *cosine similarity*, when using a bag-of-words document representation, can be written as the kernel function

$$\kappa\left(\mathbf{x}_i, \mathbf{x}_{i'}\right) = \frac{\mathbf{x}_i^T \cdot \mathbf{x}_{i'}}{||\mathbf{x}_i||\,||\mathbf{x}_i||}$$

where x_{ij} is the number of times the word j appears in document i. The kernel function above measures the cosine of the angle between the two documents, where the documents are represented by vectors. Note that $0 \leq \kappa\left(\mathbf{x}_i, \mathbf{x}_{i'}\right) \leq 1$ since the document $\mathbf{x}_i$is a count vector. From this, it is clear that if the kernel is zero, the vectors are orthogonal, and have no words in common [12].

This method is not ideal, because:

1. Non-discriminative words such as "the" and "a" may increase similarity between documents, even if documents aren't similar at all (Also known as *stop words*).
2. A document's similarity may be boosted (artificially) due to the same word occurring multiple times.

The above issues can be solved by replacing the count vectors with *TF-IDF* (term frequency - inverse document frequency) vectors [12].

Term frequency,

$$tf(x_{ij}) \equiv \log\left(1 + x_{ij}\right)$$

reduces the impact of the second point above, while inverse document frequency,

$$idf(j) \equiv \log \frac{N}{1 + \sum_{i=1}^{N} II\left(x_{ij} > 0\right)}$$

where N = number of documents
and the denominator counts documents containing j.

Finally, we have:

$$tfidf(\mathbf{x}_i) \equiv \left[tf(x_{ij}) \times idf(j)\right]_{j=1}^{V}$$

New kernel:

$$\kappa\left(\mathbf{x}_i, \mathbf{x}_{i'}\right) = \frac{\Phi(\mathbf{x}_i)^T \cdot \Phi(\mathbf{x}_i)}{||\Phi(\mathbf{x}_i)||\,||\Phi(\mathbf{x}_{i'})||}$$

where $\Phi(\mathbf{x}) = tfidf(\mathbf{x})$ [11].

TF-IDF scores assigns a weight to term j in document i given by the formula mentioned above. This weight is:

1. Highest when j occurs frequently in few documents (i.e. unique and discriminating).
2. Lower when j appears less frequently in a document or appears across many documents.
3. Lowest when j appears in all documents.

Simply put, every document is a vector with a weighted value corresponding to each term in the dictionary. Naturally, when a term does not occur in a document, the weight is zero [11].

This representation of document vectors, in one vector space, is known as the vector space model. Following from the fact that document vectors are length-normalized, there exists a direct relationship between cosine similarity and euclidean distance calculations. In fact, it hardly matters which of these metrics are used to determine the relation between two documents in a vector space [11]. The vector space model is very important going forward since Rocchio and SVM are both vector space classifiers.

2.2 Rocchio classifier

The Rocchio classifier is based on the Rocchio algorithm for relevance feedback. In the case of the classifier, instead of having to classify something as relevant or not, it can be used to classify something as positive or not-positive. It is extensively used for text classification (with TFIDF vectors used as features) [7]. Essentially it computes a prototype vector $\mathbf{p}_c$ for each class c, where the prototype vector is the average of the document vectors in either class. In order to classify a new document, it calculates the distance between the unlabelled document and the prototype vectors, and classifies according to the closest prototype vector. These prototype vectors serve the same purpose as centroids in nearest centroid classification. The average or median value within a cluster is usually used as a centroid [11]. It is assumed that the vectors of documents of the same class form a cluster.

Figure 1a from [11] illustrates what a Rocchio classification would look like. In this illustration, there are 3 classes with the feature vectors plotted onto a 2-dimensional space. The lines separating the vectors from the different classes are called decision boundaries. A simple example of classifying a new observation would be to classify it according to which boundary it falls into (e.g. the star would be classified as China). Thus it is clear that, in order to classify a new observation, class boundaries need to be constructed. Rocchio classification uses the centroids of each class to define the decision boundaries. Let c be the class, then it's centroid is calculated as either the average across the vectors or the median, as mentioned above.

In Figure 1a, the centroids are denoted by bold dots. The set of observations that are equal distance from two centroids form the class boundaries. This can be seen in the figure, represented by $|a_1| = |a_2|$ and so on. These lines can also be referred to as separating hyperplanes.

Using Rocchio for classification is quite simple. Initially, the normalised document vectors of both classes,$\mathbf{x}_i$ & $\mathbf{x}_j$, (in the case of binary classification) are summed up. Then, the prototype vector is computed as:

$$\mathbf{w} = \frac{1}{|i : y_i = +1|} \sum_{i:y_i=+1} \mathbf{x}_i - \beta \frac{1}{j : y_j = -1} \sum_{j:y_j=-1} \mathbf{x}_j$$

where β adjusts relative impact of training examples from both classes.

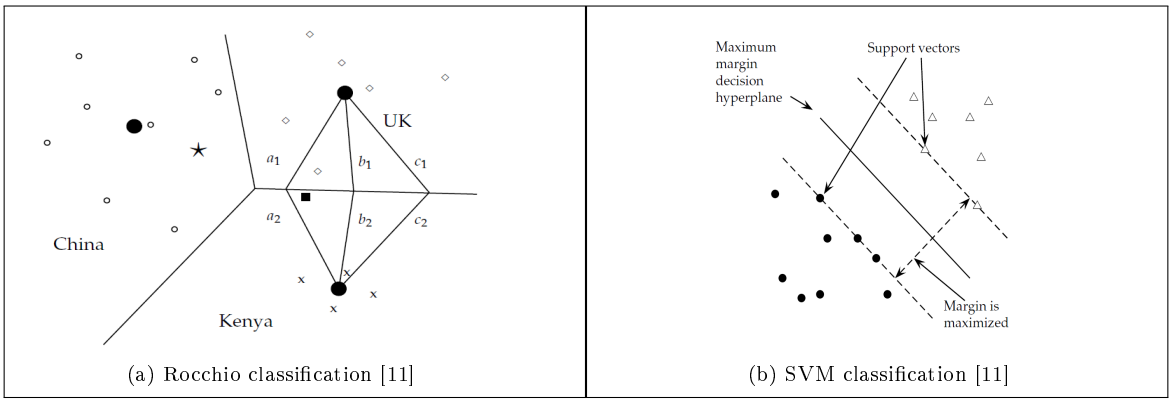
A β-value of 0.25 is recommended. As required by Rocchio, all elements $w_i < 0$ in vector $\mathbf{w}$ are set to zero. Using this classification rule to classify a new document $\mathbf{x}$, one computes the cosine between vector $\mathbf{w}$ and vector $\mathbf{x}$ [7]. If the vectors are similar, $\mathbf{x}$ is classified as "relevant" (whichever one of the classes it may be), and if not, classified as the other class.

The Rocchio classifier that will be used in the semi-supervised classification algorithms in this paper is very similar to the one mentioned above. It will be discussed in the next section.

The motivation for using Rocchio given by Li & Lui [10] is that the unlabelled document set U will exhibit the following characteristics:

- The ratio of positive documents to negative documents in the unlabelled set U, is small, which means it doesn't affect the centroid $\mathbf{c}$ all too much.
- The negative documents in the unlabelled set U are of a wide range of topics, which means they cover a significant area in the vector space.
- Topics covered by the positive documents in the positively labelled set P are not as diverse, thus covering a much smaller area in the vector space.

Given the above, assume that there exists a decision boundary S that perfectly separates the positive and negative documents. Take note that from here on out the centroid will be replaced by a prototype vector, which serves the same purpose (as mentioned above) [11]. This means that the prototype vector for the positive class $\mathbf{c}^+$ will be much closer to S relative to the prototype vector for the negative class $\mathbf{c}^-$, due to vector summation. After applying the similarity calculation, a few negative documents from U will be classified as positive because they are closer to the prototype vector for the positive class. Thus, Rocchio extracts negative documents with high precision, and positive documents with low precision (but high recall). The Rocchio algorithm used by Li & Lui [10] is explained in section **2.4.1**, under Algorithm 1.



(a) Rocchio classification [11]

(b) SVM classification [11]

Figure 1: Illustrations of Rocchio and SVM

2.3 Support vector machines

SVMs are a group of algorithms that can be implemented in classification and regression problems. On a basic binary classification level, SVMs map a hyperplane that separates data from the two classes while maximising the margin between the nearest observations. SVMs have very generalisation capabilities, which means once the classifier is trained it can easily be used on an independent data set, and they also support specialised optimisation methods which means that they are able to be trained efficiently on large quantities of data [15].

Maximum margin hyperplane Support vector machines are a generalisation of the maximum margin classifier. The maximum margin classifier uses a separating hyperplane (a hyperplane that divides the data between their respective classes) to form a decision function over the observed data. Since the data are linearly separable, an infinite amount of separating hyperplanes exist. So in the case of the maximum margin

classifier, we use the maximum margin hyperplane, also known as the optimal separating hyperplane. This is the hyperplane that is furthest from all the data points from both classes. The distances between the hyperplane and all the data points are calculated, and the smallest observed distance forms the margin. So, as the name suggests, the maximum margin classifier attempts to maximise the margin. Put differently, the optimal hyperplane is the one for which the margin is maximised. In Figure 1b [11] the maximum margin hyperplane is found to form the decision function of the maximum margin classifier. The data points that lie on the margin are referred to as support vectors. Note that these support vectors are the only observations that affect the maximum margin hyperplane. In order to compute the maximum margin hyperplane, one has to solve a restricted optimisation problem. This optimisation problem involves maximising the margin subject to some constraints[10]. One drawback of the maximum margin classifier is that it is extremely sensitive to a change in the data points because no observations are allowed to fall within the margin. Even a single data point could throw it off. To solve this problem, consider a classifier that does not perfectly separate the data.

Support vector classification Support vector classifiers do not perfectly separate the observations as the maximum margin classifier did. This makes it more robust and better at classifying most of the training observations. The support vector classifier works largely the same as the abovementioned classifier. It also generates a hyperplane in order to separate all the data, however, it is not restricted to classifying every single training observation correctly. Support vector classifiers follow the same constrained optimisation problem as the maximum margin classifier did, but for a small difference. There is an added tuning parameter which allows for slack variables to be added to the constraints. The role of the slack variables is to allow for some of the observations to be on the inside of the margin and on the wrong side of the hyperplane, while the role of the tuning parameter is to "tune" amount and the magnitude of the violations caused by the slack variables[5].

Support vector machines The main difference between SVMs and the two classifiers mentioned above (it can be argued that all three of these classifiers together form SVMs), is that both the maximum margin classifier and the support vector classifier relies on the observations from the two classes to be linearly separable. Linearly separable data is not that common in practice, though. SVMs are an extension of the abovementioned methods, generalising them and applying them to data where the data points between the classes are not linearly separable (as of yet). SVM's use kernels to add dimensions to the current feature space in an attempt to be able to separate the observations from the two classes in a non-linear fashion. The kernel approach is best for this because of its efficiency in its computations [5]. Thus, SVM's are maximum margin classifiers with added flexibility for slack variables (by softening what is meant with *separating*) and the ability to separate non-linearly separable data by making use of kernels [4].

Mathematically Consider the training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ where $\mathbf{x}_i \in \mathbb{R}^p$ is the feature vector for the i^{th} example in p dimensions, with corresponding labels $y_i \in \{1, -1\}$. Assume for now that the observations are linearly separable, and let $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$ be the equation for the separating hyperplane [4]. Then:

$$\begin{aligned} f(\mathbf{x}) &> 0 \quad \text{for points on one side of the hyperplane} \quad (y_i = +1) \\ f(\mathbf{x}) &< 0 \quad \text{for points on the other side of the hyperplane} \quad (y_i = -1) \\ f(\mathbf{x}) &= 0 \quad \text{for points on the hyperplane} \end{aligned}$$

Thus, $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$ is the separating hyperplane. As before, the objective is to find the optimal separating hyperplane, in other words the one that maximises the margin. This is a constrained optimisation problem. Equation (1) and (2) below represents the relevant constraints:

$$\max_{\beta_0,\beta_1,\ldots,\beta_p} M \text{ subject to:}$$

$$\sum_{j=1}^{p} \beta_j^2 = 1 \tag{1}$$

$$y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip}) \geq M \tag{2}$$

with

$$M \quad = \quad \text{distance of } i^{th} \text{ point from current hyperplane (defined by } \beta_j\text{'s)}$$

This can be solved efficiently using many optimisation algorithms. The most popular one is the Sequential Minimal Optimisation (SMO) algorithm by Platt [14], and is implemented in popular SVM software packages such as *libsvm* [1]. A software package that uses a different optimisation algorithm is *SVMlight* [8]. The point of the optimisation problem above is to find the parameters for the hyperplane that will maximise M (such that each observation is at least M units from the hyperplane).

The optimisation problem above only applies to data with perfectly and linearly separable features. In order to use SVMs on data that do not exhibit perfectly separable features, the optimisation problem is modified such that:

$$\max_{\beta_0,\beta_1,\ldots,\beta_p} M \text{ subject to:}$$

$$\sum_{j=1}^{p} \beta_j^2 = 1 \tag{3}$$

$$y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip}) \geq M(1 - \epsilon_i) \tag{4}$$

$$\epsilon_i \geq 0, \; \sum_{i=1}^{n} \epsilon_i \leq C$$

with

$$\begin{aligned} \epsilon_i &= \text{how much an observation is allowed beyond the margin} \\ (1 - \epsilon_i) &= \text{discount factor for allowing some slack} \\ C &= \text{total overlap} \end{aligned}$$

The above problem maximises the margin subject to a modified constraint in equation (4), and can also be solved using the aforementioned software packages. Note that a higher value for C will lead to a more robust hyperplane.

The two cases discussed both require features that are linearly separable. Since data are rarely separable in the real world, adjusting the aforementioned methods is necessary. In order to build a classifier that can effectively predict data points that aren't linearly separable, a non-linear decision boundary needs to be formulated. This can be done by using kernels to expand the feature space (polynomial transformations can also be used, but lack the efficiency of kernels as the degree of the polynomial increases) [4]. The modified

decision function:

$$f(x) = \beta_0 + \sum_{i \in S} \hat{\alpha}_i K(\mathbf{x}, \mathbf{x_i})$$

with

$$\begin{aligned} \hat{\alpha}_i &= \text{estimated parameters} \\ K(\mathbf{x}, \mathbf{x_i}) &= \text{the kernel function used to transform the feature space} \end{aligned}$$

Linear (which is just a standard support vector classifier), polynomial and radial kernels are all examples of kernel functions that are used in practice [4]. Choosing a kernel function wholly depends on the structure and separation of the features.

2.4 Theoretical implementation

Constructing negative document set

The first step to implementing the methods by Li & Lui [10], is to construct a reliable negative data set RN. As mentioned before, the first algorithm uses only the Rocchio classifier on positively labelled documents P and unlabelled documents U, in order to form RN, while the second one uses k-means clustering to further purify RN, as to get to RN'.

Method 1: Rocchio The first method is quite simple (See Algorithm 1) [10]. Every document d is represented as vector $\mathbf{d} = (q_1, ..., q_n)$, while each element q_i represents a word w_i with $q_i = tf_i \times idf_i$ (Refer to section **2.1**).

- tf_i = number of occurrences of word w_i in document d.
- $idf_i = \log(|D|/df(w_i))$ with $|D|$ = total number of documents and $df(w_i)$ = number of documents containing w_i.

From Algorithm 1 both $\mathbf{c}^+$ and $\mathbf{c}^-$ are prototype vectors for the two classes. α and β adjust for the relative impact of training examples from the two classes. In this case, values $\alpha = 16$ and $\beta = 4$ are recommended [7]. The algorithm then uses cosine similarities[2] between each test document $\mathbf{d}'$ and the prototype vectors to compute the similarity. From there each document is classified into whichever class it is most similar to. Negatively classified documents now form RN.

Algorithm 1 Rocchio

1. Assign the unlabelled set U the negative class, and the positive set P the positive class;
2. Let $\mathbf{c}^+ = \alpha \frac{1}{|P|} \sum_{\mathbf{d} \epsilon P} \frac{\mathbf{d}}{||\mathbf{d}||} - \beta \frac{1}{|U|} \sum_{\mathbf{d} \epsilon U} \frac{\mathbf{d}}{||\mathbf{d}||}$;
3. Let $\mathbf{c}^- = \alpha \frac{1}{|U|} \sum_{\mathbf{d} \epsilon U} \frac{\mathbf{d}}{||\mathbf{d}||} - \beta \frac{1}{|P|} \sum_{\mathbf{d} \epsilon P} \frac{\mathbf{d}}{||\mathbf{d}||}$;
4. **for** each document d' in U **do**
5. **if** $sim(\mathbf{c}^+, \mathbf{d}') \leq sim(\mathbf{c}^-, \mathbf{d}')$ **then**
6. $RN = RN \cup \{d'\}$;

[2] Euclidean distances serve the same purpose as the feature vectors are normalised [11].

Method 2: Rocchio with clustering There may exist an issue with the first method since Rocchio is a linear classifier that constructs it's separating hyperplane using cosine similarities. In a case where the decision boundary is non-linear, there is a chance that some of the positive documents in U may be wrongly classified and placed in RN. This could hamper the performance of the final SVM classifier. Thus, the second method in Algorithm 2 attempts to further purify RN by means of k-means clustering, eventually finalising RN' as the final set of reliable negative documents [10]. This approach uses the clustering to apportion RN into many similar groups. After that Rocchio is used again with the clusters as the negative input and P for the positive input to build new classifiers for each cluster. Use these classifiers to find and remove probable positive documents from each cluster. Unlike in the first case where the negative set was fairly heterogeneous, clustering results in both the positive set and now the cluster (negative) set, to be fairly homogeneous. This enables Rocchio to compute more representative prototype vectors.

In Algorithm 2, line 1 performs initial algorithm (Algorithm 1) to construct RN and in lines 2 and 3, k-means clustering is performed. Lines 5 and 6 construct the aforementioned prototype vectors for the positive and the negative class. The removal of positive documents starts at line 8, while lines 9 and 10 perform the actual extraction. After the extraction the final reliable negative document set RN' is formed.

Algorithm 2 Rocchio with clustering

1. Perform Algorithm 1 and generate the initial negative set RN;
2. Choose k initial cluster centres $\{M_1, M_2, ..., M_k\}$ randomly from RN;
3. Perform k-means clustering to produce k clusters $\{O_1, O_2, ..., O_k\}$;
4. **for** $j = 1$ to k
5. $\quad \mathbf{n}_j = \alpha \frac{1}{|O_j|} \sum_{\mathbf{d} \epsilon O_j} \frac{\mathbf{d}}{||\mathbf{d}||} - \beta \frac{1}{|P|} \sum_{\mathbf{d} \epsilon P} \frac{\mathbf{d}}{||\mathbf{d}||}$;
6. $\quad \mathbf{p}_j = \alpha \frac{1}{|P|} \sum_{\mathbf{d} \epsilon P} \frac{\mathbf{d}}{||\mathbf{d}||} - \beta \frac{1}{|O_j|} \sum_{\mathbf{d} \epsilon O_j} \frac{\mathbf{d}}{||\mathbf{d}||}$;
7. $RN' = \{\}$;
8. **for** each document $d_i \epsilon RN$ **do**
9. $\quad$ Find the nearest prototype vector $\mathbf{p}_v$ tod_i, where $v = \arg\max_j sim(\mathbf{p}_j, d_i)$;
10. $\quad$ **if** there exist an $\mathbf{n}_j$with $j = 1, 2, ..., k$ such that $sim(\mathbf{p}_v, d_i) \leq sim(\mathbf{n}_j, d_i)$ **then**
11. $\quad\quad RN' = RN' \cup \{d_i\}$;

Constructing final classifier

The second step in [10] by Li & Lui is to build the final classifier using the positive document set P and the newly formed reliable negative document set RN'. The method shown in Algorithm 3 is to build a SVM classifier iteratively. The reason for this is that the reliable set RN or RN' from the first step may in fact not be large enough to train the best classifier. Let Q be the remaining unlabelled documents[3]. The reason for training the SVM iteratively is to extract more negative documents from Q on each iteration, therefore bolstering the amount of available negative documents and further improving performance. The iteration from line 3 to line 5 stops when Q is exhausted of negative documents. If it happens that one of the iterations misbehaves and extracts positive documents from Q and adds them to RN / RN', then the final SVM classifier will perform quite poorly. To avoid using a poor final classifier, choose between using the first classifier S_i or the last one S_{last}. To choose between the two, first use S_{last} to classify the labelled set P. If more than 5% of the positive documents are classified incorrectly (negative), it indicates that an iteration has gone wrong and S_1 should rather be used as the final classifier.

[3] $Q = U - RN$ or $Q = U - RN'$

Algorithm 3 Train final SVM

1. Assign label +1 to every document in P;
2. Assign label -1 to every document in RN/RN';
3. Train initial SVM classifier S_i, using P and RN/RN', iteratively (starting at $i = 1$) over lines 3-5;
4. Use S_i to classify Q. Let documents in Q that are classified as negative be W;
5. **if** $W = \{\}$ **then** *stop;*
   else $Q = Q - W$;
   $RN = RN \cup W$ or $RN' = RN' \cup W$;
   go to line 3;
6. Classify P using S_{last}, the final SVM classifier;
7. **if** more than 5% classified as negative **then** use S_1 as final classifier;
   else use S_{last};

3 Application

Data

In order to test the methods on real data, it is important to first implement them on a base data set that is known to work well with classification algorithms. The data to be used for the baseline testing is the commonly used *20 News data set* [9], which consists of approximately 20 000 news documents, all labelled as one of twenty topics. The two topics used (since the methods in this paper only consider binary classification) for testing the methods are *alt.atheism* and *soc.religion.christian,* which reduces the number documents to approximately 2 000 (1 000 per topic). The data was accessed via *scikit-learn's* built-in data sets [13].

The real data that will be used to test the algorithms on is the GOP data set was acquired from *Crowdflower*[4]. This data set consists of the text of tweets which was streamed by *Crowdflower* during the first Republican debate in 2016. Originally, the data set contained tweets that were labelled as one of *Positive, Negative* or *Neutral* sentiment. Again, due to the implementation of only binary classification, all tweets labelled as *Neutral* were removed (Table 1). The composition of classes in the GOP data is relatively imbalanced. After removing duplicates in the text column, the number of tweets exhibiting positive sentiment is 1675, while the number of negative tweets is 6070.

	sentiment	**text**	**label**
1	Positive	RT @ScottWalker: Didn't catch the full #GOPdeb...	1
2	Positive	RT @RobGeorge: That Carly Fiorina is trending ...	1
4	Positive	RT @DanScavino: #GOPDebate w/ @realDonaldTrump...	1
5	Positive	RT @GregAbbott_TX: @TedCruz: "On my first day ...	1
5	Negative	RT @warriorwoman91: I liked her and was happy ...	0

Table 1: Sample GOP data

Before initiating the process of building the reliable negative data sets RN and RN', it is necessary to clean up the feature text. Some parts of text, for example, the user name at which a tweet is directed (e.g. *@ScottWalker* in Table 1) and unique URLs, have no influence on the sentiment of a tweet. Note that even though a profile name may receive a weighty TF-IDF score since it is unique and discriminating, logically it might have no effect on the sentiment of a tweet whatsoever [11]. Following the manual extraction of some obvious unique words from the text, the built-in remove stop words option in *scikit-learn's* TfIdfVectoriser [13] function removes all the common words such as *is* and *or*.

In order to implement and evaluate the performance of the abovementioned semi-supervised methods, Li & Lui [10] decided upon a method for extrapolating sets of positively labelled documents from the overall set in various fractions, giving one an opportunity to evaluate the methods for various fractions of labelled and unlabelled data. Let α take on values in the range 5, 15, 25, 35, 45 and 65. These values represent the percentage of positively labelled data (P) to be used along with the unlabelled set (U) in the implementation of the methods. Take $\alpha = 5$ as an example: Of the 1675 tweets labelled as positive, 5% is used as the positive set P, while the other 95% is combined with 95% of the negative tweets to form the unlabelled set, U. This is done for all 7 values of α. The motivation for using equal fractions of set P and set N is to preserve the natural class ratio present in the data set.

Evaluation

For this specific case, the test data set (on which the methods will be evaluated), will be the unlabelled set U for each value of α. The reason for using U as the testing set in order to align it with the objective of classifying the unlabelled data and to better compare the results with the results reported by Li & Lui [10] since used U as the test data as well. The main metrics used for evaluation is F1 scores, accuracy scores

[4] `https://www.crowdflower.com/`

and ROC curves. Along with these metrics, tabled results containing F1, precision, recall and accuracy scores have been included in the Appendix (Tables 4 to 6). For evaluating general performance, classification accuracy is used. This is simply the fraction of correctly predicted documents from the test set. Note that using accuracy as a stand-alone measure should be avoided given the large class imbalance. The F1 score is used in evaluating the prediction accuracy in binary classification and is widely used especially in text classification. It is calculated as the harmonic mean between recall and precision (also tends toward the lower of the two values) [15]. It measures the performance of a model with regards to each class, which makes it an ideal measure for a classification problem with such disparity between classes. The same metrics were used by Li & Lui [10], thus a comparison can be made between the final results. In order to achieve robust results, the whole process, from randomly generating P and U and constructing RN/RN' to the training of the final classifier was repeated 10 times. From this, the mean values for the tabled scores were calculated.

Results

When interpreting the results, the scores for the initial and final classifier of both methods used in determining RN and RN' are plotted in Figures 2a, 2b, 3a, and 3b. For each method, the initial and final SVM corresponds to S_i and S_{last} in Algorithm 3. The scores are plotted for a model, in this case, logistic regression [12], that was trained on the same data as the initial Rocchio classifier in Method 1, to illustrate the performance gained by using one of the two semi-supervised methods. The ROC curves for the initial SVM of both methods is plotted in Figure 4a and Figure 4b, to illustrate the differences between the two in terms of a different metric. Table 3 contains the total number of observations used as negatively labelled documents for training the three classifiers (Logistic regression and SVMs for Methods 1 and 2). The logistic regression model is trained using P as the positive set and the original unlabelled set as the negative set, while S_i and $S_{last}$was trained using P as the positive set with RN and RN' as the respective negative document sets.

	20 Newsgroups data		GOP data	
Method	Accuracy	F1	Accuracy	F1
SVM	0.975	0.978	0.814	0.56
Logistic regression	0.964	0.969	0.8	0.54

Table 2: F1 and accuracy scores for supervised models

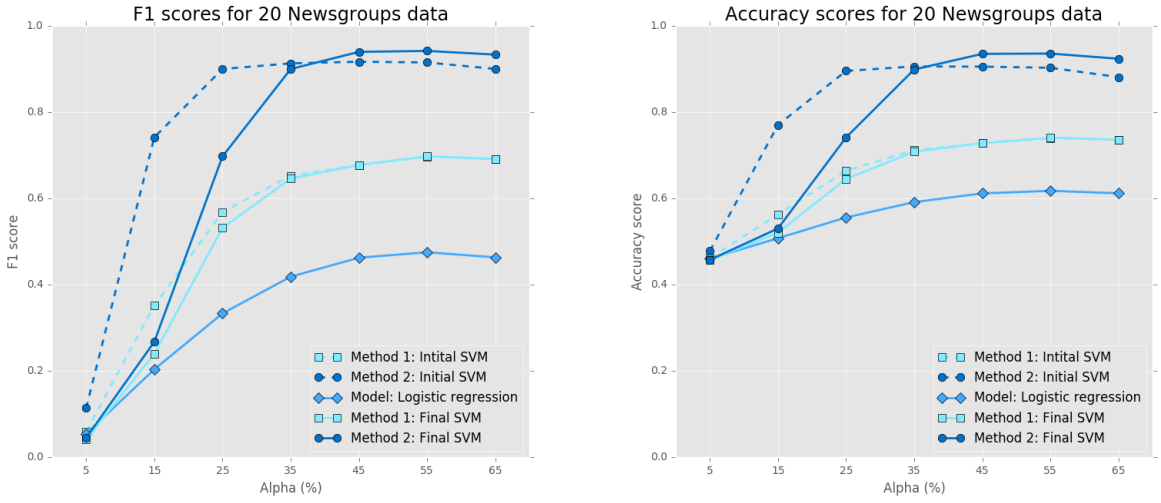
Additionally, Table 2 contains scores for a supervised logistic regression model and a supervised SVM model which were trained on both data sets (80% training data). Table 2 should serve as a reference point when comparing the differences between supervised and semi-supervised. It is important to keep in mind that a much larger fraction of labelled data is required for training purposes with the supervised models.

		20 Newsgroups data			GOP data		
α	Class	Original unlabelled set	*RN*	*RN'*	Original unlabelled set	*RN*	*RN'*
5%	Negative	760	758	749	5767	5680	3499
	Positive	948	861	742	1592	1440	839
15%	Negative	680	679	502	5160	4790	2272
	Positive	848	568	95	1424	978	342
25%	Negative	600	598	425	4553	4078	1694
	Positive	748	421	30	1257	761	236
35%	Negative	520	518	344	3946	3342	1358
	Positive	649	320	23	1089	595	147
45%	Negative	440	439	306	3339	2733	1167
	Positive	549	245	9	922	452	106
55%	Negative	360	358	221	2732	2196	1018
	Positive	449	186	10	754	325	61
65%	Negative	280	279	136	2125	1715	646
	Positive	349	138	6	587	191	44

Table 3: Number of assumed negative observations before and after implementation of semi-supervised methods.

20 Newsgroup data

In Figures 2a and 2b it is clear that the semi-supervised methods performed well on the 20 Newsgroups data. The accuracy scores (see Table 4 in the Appendix) are very high, as are the F1-scores with increasing values for α, particularly for the second method (using *RN'*). Note that scores for the second method can be compared to the supervised models in Table 2. Furthermore, it seems that the iterative training of the final classifier does not have a significant impact on performance. In fact, it performs worse for lower values of α. It is clear that none of the methods perform particularly well for α values lower than 25% since the ratio between labelled and unlabelled data at those values is too small. These results are similar to what Li & Lui [10] found. Hence this data set serves as the perfect baseline for comparison to the results of the GOP data.



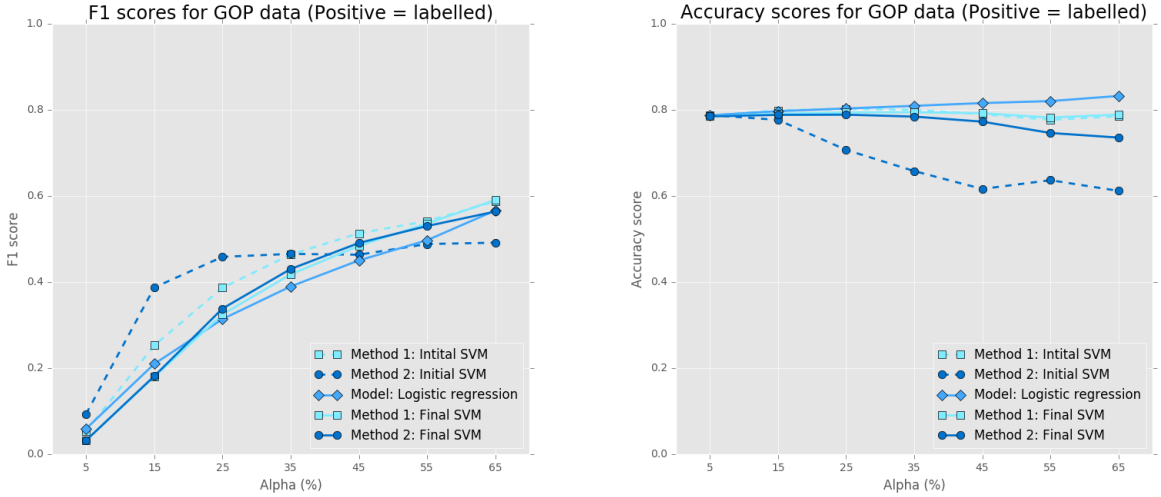
(a) F1-scores over all α

(b) Accuracy scores over all α

Figure 2: Accuracy and F1 scores for 20 newsgroups data

GOP data

Even though the accuracy scores appear to be acceptable (Figure 3b), both the methods reach F1-scores of just under 0.6 for $\alpha = 65\%$. These scores do compare well to the supervised case, only performing slightly worse. This is encouraging, considering the different amounts of labelled data used. The logistic regression (semisupervised) model performs the best overall (see Table 5), indicating that in this case, it might be better to use neither method. The poor performance may be due to the disparity in classes among the data where the labelled document class is underrepresented in the unlabelled set. Performance is considerably better (Table 6) for the case where the labelled used for implementing the methods are documents from the negative document set, which indicates that (in this case) the models benefits from the prevalence of negative documents in the unlabelled set. Li & Lui [6] did not encounter the same results in their tests, which was also implemented on a data set with considerable class imbalances, which may indicate that there exists a different issue with this data set. Figure 3a shows the plotted F1 scores for the case where positive labelled documents were used as labelled data to implement the methods. In Table 6 the opposite is shown, where negative labelled documents are used as the labelled data in the training set.



(a) F1 scores over all values of α

(b) Accuracy scores over all α

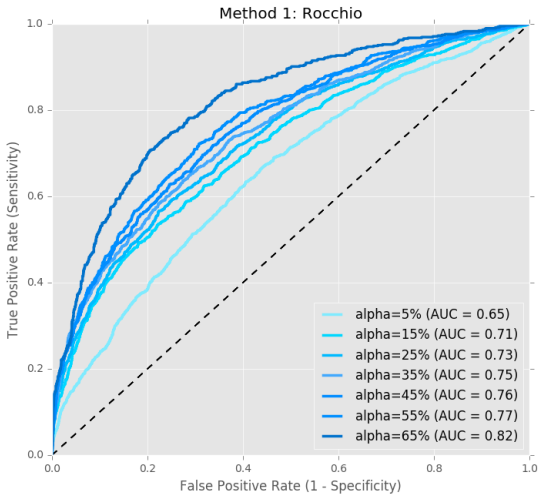
Figure 3: Accuracy and F1 scores for GOP data

The plotted ROC curves in Figures 4a and 4b show that there generally does not exist a significant difference between S_i and S_{last}, however it does indicate again the performance gains achieved by increasing the value for α.

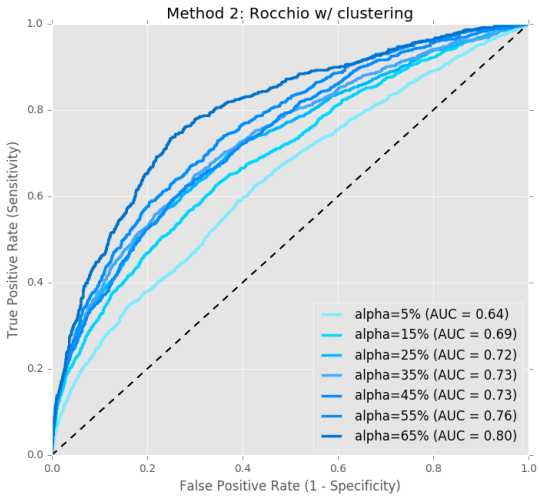
4 Conclusion

The methods discussed provide a good starting point for improving efficiency in classification tasks where labelled data sets are problematic to acquire. It seems that the optimal value for α would be some point between *25%* and *45%*, which would reduce the amount of labelled data required for classification considerably in the case where the labelled class is in the minority. Both methods thoroughly outperformed the logistic regression model on the 20 Newsgroups data when implemented on a data set with balanced classes (Figures 2a and 2b).

From Figure 3a it is clear that imbalanced classes have an adverse effect on the performance of the classifier. It should be noted that using documents that are more prevalent in the overall data set as the labelled data in



(a) Method 1: ROC curves for all values of α



(b) Method 2: ROC curves for all values of α

Figure 4: ROC curves for GOP data

these methods is pointless, since the same accuracy can be achieved by simply classifying every single document as the prevalent class (Table 6 in the Appendix) and it would be similar to training a classification model on data where the two labels are the same. Real-world data sets would likely not be of a balanced nature and would display feature set issues similar to that of the GOP data set, which may lead to undesirable results.

The performance of the two methods were overall acceptable considering the data used and that very little preprocessing and feature engineering was done, therefore it opens up various alternative avenues for development of these methods, which might enable them to be more robust to class imbalance. One alternative to these methods that can be explored is making use of Joachims' TSVMs [6] to extract and determine the final negative set, RN, or combining it with active learning could also be an option. Furthermore, if one of the two methods for creating RN or RN' is performing well then one can focus on the final step and explore the possibility of finding an optimal classifier in between Si and Slast. Finally, the ultimate goal would be the ability to expand these methods to a multi-class environment, although a very strong performing foundation would be required.

References

[1] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[2] B. Charalampakis, D. Spathis, E. Kouslis, and K. Kermanidis. A comparison between semi-supervised and supervised text mining techniques on detecting irony in greek political tweets. *Engineering Applications of Artificial Intelligence*, 2016.

[3] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[4] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer Series in Statistics Springer, Berlin, 2001.

[5] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning*, volume 6. Springer, 2013.

[6] T. Joachims. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning (ICML)*, volume 99, pages 200–209, 1999.

[7] T. Joachims. *Learning to Classify Text using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, 2002.

[8] Thorsten Joachims. Svmlight: Support vector machine. *SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund*, 19(4), 1999.

[9] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

[10] X. Li and B. Liu. Learning to classify texts using positive and unlabeled data. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 3, pages 587–592, 2003.

[11] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[12] K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[13] 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.

[14] John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, April 1998.

[15] C. Sammut and G. Webb. *Encyclopedia of Machine Learning*. Springer Science & Business Media, 2011.

[16] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, March 2002.

[17] T. Zhang and F. Oles. The value of unlabeled data for classification problems. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1191–1198. Citeseer, 2000.

Appendix

Notation

For graphs:

- *Method 1: Initial SVM* - Refers to the initial trained SVM classifier, S_i, for the first semi-supervised technique, Rocchio.
- *Method 1: Final SVM* - Refers to the final trained SVM classifier, S_{last}, for the first semi-supervised technique, Rocchio.
- *Method 2: Initial SVM* - Refers to the initial trained SVM classifier, S_i, for the second semi-supervised technique, Rocchio with clustering.
- *Method 2: Final SVM* - Refers to the final trained SVM classifier, S_{last}, for the second semi-supervised technique, Rocchio with clustering.
- *Model: Logistic Regression* - Refers to the logistic regression classifier trained on P and U (see below). Training of this classifier made no use of any semi-supervised techniques.

For algorithms:

- P - Set of positively labelled documents/observations to be used as labelled data in implementation of semi-supervised methods.
- N - Set of negatively labelled documents, some of which will form part of the unlabelled set, U.
- U - Set of unlabelled documents, comprised of $(1-\alpha)\%$ positively labelled documents and $(1-\alpha)\%$ negatively labelled documents.
- RN - Reliable negative document set obtained after implementing the first semi-supervised method.
- RN' - Reliable negative document set obtained after implementing the second semi-supervised method.
- α - Fractions of positively labelled documents used as labelled data, ranging from *5%* to *65%*.
- S_i - Initial SVM classifier trained by using either method. The training procedure is outlined in Algorithm 3.
- S_{last} - Initial SVM classifier trained by using either method. The training procedure is outlined in Algorithm 3.
- Q - Remaining unlabelled documents after generating *RN / RN'. (*$Q = U$ - RN or $Q = U$ - RN'*)*
- W - Set of documents that are classified as negative when training final classifier (Algorithm 3).

Code

Data & preprocessing:

```
#%%
import pandas as pd
df = pd.read_csv('Sentiment.csv')

#%%
##Remove unwanted entries:

df= df.drop(df.columns[[0,1,2,3,4,6,7,8,9,10,11,12,13,14,16,17,18,19,20]],axis=1)
df = df.loc[df.sentiment != 'Neutral']

#%%
##Redefine labels:

def relabel (df):
    if df.sentiment == "Negative":
        return 0
    elif df.sentiment == "Positive":
        return 1
    return "Other"

df["label"] = df.apply(lambda df: relabel (df),axis=1)

#%%
#Filter initial stopwords: (Will add common stopwords when vectorising)

df["Clean_Text"]=df.text.str.replace("(?<=\W)[@]\S*","")
df["Clean_Text"]=df.Clean_Text.str.replace("[@]\S*","")
df["Clean_Text"]=df.Clean_Text.str.replace("RT","")


#%%
df.drop(df.columns[[0,1]],axis=1,inplace=True)
print df.head(5)

#%%
df.drop_duplicates('Clean_Text',inplace=True)
df.label.value_counts()
print 'Full set : ',df.shape
#%%
df.to_csv(r'full.csv', header=True, index=None, mode='w', encoding='utf-8')
```

Semi-supervised methods & ROC plotting:

```
#%%
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.neighbors import NearestCentroid
from sklearn.cluster import KMeans
from sklearn.linear_model import LogisticRegression
```

```
from sklearn.metrics import accuracy_score, recall_score, precision_score,\
                            f1_score,roc_curve, auc
import itertools
#plt.style.use('seaborn-paper')
plt.style.use('ggplot')
#%%
df = pd.read_csv("full.csv")
vc = df.label.value_counts()
#%%
alpha = [5,15,25,35,45,55,65]
posval = [ int(float(alpha[i])/100*vc[1]) for i in xrange(len(alpha)) ]
negval = [ int(float(alpha[i])/100*vc[0]) for i in xrange(len(alpha)) ]
print 'Samples for P: ' ,posval
print 'Samples for N: ' ,negval
#%%
pos = df[df.label==1]
neg = df[df.label==0]
P,P_U = {},{}
N,N_U = {},{}
U = {}
data = {}
pos = pos.iloc[np.random.permutation(len(pos))]
pos =pos.reset_index(drop=True)
neg.iloc[np.random.permutation(len(neg))]
neg =neg.reset_index(drop=True)
for i in xrange(len(posval)):
    P[i], P_U[i]= pos.head(posval[i]), pos.tail(len(pos)-posval[i])
    N[i], N_U[i]= neg.head(posval[i]), neg.tail(len(neg)-negval[i])
    objs = [P_U[i],N_U[i]]
    U[i] = pd.concat(objs,axis=0,join='outer',ignore_index=True)
    U[i]['nlabel'] = 0
    P[i]['nlabel'] = 1
    data[i] = pd.concat([U[i],P[i]],axis=0,ignore_index=True)
#%%
X = {}
y= {}
X_U = {}
vectorize = TfidfVectorizer(stop_words='english')
for i in xrange(len(data)):
    X[i] = vectorize.fit_transform(data[i].Clean_Text)
    y[i] = data[i].nlabel.astype('int64')
    X_U[i] = vectorize.transform(U[i].Clean_Text)
#%%
rocchio = NearestCentroid(metric='euclidean')
clf_1 = {}
for i in xrange(len(data)):
    clf_1[i] = rocchio.fit(X[i],y[i])
    U[i]['preds']=clf_1[i].predict(X_U[i])
#%%
RN1 = {}
data_m1 = {}
for i in xrange(len(data)):
    RN1[i] = U[i].loc[U[i].preds == 0]
    RN1[i] = RN1[i].drop(RN1[i].columns[[2]],axis=1)
    RN1[i].rename(columns = {'preds':'nlabel'}, inplace = True)
    data_m1[i] = pd.concat([RN1[i],P[i]],axis=0,ignore_index=True)
#%%
kmeans = KMeans(n_clusters = 10)
X_RN1 = {}
```

```
clust = {}
cdata_m2 = {}
for i in xrange(len(RN1)):
    X[i] = vectorize.fit_transform(data[i].Clean_Text)
    X_RN1[i] = vectorize.transform(RN1[i].Clean_Text)
    clust[i] = kmeans.fit(X_RN1[i])
    cdata_m2[i] = pd.DataFrame({'Cluster_labels':kmeans.labels_,
                                'Clean_Text':RN1[i].Clean_Text,'nlabel' : 0})
    cdata_m2[i].set_index('Cluster_labels',drop=False,inplace=True)
    cdata_m2[i].sort_index(axis=0,inplace=True)
#%%
label = np.arange(0,10,1)
df_clust = []
clusters = {}
dfdf = {}
for j in xrange(len(data)):
    for i in xrange(10):
        df_clust1 = {}
        clusters[i] = cdata_m2[j].loc[cdata_m2[j].Cluster_labels == label[i]]
        obj = [clusters[i],P[j]]
        df_clust1[i]= pd.concat(obj,axis=0,join='outer',ignore_index=True)
        df_clust.append(df_clust1)
#%%
clustdf = {}
for j in xrange(len(data)):
    clustdf[j] = {}
    for i,dics in enumerate(df_clust):
        if (j*10)<=i<(j*10+10):
            m = (i-j*10)
            clustdf[j][m] = dics[m]
#%%
for j in xrange(7):
    for i in xrange(10):
        clustdf[j][i]= clustdf[j][i].drop(clustdf[j][i].columns[[1,2]],axis=1)
#%%
RN2 = RN1
predictions = {}
clfc = {}
for i in xrange(7):
    clfc[i]={}
    predictions[i] = {}
    X[i] = vectorize.fit_transform(data[i].Clean_Text)
    for x in xrange(10):
        clfc[i][x] = rocchio.fit(vectorize.transform(clustdf[i][x].Clean_Text),
                                 clustdf[i][x].nlabel)
        predictions[i][x] = clfc[i][x].predict(X_RN1[i])
        RN2[i]['pred_%s' %(x)] = predictions[i][x]

#%%
sums = {}
for i in xrange(len(RN2)):
    sums[i] = RN2[i].drop(RN2[i].columns[[0,1,2]],axis=1)
#%%
for i in xrange(7):
    RN2[i]['sums'] = sums[i].sum(axis=1)
#%%
valcnt = {}
RN_Final = {}
for i in xrange(len(RN2)):
```

```
    RN_Final[i] = RN2[i].loc[RN2[i]['sums']<=8]
    valcnt[i] = RN_Final[i].label.value_counts()
#%%
data_m2 = {}
for i in xrange(7):
    data_m2[i] = pd.concat([RN_Final[i],P[i]],axis=0,join='outer',ignore_index=True)
    data_m2[i] = data_m2[i].drop(data_m2[i].columns[[3,4,5,6,7,8,9,10,11,12,13]],
                                 axis=1)
#%%
uvc = {}
rnvc = {}
print "Value counts: "
print 'Alpha |   Class   | Origin  |  RN1   |  RN2   |'
print '-----------------------------------------------'
for i in xrange(len(data)):
    uvc[i] = U[i].label.value_counts()
    rnvc[i] = RN1[i].label.value_counts()
    valcnt[i] = RN_Final[i].label.value_counts()
    print ' {0}%   '.format(alpha[i]), ' Negative   ' ,uvc[i][0], '    ',rnvc[i][0], ' ',valcnt[i][0]
    print '        ', 'Positive   ' ,uvc[i][1], '    ',rnvc[i][1], '   ',valcnt[i][1]
    print '-----------------------------------------------'
#%%
X_1 = {}
X_2 = {}
y_1 = {}
y_2 = {}
X_test = {}
y_test = {}
for i in xrange(7):
    X[i] = vectorize.fit_transform(data[i].Clean_Text)
    X_1[i] = vectorize.transform(data_m1[i].Clean_Text)
    X_test[i] = vectorize.transform(U[i].Clean_Text)
    X_2[i] = vectorize.transform(data_m2[i].Clean_Text)
    y_1[i] = data_m1[i].nlabel
    y_2[i] = data_m2[i].nlabel
    y_test[i] = U[i].label.astype('int64')
#%%
final_clf1 = {}
final_clf2 = {}
final_clf3 = {}
score_m1= {}
score_m2 = {}
proba_m1 = {}
proba_m2 = {}
npred_m1 = {}
npred_m2 = {}
score_m3= {}
proba_m3 = {}
npred_m3 = {}

for i in xrange(7):
    svm = SVC(kernel='linear',class_weight='balanced')
    final_clf1[i] = svm.fit(X_1[i],y_1[i])
    score_m1[i] = final_clf1[i].score(X_test[i],y_test[i])
    proba_m1[i] = final_clf1[i].decision_function(X_test[i])
    npred_m1[i] = final_clf1[i].predict(X_test[i])

    svm = SVC(kernel='linear',class_weight='balanced')
    final_clf2[i] = svm.fit(X_2[i],y_2[i])
```

```
    score_m2[i] = final_clf2[i].score(X_test[i],y_test[i])
    proba_m2[i] = final_clf2[i].decision_function(X_test[i])
    npred_m2[i] = final_clf2[i].predict(X_test[i])

    logreg = LogisticRegression(class_weight='balanced')
    final_clf3[i] = logreg.fit(X[i],y[i])
    score_m3[i] = final_clf3[i].score(X_test[i],y_test[i])
    proba_m3[i] = final_clf3[i].predict_proba(X_test[i])
    npred_m3[i] = final_clf3[i].predict(X_test[i])

#%%
def evaluation(npred1,npred2,npred3,y2,model_list = ['Method 1',
                                                     'Method 2','Method 3']):
        acc_m1 = accuracy_score(y2,npred1)
        acc_m2 = accuracy_score(y2,npred2)
        acc_m3 = accuracy_score(y2,npred3)
        recall_m1 = recall_score(y2,npred1)
        recall_m2 = recall_score(y2,npred2)
        recall_m3 = recall_score(y2,npred3)
        precision_m1 = precision_score(y2,npred1)
        precision_m2 = precision_score(y2,npred2)
        precision_m3 = precision_score(y2,npred3)
        f1_m1 = f1_score(y2,npred1)
        f1_m2 = f1_score(y2,npred2)
        f1_m3 = f1_score(y2,npred3)
        return recall_m1,recall_m2,recall_m3,precision_m1,precision_m2,precision_m3,\
               f1_m1,f1_m2,f1_m3,acc_m1,acc_m2,acc_m3
#%%
def roc_calcs(probs,y2):
    fprs = {}
    tprs = {}
    aucs = {}
    thresh = {}

    for k in xrange(len(probs)):
        fprs[k],tprs[k],thresh[k] = roc_curve(y2,probs[k])
        aucs[k] = auc(fprs[k],tprs[k])
    return fprs, tprs, thresh, aucs

#%%
roc_vals = {}
probsa = {}
for i in xrange(7):
    probsa[i] = [proba_m1[i],proba_m2[i],proba_m3[i][:,1]]
    roc_vals[i] = roc_calcs(probsa[i],y_test[i])

#%%
fpr_m1 = {}
fpr_m2 = {}
fpr_m3 = {}
tpr_m1 = {}
tpr_m2 = {}
tpr_m3 = {}
auc_m1 = {}
auc_m2 = {}
auc_m3 = {}
fprs1 = []
tprs1 = []
aucs = []
```

```
for i in xrange(7):
    fpr_m1[i] = roc_vals[i][0][0]
    fpr_m2[i] = roc_vals[i][0][1]
    fpr_m3[i] = roc_vals[i][0][2]
    tpr_m1[i] = roc_vals[i][1][0]
    tpr_m2[i] = roc_vals[i][1][1]
    tpr_m3[i] = roc_vals[i][1][2]
    auc_m1[i] = roc_vals[i][3][0]
    auc_m2[i] = roc_vals[i][3][1]
    auc_m3[i] = roc_vals[i][3][2]
    fprs1.append([fpr_m1[i],fpr_m2[i],fpr_m3[i]])
    tprs1.append([tpr_m1[i],tpr_m2[i],tpr_m3[i]])
    aucs.append([auc_m1[i],auc_m2[i],auc_m3[i]])

#%%
list0 = []
list1 = []
list2 = []

list3 = []
list4 = []
list5 = []

list6 = []
list7 = []
list8 = []

for i in xrange(0,7):
    list0.append(fprs1[i][0])
    list1.append(fprs1[i][1])
    list2.append(fprs1[i][2])
    list3.append(tprs1[i][0])
    list4.append(tprs1[i][1])
    list5.append(tprs1[i][2])
    list6.append(aucs[i][0])
    list7.append(aucs[i][1])
    list8.append(aucs[i][2])

fprs2 = [list0,list1,list2]
tprs2 = [list3,list4,list5]
aucs2 = [list6,list7,list8]

#%%
titles = ['Method 1: Rocchio',
          'Method 2: Rocchio w/ clustering','Logistic regression']
methods1 = ['alpha=5% (AUC = {0:.2f})','alpha=15% (AUC = {0:.2f})',
            'alpha=25% (AUC = {0:.2f})','alpha=35% (AUC = {0:.2f})',
            'alpha=45% (AUC = {0:.2f})','alpha=55% (AUC = {0:.2f})',
            'alpha=65% (AUC = {0:.2f})']
colors1 = ['#7FEBFF','#00D7FF','#00BAFF','#43A9FD','#038DFF','#008FFF',
            '#0072CC']

def roc_plot(fpr,tpr,auc,methods,colors,title='foo'):
    plt.figure(figsize=(8,7))
    plt.title(title,fontsize='x-large')
    plt.xlabel('False Positive Rate (1 - Specificity)',fontsize='large')
    plt.ylabel('True Positive Rate (Sensitivity)',fontsize='large')
    for x in xrange(0,len(methods)):
```

```
        plt.plot(fpr[x],tpr[x],color=colors[x],
                 label=methods[x].format(auc[x]),linewidth=2.5)
        plt.legend(loc=4,fontsize='large')
        plt.xlim([0.0,1.0])
        plt.ylim([0.0,1.0])
        plt.grid(True)
        plt.savefig('3roc_figs{0}.png'.format(alpha[j]),dpi=100)
        plt.plot([0, 1], [0, 1], 'k--')
    plt.show()
#%%

for j in xrange(3):
    roc_plot(fprs2[j],tprs2[j],aucs2[j],methods1,colors1,title=titles[j])
plt.show()
#%%
methods = ['Method: Rocchio','Method: Rocchio w/ clustering',
           'Logistic regression']
f1_scores_m1 = []
f1_scores_m2 = []
f1_scores_m3 = []
recall_scores_m1 = []
recall_scores_m2 = []
recall_scores_m3 = []
precision_scores_m1 = []
precision_scores_m2 = []
precision_scores_m3 = []
accuracy_m1 = []
accuracy_m2 = []
accuracy_m3 = []
evalu = {}
for i in xrange(7):
    evalu[i] = evaluation(npred_m1[i],npred_m2[i],npred_m3[i],y_test[i],
                          model_list = methods)
    recall_scores_m1.append(evalu[i][0])
    recall_scores_m2.append(evalu[i][1])
    recall_scores_m3.append(evalu[i][2])
    precision_scores_m1.append(evalu[i][3])
    precision_scores_m2.append(evalu[i][4])
    precision_scores_m3.append(evalu[i][5])
    f1_scores_m1.append(evalu[i][6])
    f1_scores_m2.append(evalu[i][7])
    f1_scores_m3.append(evalu[i][8])
    accuracy_m1.append(evalu[i][9])
    accuracy_m2.append(evalu[i][10])
    accuracy_m3.append(evalu[i][11])
#%%
obj1 = {}
obj2 = {}
for i in xrange(7):
    obj1[i] = [RN1[i].drop(RN1[i].columns[np.arange(3,14,1)], axis=1),
               U[i].drop(U[i].columns[[3]],axis=1)]
    obj2[i] = [RN_Final[i].drop(RN_Final[i].columns[np.arange(3,14,1)], axis=1),
               U[i].drop(U[i].columns[[3]],axis=1)]
#%%
Q_m1 = {}
Q_m2 = {}
for i in xrange(7):
    Q_m1[i] = pd.concat(obj1[i],axis=0,join='outer')\
    .drop_duplicates('Clean_Text',keep=False)
```

```
    Q_m2[i] = pd.concat(obj2[i],axis=0,join='outer')\
    .drop_duplicates('Clean_Text',keep=False)
#%%
def iter_class(Q1,data1,test_set):
    data2 = data1
    Q=Q1
    W = pd.DataFrame()
    for i in itertools.count():
        data2 = pd.concat([W,data2],axis=0,ignore_index=True)
        print '---------------'
        print 'Iteration: ' ,i
        print '---------------'
        ytrain = data2.nlabel
        y_test = test_set.label
        Xtrain = vectorize.transform(data2.Clean_Text)
        X_pred = vectorize.transform(Q.Clean_Text)
        Xtest = vectorize.transform(test_set.Clean_Text)
        svm = SVC(kernel='linear',class_weight='balanced')
        clf = svm.fit(Xtrain,ytrain)
        Q['preds'] = clf.predict(X_pred)
        W = Q.loc[Q.preds == 0]
        W = W.drop(W.columns[[2]],axis=1)
        W.rename(columns = {'preds':'nlabel'}, inplace = True)
        print len(Q),len(W)
        Q = Q.loc[Q.preds != 0]
        Q = Q.drop(Q.columns[[3]],axis=1)
        predict = clf.predict(Xtest)
        recalls = recall_score(y_test,predict)
        precisions = precision_score(y_test,predict)
        f1s = f1_score(y_test,predict)
        accuracys = clf.score(Xtest,y_test)
        if len(W)==0 or len(Q)==0:
            break
    return predict,precisions,recalls,f1s,accuracys
#%%
predicted_val1 = {}
predicted_val2 = {}
f1_final1 = []
f1_final2 = []
rec_final1 = []
rec_final2 = []
prec_final1 = []
prec_final2 = []
acc_final1 = []
acc_final2 = []
lastiter1 = {}
lastiter2 = {}

for x in xrange(7):
    print '*****************'
    print '****SET','1','.',x,'****'
    print '*****************'
    lastiter1[x] = iter_class(Q_m1[x],data_m1[x],U[x])
    print '*****************'
    print '****SET','2','.',x,'****'
    print '*****************'
    lastiter2[x] = iter_class(Q_m2[x],data_m2[x],U[x])
    f1_final1.append(lastiter1[x][3])
    f1_final2.append(lastiter2[x][3])
```

```
    rec_final1.append(lastiter1[x][2])
    rec_final2.append(lastiter2[x][2])
    prec_final1.append(lastiter1[x][1])
    prec_final2.append(lastiter2[x][1])
    acc_final1.append(lastiter1[x][4])
    acc_final2.append(lastiter2[x][4])
    predicted_val1[x] = lastiter1[x][0]
    predicted_val2[x] = lastiter2[x][0]
#%%
f1_scores = [f1_scores_m1,f1_scores_m2,f1_scores_m3,f1_final1,f1_final2]
recall_scores = [recall_scores_m1,recall_scores_m2,recall_scores_m3,rec_final1,
                 rec_final2]
precision_scores =[precision_scores_m1,precision_scores_m2,precision_scores_m3,
                   prec_final1,prec_final2]
accuracy_scores =[accuracy_m1,accuracy_m2,accuracy_m3,acc_final1,acc_final2]
#%%
df_F1 = pd.DataFrame(f1_scores,columns=alpha)
df_prec = pd.DataFrame(precision_scores,columns=alpha)
df_rec = pd.DataFrame(recall_scores,columns=alpha)
df_acc = pd.DataFrame(accuracy_scores,columns=alpha)
#%%
df_scores = pd.concat([df_F1,df_prec,df_rec,df_acc],axis=0,join='outer')
df_scores = df_scores.reset_index()
df_scores = df_scores.drop(df_scores.columns[0],axis=1)
print df_scores
#%%
df_scores.to_csv(r'paper_test(cv10).csv', header=True, index=True, mode='w',
                 encoding='utf-8')
```

Final results & plotting

```
#%%
import pandas as pd

#%%
df1 = pd.read_csv('paper_test(cv1).csv')
df2 = pd.read_csv('paper_test(cv2).csv')
df3 = pd.read_csv('paper_test(cv3).csv')
df4 = pd.read_csv('paper_test(cv4).csv')
df5 = pd.read_csv('paper_test(cv5).csv')
df6 = pd.read_csv('paper_test(cv6).csv')
df7 = pd.read_csv('paper_test(cv7).csv')
df8 = pd.read_csv('paper_test(cv8).csv')
df9 = pd.read_csv('paper_test(cv9).csv')
df10 = pd.read_csv('paper_test(cv10).csv')

#%%
df1.drop(df1.columns[[0]],inplace=True,axis=1)
df2.drop(df2.columns[[0]],inplace=True,axis=1)
df3.drop(df3.columns[[0]],inplace=True,axis=1)
df4.drop(df4.columns[[0]],inplace=True,axis=1)
df5.drop(df5.columns[[0]],inplace=True,axis=1)
df6.drop(df6.columns[[0]],inplace=True,axis=1)
df7.drop(df7.columns[[0]],inplace=True,axis=1)
df8.drop(df8.columns[[0]],inplace=True,axis=1)
df9.drop(df9.columns[[0]],inplace=True,axis=1)
```

```
df10.drop(df10.columns[[0]],inplace=True,axis=1)

#%%
sum_df = df1.add(df2.add(df3.add(df4.add(df5.add(df6.add(df7.add(df8.add(df9.add(df10)))))))))
avg_df = sum_df.divide(10)
avg_df.to_csv(r'avg_results.csv', header=True, index=None, mode='w', encoding='utf-8')

#%%
round_df = avg_df.round(2)
round_df.to_excel(r'round_results.xlsx', header=True, index=None, encoding='utf-8')
```

```
#%%
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

plt.style.use('ggplot')

#%%
df = pd.read_csv("avg_results.csv")
#df.drop(df.columns[[0]],inplace=True,axis=1)

#%%
alpha = [5,15,25,35,45,55,65]
f1_scores = df.iloc[np.arange(0,5,1),:]
accuracy_scores = df.loc[np.arange(15,20,1),:]
precision_scores =df.iloc[np.arange(5,10,1),:]
recall_scores = df.iloc[np.arange(10,15,1),:]

#%%
f1_scores = f1_scores.values.tolist()
accuracy_scores = accuracy_scores.values.tolist()
precision_scores =precision_scores.values.tolist()
recall_scores = recall_scores.values.tolist()

#%%
colors = ['#7FEBFF','#0072CC','#43A9FD','#7FEBFF','#0072CC']
markers = ['s','o','D','s','o']
lines = ['--','--','-','-','-']
def plot_scores(methods,alpha,scores,i,line,marks,cols,title = 'Score'):
    plt.figure(figsize=(8,7))
    plt.title('{0} scores for GOP data (Positive = labelled)'.format(title),
              fontsize='xx-large')
    plt.xlabel('Alpha (%)',fontsize='large')
    plt.ylabel('{0} score'.format(title),fontsize='large')
    for x in xrange(0,i):
        plt.plot(alpha,scores[x],marker=marks[x],color=cols[x],linestyle=line[x],
                 label=methods[x],markersize=7.5,linewidth=2.0)
        plt.legend(loc=4,fontsize='large')
        plt.xlim([0,70])
        plt.ylim([0,1.0])
        plt.tick_params(labelsize='medium')
        plt.xticks(alpha)
        plt.grid(True)
        plt.savefig('{0}_P1.png'.format(title),dpi=100)
```

```
    plt.show()

#%%
methods = ['Method 1: Intital SVM','Method 2: Initial SVM','Model: Logistic regression',
           'Method 1: Final SVM','Method 2: Final SVM']

plot_scores(methods,alpha,f1_scores,len(methods),lines,markers,colors,title='F1')
plot_scores(methods,alpha,accuracy_scores,len(methods),lines,markers,colors,title='Accuracy')
plot_scores(methods,alpha,precision_scores,len(methods),lines,markers,colors,title='Precision')
plot_scores(methods,alpha,recall_scores,len(methods),lines,markers,colors,title='Recall')
```

Results

20 Newsgroup data

	α	**5%**	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**
Method	Labelled observations	49	149	249	348	448	548	648
Rocchio	Initial SVM	0,06	0,35	0,57	0,65	0,68	0,7	0,69
	Final SVM	0,04	0,24	0,53	0,65	0,68	0,7	0,69
Supervised	Logistic Regression	0,05	0,2	0,33	0,42	0,46	0,48	0,46
Rocchio w/ clustering	Initial SVM	0,11	0,74	0,9	0,91	0,92	0,92	0,9
	Final SVM	0,04	0,27	0,7	0,9	0,94	0,94	0,93

(a) F1 scores

	α	**5%**	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**
Method	Labelled observations	49	149	249	348	448	548	648
Rocchio	Initial SVM	0,03	0,21	0,4	0,49	0,52	0,54	0,53
	Final SVM	0,02	0,14	0,36	0,48	0,52	0,54	0,53
Supervised	Logistic Regression	0,03	0,11	0,2	0,26	0,3	0,31	0,3
Rocchio w/ clustering	Initial SVM	0,06	0,61	0,86	0,89	0,94	0,94	0,96
	Final SVM	0,02	0,15	0,54	0,83	0,91	0,94	0,96

(b) Recall

	α	**5%**	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**
Method	Labelled observations	49	149	249	348	448	548	648
Rocchio	Initial SVM	0,99	0,99	0,99	0,99	0,99	0,99	0,98
	Final SVM	1	0,99	0,99	0,99	0,99	0,99	0,98
Supervised	Logistic Regression	0,99	0,99	1	1	1	1	0,99
Rocchio w/ clustering	Initial SVM	0,99	0,97	0,95	0,93	0,9	0,89	0,85
	Final SVM	1	0,99	0,99	0,98	0,97	0,94	0,91

(c) Precision

	α	**5%**	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**
Method	Labelled observations	49	149	249	348	448	548	648
Rocchio	Initial SVM	0,46	0,56	0,66	0,71	0,73	0,74	0,74
	Final SVM	0,46	0,52	0,64	0,71	0,73	0,74	0,74
Supervised	Logistic Regression	0,46	0,51	0,56	0,59	0,61	0,62	0,61
Rocchio w/ clustering	Initial SVM	0,48	0,77	0,9	0,91	0,91	0,9	0,88
	Final SVM	0,46	0,53	0,74	0,9	0,94	0,94	0,92

(d) Accuracy

Table 4: Results: 20 Newsgroups data

GOP data with positive tweets as labelled data

	α	**5%**	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**
Method	Labelled observations	83	251	418	586	753	921	1088
Rocchio	Initial SVM	0,05	0,25	0,39	0,46	0,51	0,54	0,59
	Final SVM	0,03	0,18	0,32	0,42	0,48	0,54	0,59
Supervised	Logistic Regression	0,06	0,21	0,31	0,39	0,45	0,5	0,57
Rocchio w/ clustering	Initial SVM	0,09	0,39	0,46	0,47	0,46	0,49	0,49
	Final SVM	0,03	0,18	0,34	0,43	0,49	0,53	0,56

(a) F1 scores

	α	**5%**	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**
Method	Labelled observations	83	251	418	586	753	921	1088
Rocchio	Initial SVM	0,03	0,16	0,29	0,4	0,51	0,61	0,71
	Final SVM	0,02	0,11	0,23	0,34	0,45	0,58	0,7
Supervised	Logistic Regression	0,03	0,12	0,21	0,28	0,35	0,41	0,51
Rocchio w/ clustering	Initial SVM	0,05	0,33	0,57	0,69	0,77	0,8	0,86
	Final SVM	0,02	0,11	0,25	0,38	0,51	0,66	0,79

(b) Recall

	α	**5%**	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**
Method	Labelled observations	83	251	418	586	753	921	1088
Rocchio	Initial SVM	0,68	0,65	0,59	0,55	0,52	0,49	0,5
	Final SVM	0,63	0,59	0,56	0,54	0,52	0,5	0,51
Supervised	Logistic Regression	0,69	0,67	0,64	0,63	0,64	0,63	0,64
Rocchio w/ clustering	Initial SVM	0,64	0,48	0,39	0,35	0,33	0,35	0,34
	Final SVM	0,61	0,56	0,53	0,5	0,48	0,44	0,44

(c) Precision

	α	**5%**	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**
Method	Labelled observations	83	251	418	586	753	921	1088
Rocchio	Initial SVM	0,79	0,8	0,8	0,8	0,79	0,78	0,79
	Final SVM	0,79	0,79	0,79	0,79	0,79	0,78	0,79
Supervised	Logistic Regression	0,79	0,8	0,8	0,81	0,82	0,82	0,83
Rocchio w/ clustering	Initial SVM	0,79	0,78	0,71	0,66	0,62	0,64	0,61
	Final SVM	0,78	0,79	0,79	0,78	0,77	0,75	0,74

(d) Accuracy

Table 5: Results: GOP data (positive tweets as labelled)

GOP data with negative tweets as labelled data

	α	**5%**	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**
Method	Labelled observations	303	910	1517	2124	2731	3338	3945
Rocchio	Initial SVM	0,11	0,36	0,47	0,54	0,58	0,64	0,71
	Final SVM	0,08	0,34	0,46	0,53	0,58	0,63	0,7
Supervised	Logistic Regression	0,08	0,22	0,29	0,33	0,35	0,35	0,35
Rocchio w/ clustering	Initial SVM	0,42	0,77	0,8	0,82	0,84	0,85	0,87
	Final SVM	0,09	0,42	0,65	0,75	0,81	0,84	0,86

(a) F1 scores

	α	**5%**	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**
Method	Labelled observations	303	910	1517	2124	2731	3338	3945
Rocchio	Initial SVM	0,06	0,23	0,32	0,38	0,43	0,49	0,57
	Final SVM	0,04	0,21	0,31	0,38	0,43	0,49	0,57
Supervised	Logistic Regression	0,04	0,12	0,17	0,2	0,21	0,22	0,21
Rocchio w/ clustering	Initial SVM	0,28	0,71	0,79	0,8	0,85	0,86	0,89
	Final SVM	0,05	0,28	0,52	0,66	0,76	0,83	0,87

(b) Recall

	α	**5%**	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**
Method	Labelled observations	303	910	1517	2124	2731	3338	3945
Rocchio	Initial SVM	0,87	0,88	0,89	0,9	0,91	0,91	0,93
	Final SVM	0,86	0,87	0,89	0,9	0,91	0,91	0,93
Supervised	Logistic Regression	0,88	0,9	0,91	0,92	0,92	0,93	0,94
Rocchio w/ clustering	Initial SVM	0,84	0,83	0,83	0,83	0,83	0,84	0,85
	Final SVM	0,86	0,87	0,88	0,87	0,86	0,86	0,85

(c) Precision

	α	**5%**	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**
Method	Labelled observations	303	910	1517	2124	2731	3338	3945
Rocchio	Initial SVM	0,26	0,37	0,44	0,48	0,52	0,56	0,63
	Final SVM	0,24	0,36	0,43	0,48	0,52	0,56	0,63
Supervised	Logistic Regression	0,25	0,3	0,34	0,36	0,37	0,37	0,37
Rocchio w/ clustering	Initial SVM	0,4	0,66	0,7	0,72	0,75	0,76	0,78
	Final SVM	0,25	0,4	0,57	0,66	0,72	0,76	0,78

(d) Accuracy

Table 6: Results: GOP data (negative tweets as labelled)

Colour recognition for image analysis

Tinashe Hanyani 12207463

WST795 Research Report

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

Supervisor: Dr I Fabris-Rotelli

Department of Statistics, University of Pretoria

2 November 2016 (Final document)

Abstract

This research report looks at how RGB colour histograms are used for colour recognition and they can be applied to fields such as robotics. We will proceed by giving an example of how colour histograms function for colour recognition. A write up of the Kalman filter for localization will be included as well as an application of how colour histograms can be used to classify images based on colour with the aid of histogram distances for image comparison.

Declaration

I, *Tinashe Brian Hanyani,* 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 Brian Hanyani

_ _

Dr I Fabris-Rotelli

_ _

2 November 2016

Contents

1 Introduction **5**

2 Literature Review **8**
2.1 Colour histograms . . . 8
2.2 The Kalman filter and application to robotics . . . 8

3 Background Theory **9**
3.1 The Kalman filter . . . 9
3.1.1 Kalman filter background . . . 9
3.1.2 The Kalman filter algorithm . . . 10
3.1.3 Examples . . . 11
3.2 Colour Histograms . . . 16
3.2.1 Definition . . . 16
3.2.2 Image comparison . . . 17
3.2.3 Histogram distances . . . 18

4 Application **19**
4.1 Colour recognition . . . 19
4.1.1 Results . . . 21

5 Conclusion **28**

Appendix **30**

List of Figures

1 The RGB Colour space . . . 5
2 Sample images . . . 6
3 RGB colour histogram of Image 2-Figure 2 . . . 7
4 The Kalman filter algorithm . . . 12
5 One dimensional illustration of Kalman filter, initial steps . . . 13
6 One dimensional illustration of Kalman filter, subsequent steps . . . 14
7 three time steps-Kalman filter . . . 15
8 1000 time steps-Kalman filter . . . 16
9 Image comparison algorithm . . . 17
10 Test images as well as sample images from the car database . . . 20
11 Test images as well as sample images from the clean database . . . 20

List of Tables

1 Red test images with car database . . . 22
2 Green test images with car database . . . 23
3 Blue test images with car database . . . 24
4 Red test images with clean database . . . 25
5 Green test images with clean database . . . 26
6 Blue test images with clean database . . . 27

1 Introduction

Have you ever wondered how cameras these days are able to detect bright colours or how robots can visualize their surroundings? The answer lies with colour histograms. A colour histogram of an image is defined as the colour density function of the image pixels in a given colour space [10] and denotes the joint probabilities of the intensities of the three colour channels, namely the RGB colour space shown in Figure 1[1]. A colour histogram is produced by counting the number of pixels in an image of each colour. It first does this by splitting the image into its red, green and blue colour levels then counts them individually in each level. A computer typically has 16,777,216 (256^3) colours which are made up by mixing the colours red, green and blue. In Figure 1, the RGB colour space is represented in cartesian coordinate system with red, green and blue on the major axes. All 256^3 colours are represented by a mixture of red, green and blue. For example in Figure 1, the colour cyan which has coordinates (0,255,255) is a mixture of blue and green with respective coordinates (0,0,255) and (0,255,0). You may also observe the diagonal from black (0,0,0) to white (255,255,255) which makes up the greyscale of an image or in other words, black and white.

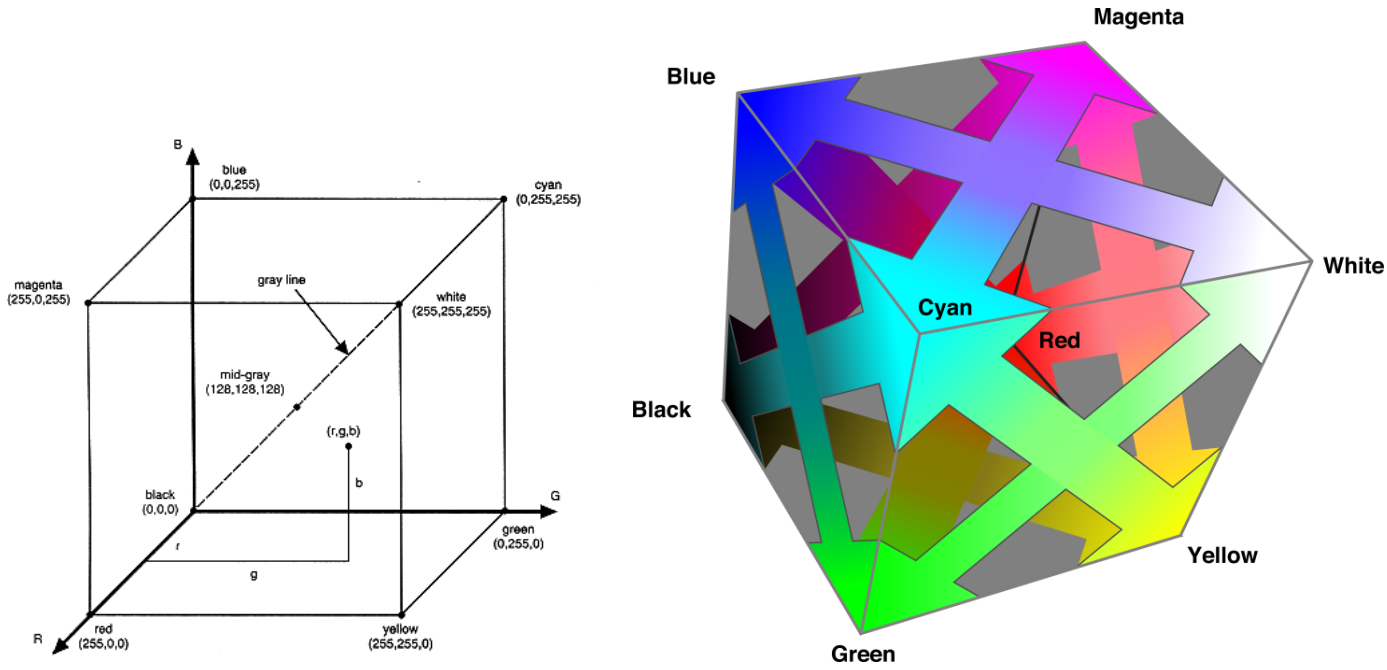


Figure 1: The RGB Colour space

Colour recognition is a good example of how colour histograms can be used. Consider the four paintings in Figure 2[2]. Firstly one would choose a query image. Secondly choose a colour space, in this case the RGB model. Then matching of colour histograms done through various measurements. Another model available is the HSV (Hue, Saturation and Value) colour space [12], but for purposes of this report, we will concentrate on the RGB colour space. It is also mentioned in [6, 12] how to interchange between the colour spaces mentioned. This is achieved by using the Euclidean distance approach.

[1] Figure 1 downloaded from http://www.viz.tamu.edu and https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/csintro/csintro_colorspace/csintro_colorspace.html

[2] Images in Figure 2 downloaded from:
Image 1- https://www.google.com/culturalinstitute/beta/asset/bgEuwDxel93-Pg?utm_source=google&utm_medium=kp&projectId=art-project,
Image 2- http://www.paintinghere.com/painting/vincent_van_gogh_branches_of_an_almond_tree_in_blossom_in_red_27683.html,
Image 3- http://www.vggallery.com/painting/p_0611.html and
Image 4 https://www.khanacademy.org/humanities/becoming-modern/symbolism/a/munch-the-scream

An image retrieval algorithm, such as [12, 6, 5], proceeds to quantize or transform the image. This helps in reducing computational time since it would take some time for comparison of 256*256*256 bins to be compared. In [6] the transformation is from 256*256*256 bins into 8*8*8 bins. The algorithm proceeds to compute the histogram which it does by merely counting the number of pixels of each colour in the image and plots them as frequencies on a histogram. For comparison purposes proportion instead of frequencies are used in order to make comparison of images of different size possible. The details of this procedure will be expanded on in the theory section of this report. Now to compare images, distance formulas are used for example histogram euclidean distance and histogram intersection[6, 5]. Results are then displayed of potential matches i.e images with the shortest distance. Referring back to the images in Figure 2, one would expect the image with the most red (Image 1) to have a peak on the red histogram, and similarly the paintings with the green (Image 2) and the blue (Image 3) will have peaks on the red and green bins respectively. One would expect to observe a unimodal histogram. Image 4 has a bit of blue, green and some red. So one would most likely observe a peak at all these three colours, not as pronounced as the other histograms however.

In Figure 3 we observe the colour histograms computed by MATLAB®[3] for Image 4. A histogram with 51 bins and one with 8 bins was computed for each colour level of Image 4. For each histogram, the y-axis is labelled the proportion of pixels so as to account for the different number of pixels in each image.

Image 1

Image 2

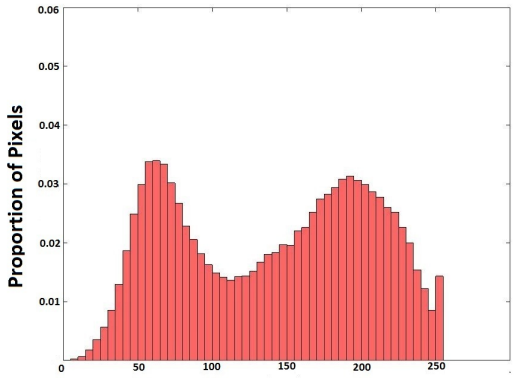
Image 3

Image 4

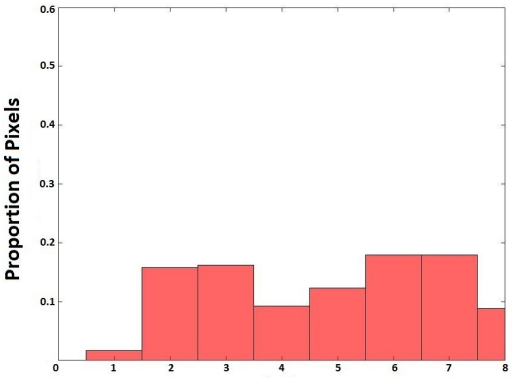
Figure 2: Sample images

Various applications of colour histograms can also be found in the field of robotics. For example the Robocup tournament [2] where uniform colour goals are now used and also how robots are able to recognize colours. In this report, we will focus on how colour histograms are used to compare images based on colour using a similar approach to the image retrieval algorithm [12, 6, 5] and how this can be applied to fields such as robotics, for example in a localization problem with the use of the Kalman filter [13, 7, 11, 18].

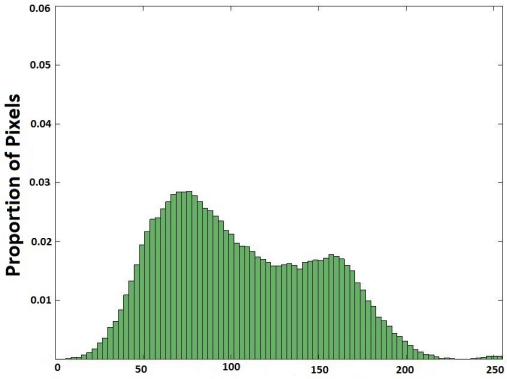
[3] http://www.mathworks.com/products/matlab/



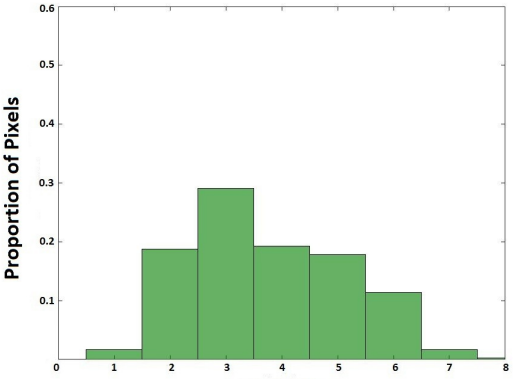
(a) Red - 51 bins



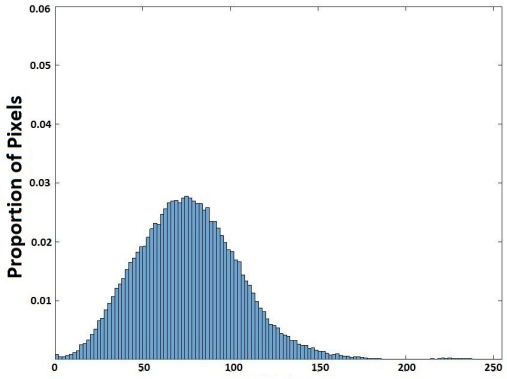
(d) Red - 8 bins



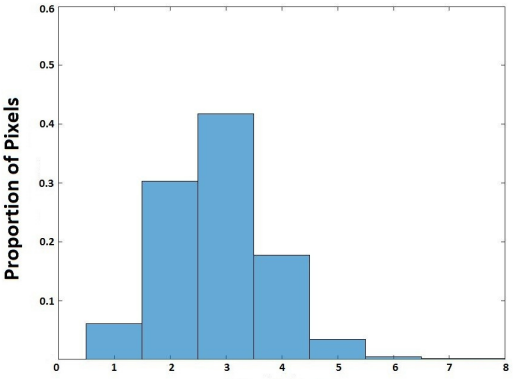
(b) Green - 51 bins



(e) Green - 8 bins



(c) Blue - 51 bins



(f) Blue - 8 bins

Figure 3: The RGB levels of Image 4 in Figure 2. (a)-(c) 51 bins (d)-(f) 8 bins

2 Literature Review

2.1 Colour histograms

Statistics for robotics: Colour Classification [8], includes a short and very brief practical approach to colour recognition with R®[4]. This paper is from a workshop in 2015 at the University of Pretoria and illustrates how a robot can be used to identify objects based on thier colour, in a colour coded environment.

Histogram-Based Color Image Retrieval [6], contains a detailed approach to image retrieval. The article focuses on six histogram based retrieval methods in the two colour spaces namely the RGB and HSV. They begin by defining a colour space as a model which represents colour intensities and a colour space defines a one to four dimensional space. They go on to define the RGB and HSV colour spaces.

Jeong [6] goes on to describe and give a comprehensive illustration of the histogram-based image retrieval method which has been mentioned earlier in the introduction of this report and conclude that the HSV colour model outperforms the RGB colour model and further mentions that the histogram intersection-based image retrieval in HSV colour space performs when taking into account computational time and retrieval performance.

Colour histogram based retrieval [5], provides a more theoretical approach to colour histogram compared to the previous paper [6]. Hussian et al [5] provide a detailed image retrieval using colour histograms.

2.2 The Kalman filter and application to robotics

Probabilistic Robotics [15] includes a broad introduction to the field of robotics. It focuses on the statistical techniques employed in robotics, for example, Gaussian filters namely the Kalman filter. The Kalman filter algorithm is explained in detail and is followed by an illustrative example. The Kalman filter relies on the linearity assumption, however Thrun et al [15] discuss another version of the Kalman filter known as the Extended Kalman filter (EKF) which does not assume linearity. The EKF is more popular than the Kalman filter as a tool for state estimation in robot localisation, however a limitation exhibited by the EKF is that it approximates state transitions using linear Taylor expansions whereas most robotics applications are nonlinear. Other versions of the Kalman filter discussed include the Unscented Kalman filter which uses a stochastic method for linearization, the information filter and lastly the extended information filter. Thrun et al [15] also include nonparametric filters such as the particle filter which represents a distribution by a set of samples drawn from this distribution. A particle is a hypothesis as to what the true world state may be at time t. Unlike the Kalman filter which uses underlying Gaussian assumption, the particle filter includes a wider range of distributions.

Kalman filtering in R® [16], provides a detailed comparison of different R® packages used for computing the Kalman filter. It also includes a detailed write up of the Kalman filter.

Statistics for robotics : Kalman filter [9], is from a workshop done in 2015 at the University of Pretoria. It gives a short and brief description as well as a write up of the Kalman filter. The algorithm carried out is done using the R® software package.

Colour Histograms as Background Description: An approach to overcoming the Uniform-Goal problem within the SPL for the RoboCup WC 2012 [2], focuses on a possible solution in order tackle the recent change in rules to the RoboCup, that now states that each team should score in their own unique goal. This is a challenge since the robots allowed to compete in the RoboCup competition do not have a GPS system. This report proposes the use of visual background as an aid to robot localization, in order to overcome the challenging uniform goal problem. They achieve this by analyzing colour histograms and how they may be used for this particular problem.

On the application of colour histograms for mobile robot localization [10], illustrates an appearance-based method to be used for topologically localising a robot. They use a non-parametric clustering paradigm, a self-organising map neural network as well as information obtained from segmentation of a single image to approximate a colour probability density function.

[4] `https://www.r-project.org/`

A self-organizing map (SOM) network approximates the colour probability distribution thus the colour histograms of the image in the RGB space and retains the topological information and three dimensional distribution of the data.
To explain how the histograms are approximated, consider M images collected while the robot is navigating a path. At time t, the robot stores q previous images denoted $I_{t-1}, I_{t-2}, ..., I_{t-q}$ and the latest image is I_t. The entire image is not used however a smaller section of the image is obtained and is known as the region of interest (ROI). A ratio r_{t-j} of the image is obtained such that the most recent images have a greater weighting than the other images.
Now at time t the trained linear SOM network, using the information from the images, approximates the colour histograms, utilizing the weights and distribution of units along the RGB space as well as a segmentation procedure for the current image. Segmentation is the process where the image is separated into the red, green and blue components of the corresponding image. The histogram is now projected through the neural network weights defined as $(\omega_i^r(t), \omega_i^g(t), \omega_i^b(t))$ with index i denoting the bin and super-script for the colours. The network goes on to segment the image I_t using the neural network weights. Hence three histograms are obtained each representing the different colour components of the RGB space.
Repeating the process through all M images, a collection of elements is obtained:

$$\{\upsilon_1(t), \upsilon_2(t), ..., \upsilon_s(t)\}$$

from $t = 0$ to M where $\upsilon_j(t) = \{(\omega_j^r(t), \omega_j^g(t), \omega_j^b(t))\}$ represents the relative frequencies of the $j-th$ bin of the RGB component with $j = 1$ to K and M is the number of images in the sequence.
Ranó et al [10] propose a number of histogram distances in order to compare histograms obtained at each time step with previously stored histograms.
The distances presented in this paper include the simple and the popular euclidean distance, the Manhattan distance, histogram intersection, chi-squared distance [10] , correlation distance [14] and finally the Bhattacharyya measure [4].

3 Background Theory

3.1 The Kalman filter

3.1.1 Kalman filter background

The Kalman filter was first introduced by Swerling [13] and Kalman [7] as a technique for filtering and prediction in linear Gaussian systems. Gaussian filters all share the same concept that the beliefs follow multivariate normal distributions. The underlying distributions of the Kalman filter are multivariate normal. The beliefs of the Kalman filter are denoted at time t by $(bel(\mathbf{x_t}))$ with mean μ_t and covariance $\sum_t$. It is simply a linear weighted average of two sensor values. It uses a combination of measurements from the same variable from different sensors namely an approximation or prediction of the system's state denoted as γ_t with an approximate measurement of the state denoted as z_t .
It has various uses in aeronautics, engineering and statistics. More specifically in navigation, guidance, radar tracking and stock price prediction [11, 18].

Let random variables X and Y be continuous random variables. If x is a value that we would like to estimate using y then the density function $p(x)$ is known as the prior distribution which is defined as

$$p(x) = \int p(x|y)p(y).$$

This distribution is the information we have with regards to X prior to incorporating Y. If we now incorporate Y we get the density function $p(x|y)$ known as the posterior distribution and is defined as

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}.$$

Now we an apply these defintions to the Kalman filter since they an important role in the algorithm.
In order for the posterior denoted as $bel(\mathbf{x_t})$ of the Kalman filter to be Gaussian, the following three properties must hold:

1. The state transition probability $p(\mathbf{x_t}|\gamma_\mathbf{t}, \mathbf{x_{t-1}})$ known as the posterior state since the density of $\mathbf{x_t}$ is now calculated using the information from the previous time step $\mathbf{x_{t-1}}$ and control information at time t, γ_t, must be a linear function with Gaussian error and is expressed by the following equation:

$$\mathbf{x_t}{=}A_t\mathbf{x_{t-1}} + B_t\gamma_\mathbf{t} + \varepsilon_\mathbf{t}$$

with intial prior knowledge $\mathbf{x_0} \sim N(\mu_\mathbf{0}, \sum_\mathbf{0})$, where $\mathbf{x_t}$ and $\mathbf{x_{t-1}}$ are $n \times 1$ state transition vector at time t and $t-1$ respectively, γ_t is an $n \times 1$ control vector at time t, A_t is a sqaure matrix of dimension n, B_t is an $n \times m$ matrix and $\varepsilon_\mathbf{t}$ is an $n \times 1$ Gaussian random vector which models the uncertainty brought about by the state transititon with mean zero and covariance matrix R_t i.e $\varepsilon_\mathbf{t} \sim N(\mathbf{0}, R_t)$. The density of the posterior is given by

$$p(\mathbf{x_t}|\gamma_\mathbf{t}, \mathbf{x_{t-1}}) = |2\pi R_t|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\mathbf{x_t} - \mathbf{A_t x_{t-1}} - B_t\gamma_\mathbf{t})' R_t^{-1}(\mathbf{x_t} - A_t\mathbf{x_{t-1}} - B_t\gamma_\mathbf{t})\}$$

which is a normal distribution with mean $A_t\mathbf{x_{t-1}} + B_t\gamma_\mathbf{t}$ and covariance matrix R_t.

2. The measurement probability $p(\mathbf{z_t}|\mathbf{x_t})$ which is also a posterior since the density of $\mathbf{z_t}$ is obtained by incorporating known information namely $\mathbf{x_t}$, must also be a linear function with a Gaussian error, that is,

$$\mathbf{z_t} = \mathbf{G_t x_t} + \delta_\mathbf{t}$$

where $\mathbf{z_t}$ is a $k \times 1$ measurement vector (e.g input from sensors, GPS) at time t , G_t is a $k \times n$ matrix and $\delta_\mathbf{t}$ is a $k \times 1$ Gaussian random vector which represents the measurement uncertainty with mean zero and covariance matrix Q_t i.e $\delta_\mathbf{t} \sim N(\mathbf{0}, Q_t)$. The density of the posterior is given by

$$p(\mathbf{z_t}|\mathbf{x_t}) = |2\pi Q_t|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\mathbf{z_t} - C_t\mathbf{x_t})' Q_t^{-1}(\mathbf{z_t} - C_t\mathbf{x_t})\}.$$

3. The initial belief $bel(\mathbf{x_0})$ must be normally distributed i.e $\mathbf{x_o} \sim N(\mu_\mathbf{0}, \sum_0)$.

3.1.2 The Kalman filter algorithm

The Kalman filter algorithm represents the belief $bel(\mathbf{x_t})$ at time t with mean $\mu_\mathbf{t}$ and the covariance $\sum_t$. The input parameters of the Kalman filter is the belief at time $t-1$ and is denoted with parameters μ_{t-1} and $\sum_{t-1}$. Also $\gamma_\mathbf{t}$ and $\mathbf{z_t}$ are defined above as the control and measurement vectors respectively and are required in order to update the input parameters and give the output parameters $\mu_\mathbf{t}$ and $\sum_t$, the belief parameters at time t .
The algorithm of the Kalman filter is as follows:

Input$\left(\mu_\mathbf{t-1}, \sum_\mathbf{t-1}\right)$
Now using information obtained at time $t-1$, namely μ_{t-1} and $\sum_{t-1}$, where μ_{t-1} is defined as the belief $bel(\mathbf{x_t})$ at time $t-1$ with covariance matrix $\sum_{t-1}$.

1. $\bar{\mu}_t = A_t\mu_{\mathbf{t-1}} + B_t\gamma_{\mathbf{t}}$

   where $\bar{\mu}_t$ is the mean of the predicted belief $\bar{bel}(\mathbf{x_t})$ and is a linear function of $\mu_{\mathbf{t-1}}$ and $\gamma_{\mathbf{t}}$.

2. $\bar{\sum}_{\mathbf{t}} = A_t \sum_{t-1} A_t^{'} + R_t$

   where $\bar{\sum}$ is the covariance of $\bar{bel}(x_t)$ which is a function of $\sum_{t-1}$ and R_t. Hence $\bar{bel}(x_t) \sim N(\bar{\mu}_t, \bar{\sum}_t)$.

3. $K_t = \bar{\sum}_t G_t^{'}(G_t \bar{\sum}_t G_t^{'} + Q_t)^{-1}$

   K_t is known as the Kalman gain.

4. $\mu_{\mathbf{t}} = \bar{\mu}_{\mathbf{t}} + K_t(\mathbf{z_t} - G_t\bar{\mu}_{\mathbf{t}})$

   where $\mu_{\mathbf{t}}$ is the mean of the belief $bel(\mathbf{x_t})$ and is a function of the mean predicted belief $\bar{\mu}_{\mathbf{t}}$, K_t and the measurement vector $\mathbf{z_t}$.

5. $\sum_{\mathbf{t}} = (I - K_t G_t)\bar{\sum}_t$

   where $\sum_t$ is the covariance of $bel(\mathbf{x_t})$.

Output $(\mu_{\mathbf{t}}, \sum_t)$

In steps 1 and 2 the predicted belief $\bar{\mu}_{\mathbf{t}}$ and $\bar{\sum}_t$ is computed using the control vector $\gamma_{\mathbf{t}}$. Then $\bar{\mu}_{\mathbf{t}}$ is computed using the state transition vector with $\mu_{\mathbf{t-1}}$ in place of $\mathbf{x_{t-1}}$ and $\bar{\sum}_t$ is computed using the fact that states are dependent on past states with linear matrix A_t. The first two steps of the Kalman filter algorithm are known as the control update or prediction steps. In step 3, K_t determines the extent to which $\mathbf{z_t}$ is incorporated into the new state estimate.
In steps 4 and 5, $\mu_{\mathbf{t}}$ and $\sum_t$ are obtained using $\overline{\mu}_t$, $\bar{\sum}_t$ as well as the measurement vector $\mathbf{z_t}$. Steps 3 to 5 are known as the measurement update steps. By using $\bar{bel}(\mathbf{x_t})$ these three steps help in refining the predicted values calculated in step 1 and 2 thus giving a better estimate of the mean which has a smaller .
The Kalman filter algorithm returns $\mu_{\mathbf{t}}$ and $\sum_t$ which are the belief parameters at time t denoted as $bel(\mathbf{x_t})$ [15].
In Figure 4, the Kalman filter algorithm is summarized. The mathematics behind the Kalman filter can be a bit complex to understand at first, however Figures 5 and 6 provide a one dimensional representation of the Kalman filter as well as a numerical example with some graphs included for visual illustration.

3.1.3 Examples

Figure 5(a) shows the initial belief ($bel(\mathbf{x_0})$) or the prior of the robot and is given as a normal distribution. In Figure 5(b), the robot analyses its sensor input for example, GPS system or infrared, and returns a measurement of its position which is found at the peak of Figure 5(b) which is approximated by the sensor input and the spread corresponds to the uncertainty of the measurement. Now in Figure 5(c) the dashed

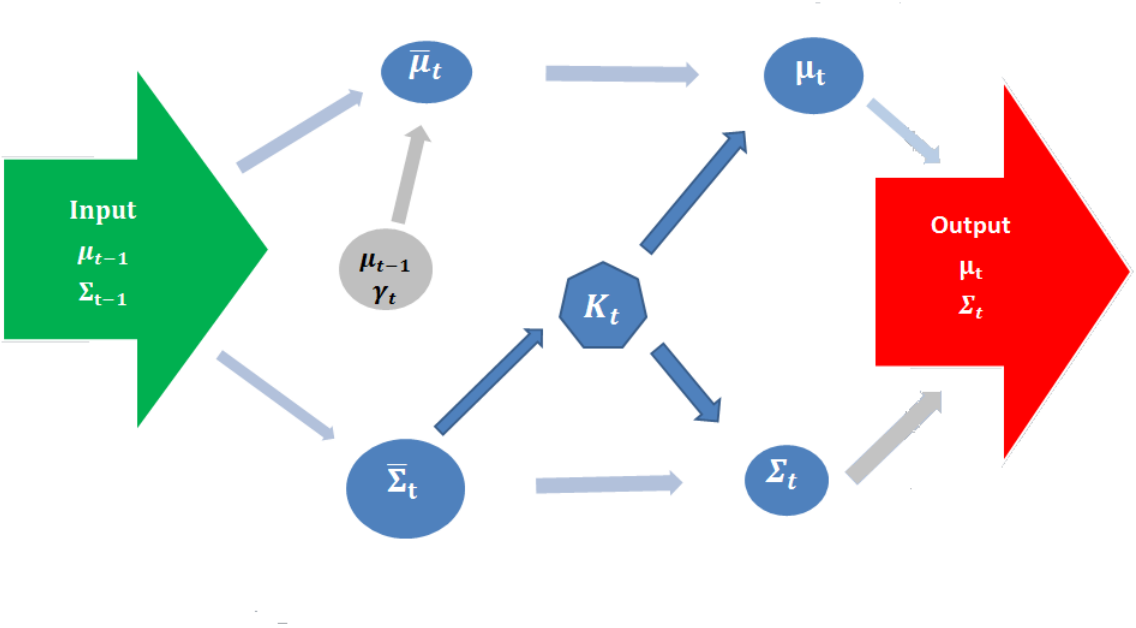


Figure 4: The Kalman filter algorithm

blue line is as a result of steps 4 and 5 of the Kalman filter algorithm. The mean of the belief at time $t = 1$ is found between the two previous means and the uncertainty of the robot's position is smaller.

Now, suppose the robot moves to the right. The Gaussian shown in bold red in Figure 6(a) is a result of steps 2 and 3 which compute $\bar{\mu}_2$ and $\bar{\sum}_2$ of the Kalman filter, namely the prediction steps. It is shifted by the magnitude that the robot has moved. It has a wider spread compared to the belief at $t = 1$ due to the increased uncertainty brought about by the stochastic nature of the state transtition. A second measurement is received by the robot which results in the Gaussian in Figure 6(b) shown as the bold blue line. Once again this leads to lines 3 to 5 of the Kalman filter where μ_2 and $\sum_2$ are computed resulting in the Gaussian in Figure 6(c) represented by the dashed blue line.

This example perfectly illustrates that steps 3 to 5 of the Kalman filter algorithm decreases the uncertainty as is seen in Figures 5(c) and 6(c), whereas steps 1 and 2 increase the uncertainty of the robots position as illustrated in Figure 6(b).

We now provide an example in two dimensions.

Initial starting values at time $t = 0$:

$$\mu_0 = \begin{pmatrix} 0.2 \\ -0.2 \end{pmatrix}$$

$$\sum_0 = \begin{pmatrix} 0.4 & 0.3 \\ 0.3 & 0.45 \end{pmatrix}$$

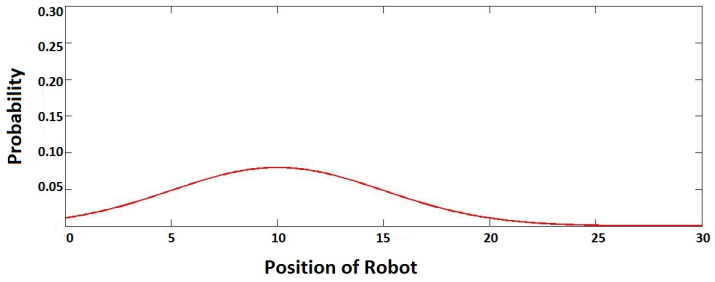
$$A_t = A = \begin{pmatrix} 0.4 & 0.5 \\ 0.2 & 0.3 \end{pmatrix}$$

$$B_t = B = \begin{pmatrix} 0.1 & 0.5 \\ 0.2 & 0.1 \end{pmatrix} \quad \gamma_t = \gamma = \begin{pmatrix} 0.4 \\ 0.2 \end{pmatrix}$$

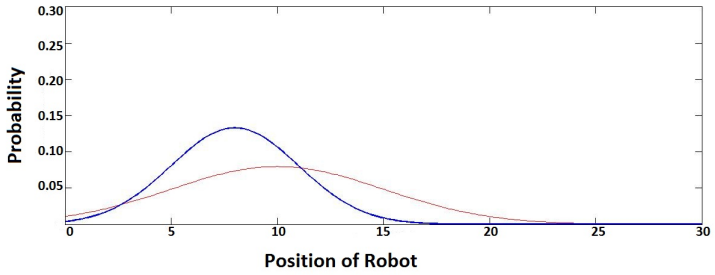
$$R_t = R = \begin{pmatrix} 0.5 & 0.2 \\ 0.2 & 0.1 \end{pmatrix} \quad G_t = G = \begin{pmatrix} 0.1 & 0.4 \\ 0.3 & 0.6 \end{pmatrix}$$

$$Q_t = Q = \begin{pmatrix} 0.4 & 0.1 \\ 0.1 & 0.2 \end{pmatrix}$$

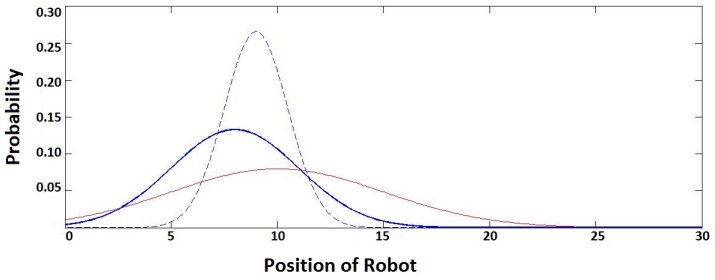
and z_t is a random vector generated from $N(0, I)$ distribution for each time step. All the values defined above have been randomly chosen. The example is to illustrate how the Kalman filter works. This is a simple



(a) Intial belief of robot at $t = 0$



(b) Position of robot after measurement step at $t = 1$.



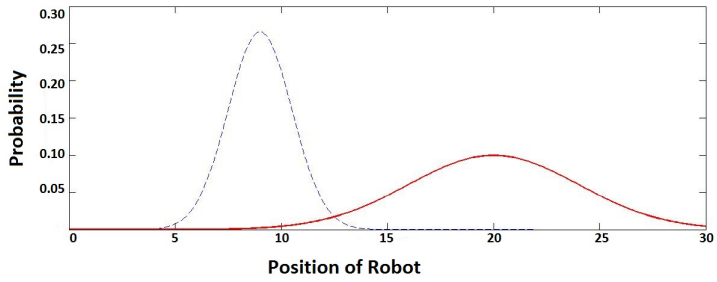
(c) Belief of robot at $t = 0$. Notice how the uncertainty at this point is reduced.

Figure 5: One dimensional illustration of Kalman filter, initial steps

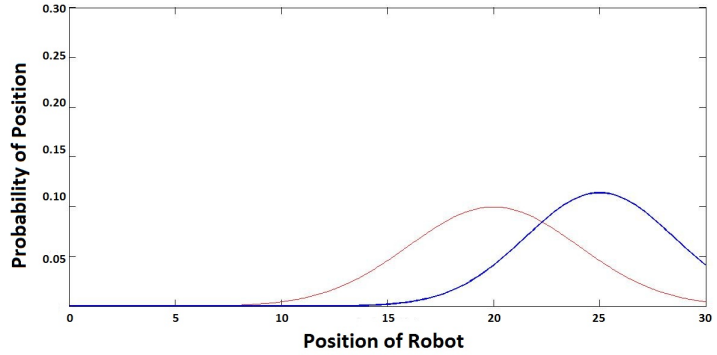
example since we assumed A_t, B_t, R_t, G_t and Q_t to be constant. The algorithm can also be adjusted in order to be time non-homogenous.

From $t = 1$ to $t = 2$
At $t = 1$:

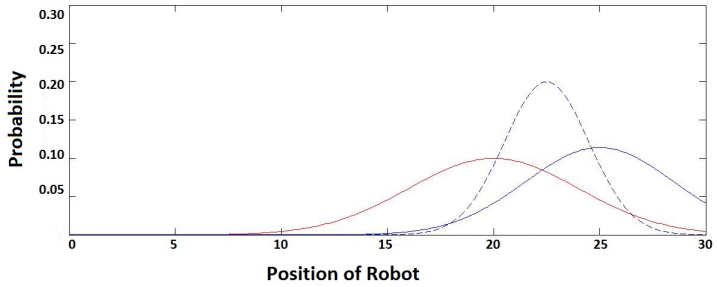
1. $\bar{\mu}_{\mathbf{1}} = A\mu_{\mathbf{0}} + B\gamma = \begin{pmatrix} 0.12 \\ 0.08 \end{pmatrix}$
2. $\bar{\sum}_1 = A\sum_0 A^T + R = \begin{pmatrix} 0.7965 & 0.3655 \\ 0.3655 & 0.1925 \end{pmatrix}$ since this the prediction step, it can be seen that the covariance matrix in the step above increases the uncertainty given the inital covariance structure
3. $K_1 = \bar{\sum}_1 G^T \left(G\bar{\sum}_1 G^T + Q\right)^{-1} = \begin{pmatrix} -0.0082 & 0.9738 \\ 0.0033 & 0.4748 \end{pmatrix}$
4. $\mu_{\mathbf{1}} = \bar{\mu}_{\mathbf{1}} + K_1(\mathbf{z_1} - G\bar{\mu}_{\mathbf{1}}) = \begin{pmatrix} 2.4550 \\ 1.2116 \end{pmatrix}$



(a) Position of robot after prediction steps at $t = 2$.



(b) Position of robot after measurement at $t = 2$.



(c) Belief f robot at $t = 2$. Again the uncertainty is reduced from the two previous steps.

Figure 6: One dimensional illustration of Kalman filter, subsequent steps

5. $\sum_1 = (I - K_1 G)\bar{\sum}_1 = \begin{pmatrix} 0.3521 & 0.1472 \\ 0.1472 & 0.0852 \end{pmatrix}$

The uncertainty in step 2 which is represented as covariance matrix $\bar{\sum}_t$ is reduced in step 5. The uncertainty in step 5 is also smaller than the initial uncertainty given by $\bar{\sum}_0$. Hence it is evident that the step 5 of the Kalman filter otherwise known as part of the measurement update step decreases uncertainty. This helps to explain the function that the Kalman filter serves in decresing the uncertainty of the prediction steps by using the measurement update.

At $t = 2$:

1. $\bar{\mu}_{\mathbf{2}} = A\mu_{\mathbf{1}} + B\gamma = \begin{pmatrix} 1.7278 \\ 0.9545 \end{pmatrix}$

2. $\bar{\sum}_2 = A \sum_1 A^T + R = \begin{pmatrix} 0.6365 & 0.2733 \\ 0.2733 & 0.1394 \end{pmatrix}$

3. $K_2 = \bar{\sum}_2 G^T \left(G \bar{\sum}_2 G^T + Q\right)^{-1} = \begin{pmatrix} -0.0099 & 0.8794 \\ 0.0022 & 0.4071 \end{pmatrix}$

4. $\mu_{\mathbf{2}} = \bar{\mu}_{\mathbf{2}} + K_2(\mathbf{z_2} - G\bar{\mu}_{\mathbf{2}}) = \begin{pmatrix} 1.2096 \\ 0.7159 \end{pmatrix}$

5. $\sum_2 = (I - K_2 G)\bar{\sum}_2 = \begin{pmatrix} 0.3261 & 0.1285 \\ 0.1285 & 0.0718 \end{pmatrix}$

The predicted uncertainty represented by covariance matrix $\bar{\sum}_2$ calculated in step 2 of the Kalman filter algorithm at time (which is a function of $\sum_1$) has a larger uncertainty compared to $\sum_1$. This is due to the fact that the robot has not receieved any new information of its location in order to better approximate its position. It only has previous information on which to rely on. Once the robot has receieved measurement update then only can it go through steps 3 to 5 of the Kalman filtter in order to reduce the uncertainty of its position and so on .

Figures 7 and 8 represent the measurement vectors $\mathbf{z_t}$ (red crosses) as well as the $\mu_{\mathbf{t}}$ vectors (blue stars). Figure 7 represents three time steps and Figure 8 represents 1000 time steps. In Figure 7, the points labelled $t = 1$ and $t = 2$ are plotted using the values of $\mu_{\mathbf{1}}$ and $\mu_{\mathbf{2}}$ respectively in the example above. The algebra for $t = 3$ is not shown in example but is calculated in the same manner. Time $t = 0$ was a randomly chosen starting point.

In Figure 8, similar steps shown in the example above are carried out one thousand times. This plot shows a graphical representation of what the Kalman filter does. It can be seen that the Kalman filter processes the random measurement vectors and filters them and decreases the uncertainty brought about by the measurement vectors.

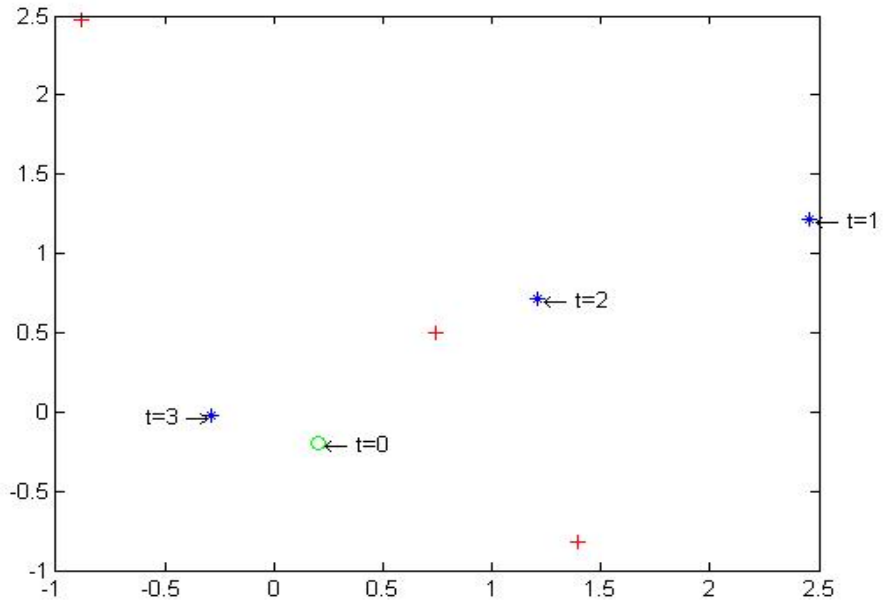


Figure 7: Three time steps of the Kalman filter example

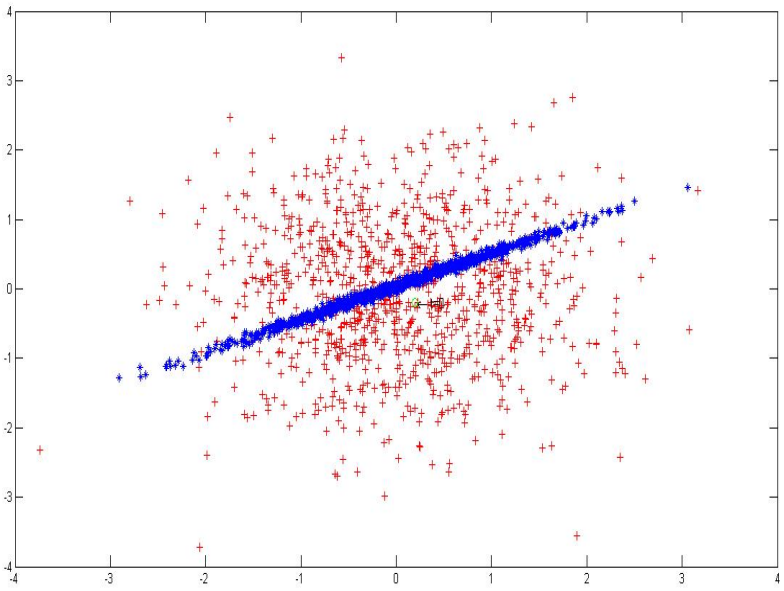


Figure 8: One thousand time steps of the Kalman filter example. The red illustrates the random vectors and the blue is after the prediction steps of the Kalman filter. Notice how the uncertainty is reduced by the prediction steps.

3.2 Colour Histograms

3.2.1 Definition

A histogram denoted $H = \{h^i\}_{i=1}^m$, is defined as a mapping from a set of d-dimensional integer vectors m to the set of non-negative reals. In this case m represents the number of bins, i represents the i^{th} bin. For example, when constructing a colour histogram, each pixel in an image has three values hence a 3-dimensional integer vector. The number of bins m is predetermined. The non-negative reals can either be the number or proportion of pixels in the bin range. Colour histogram refers to the probability mass function (pdf) of the colour intensitites [6]. Hence colour histograms are a type of histogram and are defined as:

$$h_c^i = P(C = c, I = i)$$

which is the joint probability of the colour intensity and the bin value, where C represents either R, G or$B$ of the RGB colour space. Colour histograms are formed by separating (discretizing) the colours of an image and counting the number of pixels with that colour. Several types of histograms can be used, for example, the separate RGB histogram in [1] which computes three separate histograms each representing one of the three colour levels of the RGB space. This is the method represented by h_c^i.
Colour histograms have several advantages and are thus widely used in image retrieval or image recognition [3] .

- They provide a condensed version of the image which is much easier for a computer to store.
- Image retrieval based on colour histograms should accurately retrieve images regardless of orientation, size and position of the image.
- They are quite efficient in terms of content information.

Colour histograms are not without limitations. They are unable to incorporate the spatial information of the colours in an image. Comparison of black and white images is not necessarily useful based on a colour algorithm [3].
Colour histograms contain statistical information that can be useful in robot localization.

3.2.2 Image comparison

Figure 9, illustrates an image comparison algorithm with the use of colour histograms. This algorithm is to be used in the application section of this report.

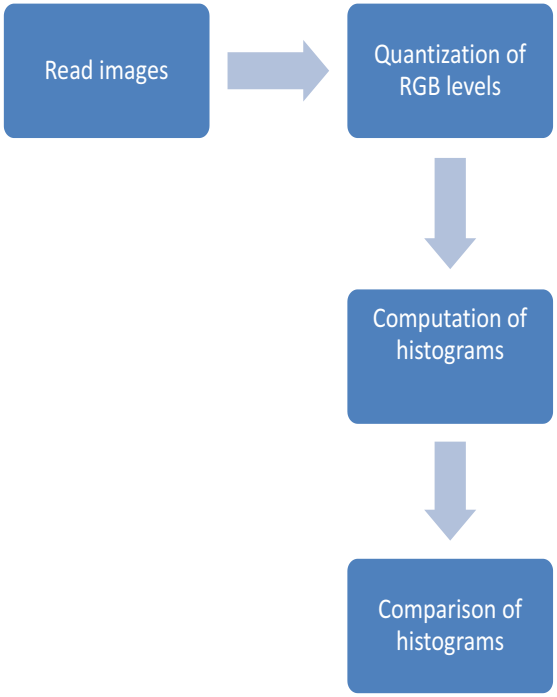


Figure 9: Image comparison algorithm

Firstly, images are read into MATLAB®. An image is represented as an $m \times n \times j$ array whereby $m \times n$ is the number of pixels in an image and j is the R, G and B level of an image. Each pixel in an image has three values corresponding to the R, G and B levels. These three values range from 0 to 255 which represent the colour intensities with 255 being the highest image. For example, a pixel in an image can have the values $[26, 90, 220]$ which represent the red, blue and green levels respectively.
The algorithm proceeds to quantize the images. This process groups the values of the respective colour levels into a predetermined number of groups. For example, suppose an image is quantized such that there are eight groups. Using the values $[26, 90, 220]$ from the previous example, the red value will be in group one, the green value in group two and the blue value in group six. The more groups there are, the better the algorithm is at comparing images at the expense of computational time.
Once the images have been quantized, the groups obtained from this process are used to construct a histogram. The x-axis of the histogram represents the groups (otherwise known as bins) from the quantization process and the y-axis represents the proportion of pixels in each bin so as to compare histograms of different images since each image has a different number of pixels.
Finally, the histograms can be compared for image matching. This is done by using the distance formulas discussed in the next section.

3.2.3 Histogram distances

This section gives a more in depth look as to how two histograms are compared in the algorithm in Figure 9. Consider $H_{C,1}$ and $H_{C,2}$ which represent two histograms with the same number of bins and C represents the three colours red, green or blue and for comparison purposes the non-negative reals is the relative frequency. Let N represent the number of bins, and h^i_{cj} represents the proportion of pixels of the i^{th} bin of colour c and the j^{th} histogram and $\overline{h}_{cj} = \frac{1}{m}\sum_{i=1}^{m} h^i_{cj}$ is the mean proportion of pixels of the j^{th} histogram, where $j = 1, 2$ for the two histograms being compared. The distances illustrated below will compare the similarity of images based on their histograms [10, 4, 14].

Euclidean distance:

$$D(H_{C,1}, H_{C,2}) = \sqrt{\sum_{i=1}^{m}(h^i_{c1} - h^i_{c2})^2}.$$

This is the most used out of the distances defined. The smaller the value, the more similarity there is in that colour of the image. Hence a value of zero means that the two images have the same amount of the colour. If all three values of the colour levels is zero, then the two images are the same.

Manhattan distance:

$$D(H_{C,1}, H_{C,2}) = \sum_{i=1}^{m} |h^i_{c1} - h^i_{c2}|.$$

The Manhattan distance sums the absolute value of the differences between the proportion of pixels between bins of different histograms. The smaller the value, the more similarity there is in the colour of the image. Hence a value of zero means that the two images has the same amount of that colour. If all three values of the colour levels is zero, then the two images are the same.

Histogram intersection:

$$D(H_{C,1}, H_{C,2}) = \sum_{i=1}^{m} \min(h^i_{c1}, h^i_{c2}).$$

The histogram intersection method calculates a goodness of match value. It computes the sum of the minimum value between the same bin from different histograms. The closer the value of a colour level is to one, the more similarity there is in the colour of the image. Hence a value of one means that the two images have the same amount of the colour level. If all three values of the colour levels is one, then the two images are the same.

Chi-Square distance:

$$D(H_{C,1}, H_{C,2}) = \sum_{i=1}^{m} \frac{(h^i_{C1} - h^i_{C2})^2}{h^i_{C2}}.$$

This distance function determines whether a histogram or distribution comes from the distribution of interest. It computes the sum of the squared differences of the observed histogram bin and the corresponding bin from histogram divided by the histogram of interest bin.

The smaller the value of each colour level, the more similarity there is in the colour distribution of that colour. Hence a value of zero means that the colour distribution of a colour level is the same for both images. If all three values of the colour levels is zero, then the two images are the same. Issues may arise when the value in the denominator is zero then the value cannot be calculated (singularity issue).

Correlation distance:

$$D(H_{C,1}, H_{C,2}) = \frac{\sum_{i=1}^{m}(h_{c1}^i - \overline{h}_{c1})(h_{c2}^i - \overline{h}_{c2})}{\sqrt{\sum_{i=1}^{m}(h_{c1}^i - \overline{h}_{c1})^2(h_{c2}^i - \overline{h}_{c2})^2}}.$$

This distance formula calculates the dependence between two images. The closer the value is to one of a colour level, the more similarity there is in the colour of the images. Hence a value of one means that the two images has the same amount of that colour level.

Bhattacharyya distance:

$$D(H_{C,1}, H_{C,2}) = \sqrt{1 - \frac{1}{\sqrt{\overline{h}_{c1}\overline{h}_{c2}N^2}}\sum_{i=1}^{m}\sqrt{h_{c1}h_{c2}}}.$$

The closer the value is to one, the more similarity there is in the colour of the image. Hence a value of zero means that the two images has the same amount of the colour level. If all three values of the colour levels is zero, then the two images are the same.

An interesting thing to note is that the Bhattacharyya distance is an approximation of the chi-square distance [4]. This helps avoid dividing by zero in the chi-square formula in the case that both distribution are zero.

These six distances [10, 4, 14] are used to compare images based on their colour content in the next section. This provides the basis in order for a robot to be able to learn its surrounding by comparing images it has stored.

4 Application

4.1 Colour recognition

The aim of the application is to write an algorithm that classifies red, green and blue cars as either red, green or blue with the use of the histogram distances mentioned in Section 3.2.3 in the background theory. Three main databases were created namely the car database, clean database and the test database. The car database has three groups of images where each group has 30 cars of the same colour (red, green or blue). Sample images of this database are found in Figure 10. The clean database has three groups with 30 images whereby each group has different images that are predominately one of the three colours (red, green and blue). Sample images of this database are found in Figure 11. The test database consists of three red, three green and three blue cars. One red test image is found in Figure 10 and one of the green test images is in Figure 11. All images in the three databases are of similar size (roughly 300*150). Each of nine test images was compared to each of the six groups of images in both of the two databases using the method discussed in Section 3.2.2 and 3.2.3 of this report. A mean of the 30 comparisons is obtained and this results in a set of 18 distances from the six distances formulas in Section 3.2.3 (three distances for each colour level).

To classify a test image, the colour level corresponding to the colour of the car in the test image is compared across the three groups in the database since one would expect, for example an image of a blue car to have a higher amount of blue in an image than any of the other colour. Hence for a car test image in Figure 10, the value of each of the histogram distances is observed for the three groups in each database.

A test image








Some red sample images

Some green sample images

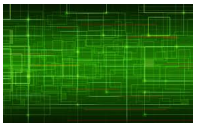



Some blue sample images

Figure 10: Test images as well as sample images from the car database

A test image








Some red sample images

Some green sample images




Some blue sample images

Figure 11: Test images as well as sample images from the clean database

4.1.1 Results

The results for the comparison algorithm are found in Tables 1-3 for the car database and Tables 4-6 for the clean database. Each table has the results of three test images of the same colour labelled car 1, car 2 and car 3 to each of the three groups in one database. Since there are nine test images (in groups of three) and two databases (three groups in each database), there are six tables. Three sets of results in a row arises from comparing one test image to the three groups, labelled red images, green images and blue images in any one of the two databases. Each test image is categorized using each of the distance formulas. For example, consider the test image in Figure 10 which is compared each group of images in the car database with sample images in Figure 10. The results of this comparison is found in first row of Table 1 under the heading car 1. The value of the euclidean distances for the red colour level after comparison to the three groups of car images are $0.283477, 0.342454$ and 0.347446. Since 0.283477 is the smallest value, the test image is classified as red which is a correct classification since the car is in fact red and this value is highlighted in red. Another example is test image in Figure 11 compared to the clean database with sample images in Figure 11 with results are displayed in Table 5 under the heading car 3. The values for the correlation distances for the green colour level after comparison are $-0.26322, 0.067394$ and -0.2171. Since 0.067394 (highlighted in red) is the closest to 1, the car is classified as red when in fact it is green hence it is misclassified. The classification is done in the same manner for all the other comparison using the distances discussed in the background theory.

Another thing to note is that the blue cars have a higher classification rate than the red or green cars. This may be a results of the shadows in the images or any dark colours in an image that may influence the blue values of the RGB colour space and hence bring a false classification as a result. This may be useful to look into in the future and find ways in which to possibly eliminate the effect of the shadows or colour depth in and image and only extract the useful information in an image. Another option would be to look at other colour spaces as well. Also a way to use all three histograms to classify could be investigated in future, as herein we focus on the histogram average of the colour associated with the test image.

Red									
	Red Images			Green Images			Blue images		
Car 1									
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**	Red	Green	Blue
Euclidean	0.283477	0.356948	0.353919	0.342454	0.391332	0.358432	0.347446	0.393235	0.364647
Manhattan	0.590751	0.77836	0.7594	0.738959	0.896986	0.812253	0.749211	0.883419	0.795221
Intersection	0.704625	0.61082	0.6203	0.630521	0.551507	0.593874	0.625395	0.558291	0.60239
ChiSquare	0.632114	1.255561	1.184787	0.873392	1.755563	1.333053	0.927665	1.537053	1.292813
Correlation	0.236547	0.28758	0.278704	-0.2371	-0.02014	0.190964	0.097516	0.107349	0.14741
Bhattacharyya	0.253794	0.345892	0.350209	0.342948	0.383231	0.371418	0.338093	0.37859	0.334091
Car 2									
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Euclidean	0.436991	0.576741	0.556032	0.294523	0.649541	0.520681	0.285884	0.627553	0.669915
Manhattan	0.857268	1.048878	1.02981	0.619843	1.255984	0.98003	0.583032	1.140567	1.29419
Intersection	0.571366	0.475561	0.485095	0.690079	0.372008	0.509985	0.708484	0.429716	0.352905
ChiSquare	2.482313	13.72353	12.1883	0.794567	10.36487	6.419179	1.526692	13.00023	16.24254
Correlation	-0.12002	0.40852	0.3886	0.324759	0.093073	0.491602	0.558739	0.234847	0.072027
Bhattacharyya	0.370745	0.459418	0.452719	0.263275	0.531524	0.421448	0.259353	0.492856	0.556666
Car 3									
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Euclidean	0.348913	0.358857	0.361103	0.256304	0.320337	0.290693	0.336813	0.334444	0.445057
Manhattan	0.73824	0.741946	0.723584	0.549406	0.712056	0.629318	0.720783	0.691926	0.931545
Intersection	0.63088	0.629027	0.638208	0.725297	0.643972	0.685341	0.639609	0.654037	0.534227
ChiSquare	2.773482	16.22817	16.15115	0.823696	6.828182	5.247432	2.126992	15.43554	23.5044
Correlation	-0.24452	0.215292	0.370877	0.280372	0.078674	0.512867	0.128772	0.254108	0.025647
Bhattacharyya	0.325167	0.346557	0.337695	0.241372	0.329715	0.281051	0.312233	0.325777	0.423047

Table 1: The table shows the mean distances after comparison of the red test images labelled car 1, 2 and 3 to the three groups of images in the cars database labelled red, green and blue images. Results for each colour level are displayed for each of the six distances. A highlighted values shows in which colour the test image is classfied to with respect to the distance used when looking only at the red histograms.

Green									
	Green Images			Red Images			Blue Images		
Car 1									
	Red	**Green**	Blue	Red	**Green**	Blue	Red	**Green**	Blue
Euclidean	0.302427	0.259879	0.249498	0.424178	0.345451	0.320687	0.381338	0.309858	0.357882
Manhattan	0.669409	0.604343	0.552912	0.962185	0.734738	0.668181	0.789179	0.65502	0.771274
Intersection	0.665295	0.697829	0.723544	0.518907	0.632631	0.66591	0.605411	0.67249	0.614363
ChiSquare	2.836859	1.091813	0.96365	13.51144	2.15736	2.436522	8.845633	1.912586	3.324649
Correlation	0.31515	0.12595	0.453047	-0.20408	0.066166	0.271865	0.173284	0.154909	-0.02218
Bhattacharyya	0.293178	0.259471	0.249499	0.426182	0.31773	0.304535	0.361743	0.289332	0.339747
Car 2									
	Red	**Green**	Blue	Red	**Green**	Blue	Red	**Green**	Blue
Euclidean	0.294035	0.302751	0.419821	0.452369	0.367792	0.470698	0.304205	0.320707	0.577209
Manhattan	0.648136	0.675661	0.812312	1.012679	0.79095	0.894478	0.643828	0.681623	1.150346
Intersection	0.675932	0.66217	0.593844	0.49366	0.604525	0.552761	0.678086	0.659188	0.424827
ChiSquare	2.311119	2.983545	8.762842	10.91476	6.991104	22.25712	6.90409	6.50753	30.76439
Correlation	0.347087	0.048745	0.504894	-0.24769	0.078766	0.385292	0.485282	0.211192	0.063083
Bhattacharyya	0.278924	0.303735	0.367323	0.437082	0.343507	0.417499	0.295399	0.307181	0.519495
Car 3									
	Red	**Green**	Blue	Red	**Green**	Blue	Red	**Green**	Blue
Euclidean	0.487412	0.41025	0.464274	0.539891	0.392422	0.462432	0.419983	0.401069	0.516214
Manhattan	1.077952	0.839976	1.018195	1.110406	0.76465	0.968154	0.875043	0.810635	1.083756
Intersection	0.461024	0.580012	0.490903	0.444797	0.617675	0.515923	0.562478	0.594683	0.458122
ChiSquare	3.366059	1.212593	2.954437	4.253202	1.118624	3.389691	2.446277	1.246299	3.796224
Correlation	0.118559	-0.08827	0.32211	0.051661	0.080658	0.333056	0.508826	0.005353	0.132154
Bhattacharyya	0.465197	0.370938	0.442332	0.474939	0.358179	0.42386	0.387122	0.368365	0.451342

Table 2: The table shows the mean distances after comparison of the green test images labelled car 1, 2 and 3 to the three groups of images in the cars database labelled red, green and blue images. Results for each colour level are displayed for each of the six distances. A highlighted values shows in which colour the test image is classfied to with respect to the distance used when looking only at the green histograms.

Blue									
	Blue Images			Red Images			Green Images		
Car 1									
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Euclidean	0.376437	0.345015	0.435477	0.341381	0.377478	0.529555	0.366245	0.339177	0.570628
Manhattan	0.786467	0.699207	0.81499	0.754417	0.777971	1.02465	0.747137	0.708823	1.122367
Intersection	0.606767	0.650396	0.592505	0.622792	0.611015	0.487675	0.626432	0.645589	0.438817
ChiSquare	1.194942	1.012877	1.608237	1.26642	1.30832	2.737944	1.134476	1.004257	2.902829
Correlation	0.055359	0.027386	0.099775	0.05998	-0.04672	-0.06179	-0.16796	-0.11608	-0.2606
Bhattacharyya	0.35055	0.322385	0.350755	0.320874	0.354531	0.463267	0.353093	0.325398	0.495426
Car 2									
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Euclidean	0.342508	0.329367	0.31046	0.387981	0.359665	0.337979	0.260958	0.260836	0.270842
Manhattan	0.716285	0.698228	0.653523	0.874889	0.756918	0.739314	0.577811	0.591663	0.630694
Intersection	0.641858	0.650886	0.673239	0.562555	0.621541	0.630343	0.711095	0.704169	0.684653
ChiSquare	6.108251	2.810421	1.462182	9.225207	3.065671	1.343267	1.887023	1.356843	0.708391
Correlation	0.22046	0.082441	-0.05677	-0.22136	0.017419	-0.02927	0.360216	0.145849	0.14292
Bhattacharyya	0.329114	0.309067	0.278683	0.388724	0.330778	0.314622	0.254607	0.256636	0.265242
Car 3									
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Euclidean	0.430903	0.418808	0.435947	0.352331	0.424986	0.498646	0.332633	0.327642	0.492401
Manhattan	0.931096	0.912738	0.902395	0.711936	0.922346	1.094104	0.731754	0.717282	1.093838
Intersection	0.534452	0.543631	0.548802	0.644032	0.538827	0.452948	0.634123	0.641359	0.453081
ChiSquare	34.40743	13.70386	2.32678	50.55937	14.1237	3.211515	9.130684	5.526374	3.405156
Correlation	-0.16844	-0.20169	-0.00978	0.061326	-0.12893	-0.14703	0.072155	0.017671	-0.23372
Bhattacharyya	0.4197	0.410669	0.380579	0.359862	0.408622	0.467125	0.327877	0.32753	0.471503

Table 3: The table shows the mean distances after comparison of the blue test images labelled car 1, 2 and 3 to the three groups of images in the cars database labelled red, green and blue images. Results for each colour level are displayed for each of the six distances. A highlighted values shows in which colour the test image is classfied to with respect to the distance used when looking only at the blue histograms.

Red									
	Red Images			Green Images			Blue Images		
Car 1									
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Euclidean	0.467019	0.638711	0.672634	0.625243	0.549303	0.648148	0.819339	0.522263	0.494737
Manhattan	0.983458	1.210823	1.258428	1.294312	1.226472	1.193156	1.535155	1.138616	1.074626
Intersection	0.508271	0.394589	0.370786	0.352844	0.386764	0.403422	0.232423	0.430692	0.462687
ChiSquare	1.822439	2.597958	3.361646	3.446627	4.524538	1.950194	4.55439	3.254578	2.581738
Correlation	0.231105	0.521803	0.417299	-0.21093	-0.28726	0.576279	0.106009	0.013244	-0.00112
Barchat	0.466886	0.578928	0.598942	0.613942	0.570052	0.582529	0.698868	0.530688	0.507747
Car 2									
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Euclidean	0.646647	0.340315	0.428776	0.428142	0.771012	0.302965	0.549265	0.573774	0.805733
Manhattan	1.302337	0.588143	0.714385	0.858624	1.474704	0.517585	1.03688	1.022455	1.564608
Intersection	0.348832	0.705929	0.642807	0.570688	0.262648	0.741208	0.48156	0.488772	0.217696
ChiSquare	6.915857	1.454502	1.853419	1.185673	13.28044	0.52785	1.526098	3.996262	31.59082
Correlation	-0.32139	0.819998	0.729632	0.626258	-0.12524	0.906059	0.884966	0.378692	-0.18574
Barchat	0.609021	0.32799	0.371022	0.420869	0.664894	0.279921	0.502532	0.452731	0.719804
Car 3									
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Euclidean	0.547039	0.695976	0.549745	0.508879	0.424263	0.524786	0.818548	0.369034	0.596809
Manhattan	1.163711	1.294028	1.006583	0.992632	0.943205	0.947935	1.516187	0.742726	1.253755
Intersection	0.418144	0.352986	0.496708	0.503684	0.528397	0.526032	0.241907	0.628637	0.373123
ChiSquare	7.656759	3.068399	1.758132	2.437924	5.291455	1.06426	5.274401	1.800974	48.11782
Correlation	-0.29202	0.400395	0.708894	0.424116	0.096906	0.817686	0.088612	0.524485	-0.22923
Barchat	0.550056	0.585222	0.479711	0.477728	0.450334	0.443118	0.677746	0.358702	0.600446

Table 4: The table shows the mean distances after comparison of the red test images labelled car 1, 2 and 3 to the three groups of images in the clean database labelled red, green and blue images. Results for each colour level are displayed for each of the six distances. A highlighted values shows in which colour the test image is classfied to with respect to the distance used when looking only at the red histograms.

Green									
	Green Images			Red Images			Blue Images		
				Car 1					
	Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
Euclidean	0.527456	0.363245	0.652796	0.598579	0.738609	0.653829	0.765714	0.396378	0.494581
Manhattan	1.025381	0.822591	1.218929	1.309931	1.407091	1.234451	1.381915	0.835649	1.065996
Intersection	0.48731	0.588705	0.390536	0.345034	0.296455	0.382775	0.309042	0.582176	0.467002
ChiSquare	1.951295	1.578827	2.024262	34.9927	3.655167	2.398718	2.964129	1.375177	6.830974
Correlation	0.384217	0.239372	0.655826	-0.21686	0.233159	0.557534	0.353642	0.3809	-0.1927
Bhattacharyya	0.468379	0.398145	0.558431	0.615591	0.635269	0.57429	0.62249	0.385667	0.513728
				Car 2					
	Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
Euclidean	0.41232	0.379824	0.370854	0.657637	0.746856	0.461755	0.646339	0.365418	0.720134
Manhattan	0.780493	0.854604	0.653603	1.416107	1.405985	0.804855	1.169956	0.75043	1.451243
Intersection	0.609753	0.572698	0.673199	0.291946	0.297008	0.597572	0.415022	0.624785	0.274379
ChiSquare	0.990641	2.648061	0.562228	28.511	4.457473	1.778776	1.767652	1.471665	61.65813
Correlation	0.700097	0.220837	0.898995	-0.41005	0.199012	0.734154	0.685635	0.50641	-0.2014
Bhattacharyya	0.376395	0.408715	0.334359	0.657679	0.629945	0.40532	0.534565	0.357917	0.686035
				Car 3					
	Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
Euclidean	0.556195	0.554356	0.510197	0.707345	0.805354	0.58418	0.532803	0.59475	0.641623
Manhattan	1.111506	1.191771	0.881952	1.404789	1.487184	1.040031	0.912394	1.210429	1.333271
Intersection	0.444247	0.404114	0.559024	0.297605	0.256408	0.479985	0.543803	0.394785	0.333364
ChiSquare	3.518642	3.281066	1.509499	8.201283	4.220937	4.289716	2.375959	3.296989	6.523278
Correlation	0.373542	-0.26322	0.749338	-0.09848	0.067394	0.559294	0.767641	-0.2171	-0.05908
Bhattacharyya	0.54596	0.546048	0.483562	0.649581	0.675625	0.52515	0.499572	0.56008	0.609998

Table 5: The table shows the mean distances after comparison of the green test images labelled car 1, 2 and 3 to the three groups of images in the clean database labelled red, green and blue images. Results for each colour level are displayed for each of the six distances. A highlighted values shows in which colour the test image is classfied to with respect to the distance used when looking only at the green histograms.

Blue									
	Blue Images			Red Images			Green Images		
Car 1									
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Euclidean	0.863593	0.512948	0.509756	0.502994	0.827012	0.967571	0.624565	0.455337	0.990835
Manhattan	1.569483	1.019486	0.981223	1.114219	1.564232	1.753994	1.212187	0.978523	1.770648
Intersection	0.215258	0.490257	0.509389	0.44289	0.217884	0.123003	0.393907	0.510738	0.114676
ChiSquare	6.314409	2.377569	2.178261	2.807302	6.954755	16.55229	3.383133	1.941698	17.10865
Correlation	-0.04455	-0.12714	0.192786	0.125203	-0.14493	-0.24764	-0.05344	-0.10173	-0.2486
Bhattacharyya	0.715935	0.493326	0.458296	0.508824	0.706969	0.801023	0.58467	0.470707	0.820508
Car 2									
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Euclidean	0.755399	0.426421	0.427983	0.57316	0.757094	0.753977	0.503688	0.367691	0.770095
Manhattan	1.382924	0.903363	0.920681	1.254534	1.454709	1.470593	0.990357	0.826861	1.475313
Intersection	0.308538	0.548319	0.53966	0.372733	0.272646	0.264704	0.504822	0.586569	0.262344
ChiSquare	3.055424	1.715417	3.178416	24.2633	3.977833	4.438983	1.757173	1.772993	4.310073
Correlation	0.403728	0.247797	-0.08097	-0.259	0.142149	0.078487	0.451069	0.245945	0.138577
Bhattacharyya	0.62325	0.411864	0.441745	0.592627	0.653239	0.661282	0.457308	0.398311	0.662933
Car 3									
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Euclidean	0.891834	0.553563	0.511998	0.518482	0.859381	0.924021	0.64686	0.441893	0.944312
Manhattan	1.64566	1.199165	1.035963	1.049305	1.658704	1.762198	1.315098	0.988413	1.784622
Intersection	0.17717	0.400418	0.482018	0.475348	0.170648	0.118901	0.342451	0.505793	0.107689
ChiSquare	8.187011	4.480621	3.360721	134.5703	8.87667	17.12948	4.373949	5.013136	14.71736
Correlation	-0.1697	-0.24567	0.095793	0.060191	-0.22997	-0.29004	-0.15795	0.009599	-0.29524
Bhattacharyya	0.743776	0.543188	0.460104	0.540483	0.747634	0.804204	0.59605	0.459375	0.813498

Table 6: The table shows the mean distances after comparison of the blue test images labelled car 1, 2 and 3 to the three groups of images in the clean database labelled red, green and blue images. Results for each colour level are displayed for each of the six distances. A highlighted values shows in which colour the test image is classfied to with respect to the distance used when looking only at the blue histograms.

5 Conclusion

The aim of this research was to analyse a method that can be used for visual information in the robotics industry. The field of robotics continues to grow and new challenges arise in order to make a robot self learning. This requires a lot of input so that a robot can make use of this information in order to predict where it may be at any given time. This led to an analysis of images based on their colour with the use of RGB colour histograms. Histograms of different images were compared using histogram distances and this can be used to compute similarities of a current image and training images as discussed in the self-organizing maps, in which a robot can make a map based on this information. Colour histograms can be used for a wide range of colour spaces such as the HSV space which may have an advantage over the RGB space [6]. Information of the distances calculated can be used as sensor input for the Kalman filter in order to better predict a robot's location.

References

[1] M Ali, J Sitte, and U Witkowski. Parallel early vision algorithms for mobile robots. In *Proceedings of the 4th International Symposium on Autonomous Mini-Robots for Research and Edutainment (AMiRE)*, pages 133–140. Research Gate, 2007.

[2] M Bader, H Brunner, T Hambock, A Hofmann, and M Vincze. Colour histograms as background description: An approach to overcoming the uniform-goal problem within the SPL for the Robocup WC 2012. Technical report, Institute of Computer Science, University of Applied Sciences and Automation and Control Institute (ACIN), Vienna University of Technology, 2012.

[3] Rishav Chakravarti and Xiannong Meng. A study of color histogram based image retrieval. In *ITNG*, pages 1323–1328, 2009.

[4] Konstantinos G Derpanis. The Bhattacharyya measure. *Mendeley Computer*, 1(4):1990–1992, 2008.

[5] C Hussain, D Venkata Rao, and T Praveen. Color histogram based image retrieval. *International Journal of Advanced Engineering Technology*, 4(3):63–66, July-Sept 2013.

[6] S Jeong. Histogram-based color image retrieval. Technical report, University of Stanford, March 2001.

[7] R Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[8] R King. Statistics for robotics: Colour classification. Workshop notes, December 2015.

[9] R King. Statistics for robotics: Kalman filter proposal. Workshop notes, December 2015.

[10] I Ranó, E Lazkano, and B Sierra. On the application of colour histograms for mobile robot localisation. In *ECMR (European Conference on Mobile Robotics)*, volume 1, pages 189–193, 2005.

[11] R Rojas. The Kalman filter. *Institute for Informatik*, 2003.

[12] J Smith and S Chang. Tools and techniques for color image retrieval. In *Electronic Imaging: Science & Technology*, pages 426–437. International Society for Optics and Photonics, 1996.

[13] P Swerling. A proposed stagewise differential correction procedure for satellite tracking and prediciton. Technical Report P-1292, RAND Corporation, 1958.

[14] Gábor J Székely, Maria L Rizzo, Nail K Bakirov, et al. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

[15] S Thrun, W Burgard, and D Fox. *Probabilistic Robotics*. MIT Press, 2006.

[16] F Tussel. Kalman filtering in R. *Journal of Statistical Software*, 39(2):1–27, March 2011.

[17] I Urich and I Nourbakhsh. Appearance-based place recognition for topological localization. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1023–1029. International Conference on Robotics and Automation, 2000.

[18] Max Welling. The Kalman filter. Technical Report / Lecture Notes, California Institute of Technology, 2010.

Appendix

Kalman filter illustrative example

```
x=[0:0.1:30];

norm=normpdf(x,10,5)


norm2=normpdf(x,8,3)

norm3=normpdf(x,9,1.5)

% figure;
% plot(x,norm,'color','r','Linewidth',1)
% hold on;
% plot(x,norm2,'color','b','Linewidth',2)
% hold on;
% plot(x,norm3,'Linestyle','--')
% hold off

rnorm=normpdf(x,20,4)
rnorm2=normpdf(x,25,3.5);
rnorm3=normpdf(x,22.5,2)

figure;
plot(x,rnorm,'color','r','Linewidth',1)
hold on;
plot(x,rnorm2,'color','b','Linewidth',1)
hold on;
plot(x,rnorm3,'Linestyle','--')
hold off;
```

Kalman filter algorithm

```
n=1000
sigma=[0.4 0.3;0.3 0.45]
mut=[0.2;-0.2]
At=[0.4 0.5;0.2 0.3]
Bt=[0.1 0.5;0.2 0.1]
gamma=[0.4;0.2]
Rt=[0.5 0.2; 0.2 0.1]
Gt=[0.1 0.4;0.3 0.6]
I=[1 0;0 1]
Qt=[0.4 0.1;0.1 0.2]
muz=[0;0]
sigmaz=[1 0;0 1]
zt=mvnrnd(muz,sigmaz,n)
mu=zeros(n,2)

for t=1:n
mutbar=At*mut + Bt*gamma
sigmabar=At*sigma*At' + Rt
Y=inv(Gt*sigmabar*Gt'+ Qt)
```

```
Kt=sigmabar*Gt'*Y

mutnew=mutbar + Kt*(zt(t,:)' - Gt*mutbar)
sigmanew=(I-Kt*Gt)*sigmabar
%update step:
mut=mutnew
sigma=sigmanew
mu(t,:)=mutnew
end

plot(zt(:,1),zt(:,2),'+','Color',[1 0 0]);
hold on
plot(mu(:,1),mu(:,2),'*','Color',[0 0 1]);
hold on
plot(0.2,-0.2,'0','Color',[0 1 0]);
hold off
x0=0.2;
y0=-0.2;



txt= ' \leftarrow t=0';

text(x0,y0,txt)
```

Colour Histogram algorithm (including histogram comparisons)

```
function  [euclidean,manhattan,inter,chisq,corr,bharchat, distance]
=comparison(image1)

imgPath = 'C:\Users\User\Dropbox\My shared Folder\Tinashe and Carel\
Histogram\Green Cars\'; dCell = dir([imgPath '*.jpg']);
disp('Loading image files.');
for d = 1:length(dCell)
img{d} = imread([imgPath dCell(d).name]);
pause(2)
end


g=imread(image1)

dist=zeros(length(dCell),18)

for d = 1:length(dCell)

img{d} = imread([imgPath dCell(d).name]);

h=img{d};


[n,m,j]=size(g)
[a,b,c]=size(h)
```

```
newg= zeros(n,m,j);
newh=zeros(a,b,c);

for k = 1 : j
    for i = 1 : n
        for l = 1: m
            value = g(i,l,k);

            if value <=32
                newg(i,l,k) = 1;
            elseif value <= 64
                 newg(i,l,k) = 2;
            elseif value <= 97
                 newg(i,l,k) = 3;
            elseif value <= 129
                 newg(i,l,k) = 4;
            elseif value <= 161
                 newg(i,l,k) = 5;
            elseif value <= 193
                 newg(i,l,k) = 6;
            elseif value <= 225
                 newg(i,l,k) = 7;
            else  newg(i,l,k) = 8;
            end
        end
    end
end


for k = 1 : c
    for i = 1 : a
        for l = 1: b
            value = h(i,l,k);

            if value <=32
                newh(i,l,k) = 1;



            elseif value <= 64
                 newh(i,l,k) = 2;
            elseif value <= 97
                 newh(i,l,k) = 3;
            elseif value <= 129
                 newh(i,l,k) = 4;
            elseif value <= 161
                 newh(i,l,k) = 5;
            elseif value <= 193
                 newh(i,l,k) = 6;
            elseif value <= 225
                 newh(i,l,k) = 7;
            else  newh(i,l,k) = 8;
```

```
            end
        end
    end
end

count=zeros(8,3);%red
count2=zeros(8,3)
proportion=zeros(8,3)
proportion2=zeros(8,3)

total=n*m
total2=a*b

for f=1:j
for k=1:8
    for i=1:n
        for l=1:m
            value2=newg(i,l,f);

if value2 == k
    count(k,f)=count(k,f)+1;
        end
        end
    end
    proportion(k,f)=count(k,f)/total;
end
end

for f=1:c
for k=1:8
    for i=1:a
        for l=1:b
            value3=newh(i,l,f);

if value3 == k
    count2(k,f)=count2(k,f)+1;
        end
        end
    end
    proportion2(k,f)=count2(k,f)/total2;
end
end



    euclidean=sqrt(sum((proportion-proportion2).^2))
    manhattan=sum(abs(proportion-proportion2))


    sum1=zeros(1,3);

    for j=1:3
for i=1:8
```

```
      sum1(1,j)=sum1(1,j)+min(proportion(i,j),proportion2(i,j));
  end
    end
s1=sum(proportion);
s2=sum(proportion2);

inter=sum1
chisq=sum(((proportion-proportion2).^2)./proportion)

hb1=(1/8)*s1;
hb2=(1/8)*s2;
hbar1=vertcat(hb1,hb1,hb1,hb1,hb1,hb1,hb1,hb1);
hbar2=vertcat(hb2,hb2,hb2,hb2,hb2,hb2,hb2,hb2);

numcorr=sum((proportion-hbar1).*(proportion2-hbar2))

varcovh1=sum((proportion-hbar1).^2)
varcovh2=sum((proportion2-hbar2).^2)

dencorr=sqrt(varcovh1.*varcovh2)

corr=numcorr./dencorr

bt1=1./sqrt(hb1.*hb2.*(8^2))
bt2=sum(sqrt(proportion.*proportion2))

bharchat=sqrt(1-bt1.*bt2)
    dist(d,1:18)=(horzcat(euclidean,manhattan,inter,chisq,corr,bharchat))
end

dist2=mean(dist)

distance=vertcat(dist2(1,1:3),dist2(1,4:6),dist2(1,7:9),dist2(1,10:12),
dist2(1,13:15),dist2(1,16:18))
```

Ridge regression

Lebogang Komane 13257022

STK795 Research Report

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

Supervisor: Dr N Strydom

Department of Statistics, University of Pretoria

2 November 2016

Abstract

The presence of multicollinearity in multiple regression affects the estimation of regression coefficients. Particularly, the ordinary least squares (OLS) estimates become highly unstable and have a large prediction variance when multicollinearity is present. Consequently, having a large variance means that some variables are coming out statistically insignificant.

The main focus of this paper is on ridge regression. Hoerl and Kennard [5] introduced ridge regression as a remedial measure for multicollinearity. Ridge regression introduces a small bias, the ridge constant (k), to shrink the OLS estimates towards zero so that more stable and accurate estimates can be obtained. This paper presents the ridge estimator and its properties and also how the ridge estimator is obtained geometrically.

Furthermore we discuss different methods for selecting the ridge constant. Variations and applications on ridge regression will also be discussed. In order to illustrate ridge regression practically with a data set we primarily use SAS procedures [2]. The applications are intended to give more insight on how ridge regression works in practice.

Declaration

I, *Lebogang Komane,* 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.

_____
Lebogang Komane

_____
Dr Nina Strydom

_____
Date

Acknowledgments

I would like to thank my supervisor, Dr Nina Strydom for her guidance and valuable advice. I honestly appreciate and acknowledge her help and for providing most of the essential materials evaluated in this study. Her expertise, understanding, and patience, added considerably to this research report. I appreciate her assistance provided at all levels in writing this report. It was under her tutelage that I developed a focus and interest in this topic. She provided me with direction and became more of a mentor and friend, than a professor.

The author would also like to thank the Centre of Artificial Intelligence Research (CAIR) for financial support in the form of a postgraduate bursary.

Contents

1 Introduction **6**

2 Theoretical Background **6**
2.1 The role of the mean square error (MSE) and variance 7
2.2 Ridge regression 8
2.3 Properties of the ridge estimator 9
2.4 Geometric interpretation of ridge regression 9
2.5 The ridge constant 10
2.6 Variations and developments 11

3 Application **12**

4 Conclusion **15**

Appendix **17**

List of Figures

1 Geometric interpretation of ridge regression (from STAT 897D [1]) 10
2 Ridge trace 14

1 Introduction

A multiple regression model is one of the most used statistical methods in almost every field of science and technology, finance and economics. It is primarily used to examine the relationship between a dependent variable and multiple independent variables [3]. Many a times when a multiple regression model is fitted to the observed data set, the independent variables tend to be highly correlated. This often occurs in multiple regression analysis in the presence of multicollinearity.

Multicollinearity is often used to describe the existence of high correlation between independent variables. When multicollinearity is present in the data, very poor estimates are usually obtained. The ordinary least squares (OLS) estimator has a large prediction variance even though it has the property of being the best linear unbiased estimator (Gauss-Markov theorem [3]). Consequently, having an inflated variance means that some variables are coming out statistically insignificant when they might be significant without multicollinearity.

Hoerl and Kennard [5] suggested using ridge regression as a bias regression technique that analyzes a multiple regression model in the presence of multicollinearity. Ridge regression introduces a small bias, the ridge constant (k) to the diagonal elements of the OLS estimator. This defines the ridge estimator which shrinks the least squares estimates towards zero to get more reliable and stable estimates. The ridge constant is also used to reduce the inflated variance. Therefore, determining an optimal value of the ridge constant is crucial in ridge regression since it controls the amount of shrinkage.

Several methods have been suggested to estimate the best value for the ridge constant, such that the introduction in bias does not exceed the reduction in the prediction variance. Hoerl and Kennard [5] proposed using a graphical method, the ridge trace. Hoerl *et al.* [7] suggested the fixed point method. In a following paper Hoerl and Kennard [6] proposed the iterative method. Many researchers also suggested other methods for selecting the ridge constant such as Lawless and Wang [9], McDonald and Galarneau [11], Mallows [10], Khalaf and Shukur [8] and many more others.

The introduction of ridge regression as a bias technique to deal with the problem of multicollinearity has been followed by a number of papers in statistical literature. This includes the the least absolute shrinkage and selection operator (LASSO) introduced by Tibshirani [13] followed by the elastic net introduced by Zou and Hastie [14].

2 Theoretical Background

Suppose a multiple regression model is written in matrix notation as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u},$$

where $\mathbf{y}$ is a vector of observations on the dependent variable, $\mathbf{X}$ is a design matrix, $\boldsymbol{\beta}$ is a column vector of regression coefficients to be estimated and $\mathbf{u}$ is a vector of residuals [3]. Essentially, $\mathbf{y}$ is a vector of independent normal random variables,

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$$

where σ^2 is the variance and $\mathbf{I}$ is the identity matrix.

In the estimation of the regression coefficients, the method of ordinary least squares minimizes the criterion,

$$Q = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

yielding the following OLS estimator,

$$\hat{\boldsymbol{\beta}}_{ols} = (\mathbf{X'X})^{-1}\mathbf{X'y}$$

When perfect multicollinearity (exact relationships) is present, the determinant of the matrix $\mathbf{X'X}$ becomes singular and will not be invertible, thus the estimation of the regression coefficients and standard errors cannot be determined. For variables that are highly correlated, the determinant of the matrix $\mathbf{X'X}$ becomes nearly singular. Therefore, very poor estimates of the regression coefficients are usually obtained.

2.1 The role of the mean square error (MSE) and variance

The expected value of the OLS estimator is:

$$\begin{aligned} \mathrm{E}(\hat{\boldsymbol{\beta}}_{ols}) &= \mathrm{E}[(\mathbf{X'X})^{-1}\mathbf{X'y})] \\ &= (\mathbf{X'X})^{-1}\mathbf{X'}\mathrm{E}\{\mathbf{y}\} \\ &= (\mathbf{X'X})^{-1}\mathbf{X'X}\boldsymbol{\beta} \\ &= \boldsymbol{\beta} \end{aligned}$$

$\therefore \hat{\boldsymbol{\beta}}_{ols}$ is an unbiased estimator of $\boldsymbol{\beta}$.

The variance of the OLS estimator is defined as:

$$\begin{aligned} \mathrm{Var}\{\hat{\boldsymbol{\beta}}_{ols}\} &= \sigma^2[(\mathbf{X'X})^{-1}\mathbf{X'y}] \\ &= (\mathbf{X'X})^{-1}\mathbf{X'}\sigma^2\{\mathbf{y}\}\mathbf{X}((\mathbf{X'X})^{-1}\mathbf{X'})' \\ &= (\mathbf{X'X})^{-1}\mathbf{X'}(\sigma^2\mathbf{I})\mathbf{X}(\mathbf{X'X})^{-1})' \\ &= \sigma^2(\mathbf{X'X})^{-1}\mathbf{X'X}((\mathbf{X'X})')^{-1} \\ &= \sigma^2(\mathbf{X'X})^{-1} \end{aligned}$$

The equation above shows that the variance of the OLS estimator becomes inflated when the determinant of the matrix $\mathbf{X'X}$ becomes nearly singular. Therefore, the OLS estimates will have large standard errors resulting in statistical inferences becoming unreliable.

The MSE of the OLS estimator is:

$$\begin{aligned} \mathrm{MSE}(\hat{\boldsymbol{\beta}}_{ols}) &= E(\hat{\boldsymbol{\beta}}_{ols} - \boldsymbol{\beta})'(\hat{\boldsymbol{\beta}}_{ols} - \boldsymbol{\beta}) \\ &= \mathrm{TraceVar}(\hat{\boldsymbol{\beta}}_{ols}) \\ &= \sigma^2\mathrm{Trace}(\mathbf{X'X})^{-1} \\ &= \sigma^2 \sum_{i=1}^{p} \frac{1}{\lambda_i} \end{aligned}$$

where λ_i is the i^{th} eigenvalue of the matrix $\mathbf{X'X}$. When multicollinearity is present, the eigenvalues become relatively small, thus the MSE of the OLS estimator becomes very large. When the variance is inflated we get wider confidence intervals, resulting in the acceptance of the null hypothesis (i.e. the true population coefficient is zero).

2.2 Ridge regression

Hoerl and Kennard [5] suggested that the ill-conditioning problem of the OLS estimates could be improved by adding the ridge constant (k), to the matrix $\mathbf{X'X}$ before inverting it. Therefore, ridge regression modifies the matrix $\mathbf{X'X}$ such that its determinant is non-singular, this ensures that the inverse of the matrix $\mathbf{X'X}$ can be determined. The matrix becomes $(\mathbf{X'X} + k\mathbf{I})$ and the term $k\mathbf{I}$ effectively shrinks the OLS estimates towards zero.

The ridge estimator is defined as:

$$\hat{\boldsymbol{\beta}}_{ridge} = (\mathbf{X'X} + k\mathbf{I})^{-1}\mathbf{X'y}$$

The ridge estimator is basically the same as the OLS estimator when $k = 0$. Essentially, the ridge constant is the parameter that differentiates the ridge estimator from the OLS estimator [12].

The expected value of the ridge estimator is:

$$\begin{aligned} \mathrm{E}(\hat{\boldsymbol{\beta}}_{ridge}) &= \mathrm{E}[(\mathbf{X'X} + k\mathbf{I})^{-1}\mathbf{X'y})] \\ &= (\mathbf{X'X} + k\mathbf{I})^{-1}\mathbf{X'}\mathrm{E}\{\mathbf{y}\} \\ &= (\mathbf{X'X} + k\mathbf{I})^{-1}\mathbf{X'X}\boldsymbol{\beta} \\ &= \mathrm{L}_k\boldsymbol{\beta} \end{aligned}$$

where $\mathrm{L}_k = \mathbf{X'X} + k\mathbf{I})^{-1}\mathbf{X'X}$. Therefore, $\hat{\boldsymbol{\beta}}_{ridge}$ is a biased estimator of $\boldsymbol{\beta}$.

The variance of the ridge estimator is:

$$\begin{aligned} \mathrm{Var}(\hat{\boldsymbol{\beta}}_{ridge}) &= \sigma^2[(\mathbf{X'X} + k\mathbf{I})^{-1}\mathbf{X'y}] \\ &= ((\mathbf{X'X} + k\mathbf{I})^{-1}\mathbf{X})\boldsymbol{\sigma^2}\{\mathbf{y}\}((\mathbf{X'X} + k\mathbf{I})^{-1}\mathbf{X})' \\ &= \sigma^2(\mathbf{X'X} + k\mathbf{I})^{-1}\mathbf{X'X}(\mathbf{X'X} + k\mathbf{I})^{-1} \\ &= \sigma^2(\mathbf{X'X}+k\mathbf{I})^{-1}\mathbf{X'X}(\mathbf{X'X}+k\mathbf{I})^{-1} \end{aligned}$$

The variance of the ridge estimator shows that the ridge constant reduces the inflated variance in the presence of multicollinearity. Thus as the ridge constant increases the variance decreases.

The MSE of the ridge estimator is:

$$\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{ridge}}) = \text{Var}(\hat{\boldsymbol{\beta}}_{ridge}) + \text{Bias}^2$$

$$= \sigma^2 \text{Trace}[(\mathbf{X'X}+k\mathbf{I})^{-1}\mathbf{X'X}(\mathbf{X'X}+k\mathbf{I})^{-1}] + k^2\boldsymbol{\beta'}(\mathbf{X'X}+k\mathbf{I})^{-2}\boldsymbol{\beta}$$

$$= \sigma^2 \sum_{i=1}^{p} \frac{\lambda_i}{(\lambda_i + k)^2} + k^2\boldsymbol{\beta'}(\mathbf{X'X}+k\mathbf{I})^{-2}\boldsymbol{\beta}$$

This equation shows that the trade-off between the introduction in bias and the reduction in variance can be best explained by the MSE. Thus, the ridge constant should be chosen such that the increase in bias does not exceed the reduction in the prediction variance. This will result in a better MSE of the ridge estimator than the MSE of the unbiased OLS estimator.

2.3 Properties of the ridge estimator

- The ridge estimator is obtained by minimizing the residual sum of squares (RSS):

  $$\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2$$

  subject to the constraint $\sum_{j=1}^{p} {\beta_j}^2 \leq c$, where c is a positive constant [4].

- The ridge estimator defined as:

  $$\hat{\boldsymbol{\beta}}_{ridge} = (\mathbf{X'X} + k\mathbf{I})^{-1}\mathbf{X'y},$$

  is a linear transformation of the unbiased OLS estimator.

- The ridge estimator is a biased estimator but has a smaller MSE than the OLS estimator.

- The ridge estimator always produces shrinkage towards zero and the amount of shrinkage is controled by the ridge constant .

- Bias in the ridge estimator increases and decreases with the ridge constant.

2.4 Geometric interpretation of ridge regression

Figure 1 shows a geometric representation of how the ridge estimator is obtained with only two independent variables in the model. The ellipses around the OLS estimate correspond to the contours of the RSS. The OLS estimate at the center of the ellipse is the least squares solution where the RSS, achieves its minimum. The blue shaded area of the circle corresponds to the constraint in ridge regression, $\sum_{j=1}^{p} {\beta_j}^2 \leq c$. Therefore, the ridge estimate is obtained at the point at which the ellipse touches the circumference of the circle.

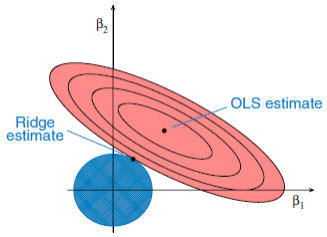


Figure 1: Geometric interpretation of ridge regression (from STAT 897D [1])

2.5 The ridge constant

In the following section we discuss different methods proposed by many researchers for selecting the best value for the ridge constant:

The ridge trace [5] is a two-dimensional plot of the coefficient estimates as a function the ridge constant (k). Essentially, we use the ridge trace to help us visualize where the estimates stabilize. From the ridge trace we would select the smallest value of the ridge constant for which stabilization occurs, since the ridge constant (normally lies between 0 and 1) is related to the amount of bias introduced.

Hoerl *et al.* [7] suggested that the ridge constant can be estimated by using the formula:

$$\hat{k} = \frac{p\hat{\sigma}^2}{\hat{\boldsymbol{\beta}}_{ols}`\hat{\boldsymbol{\beta}}_{ols}}$$

where p represents the number of predictors and $\hat{\sigma}^2$ is the estimated variance. This method is well known as the fixed point method. They argued that using this method is the best choice for the ridge constant because it obtains the minimum variance.

Hoerl and Kennard [6] proposed the iterative procedure. In this method the ridge constant is calculated as:

$$\hat{k}_i = \frac{p\hat{\sigma}^2}{\sum_{j=1}^{p} (\hat{\beta}_j(\hat{k}_i - 1))^2}$$

for $i \geq 1$ until the difference between successive estimates $\hat{k}_i$ of k is relatively small and insignificant.

Lawless and Wang [9] suggested using this formula for selecting the ridge constant:

$$\hat{k} = \frac{p\hat{\sigma}^2}{\sum_{i=1}^{p} \lambda_i \hat{\beta}_i^2}$$

where λ_i is the i^{th} eigenvalue of the matrix $\mathbf{X}'\mathbf{X}$. This method is a modifies the fixed point method, by multiplying the denominator of the fixed point method with the eigenvalues.

Mallows [10] modified the C_p statistic to a C_k statistic. The C_p statistic is defined as:

$$C_p = \frac{\text{RSS}}{\text{MSR}} - (n - 2p) + 1$$

where MSR is the residual mean square for the model.

The C_k statistic suggested is computed as follows:

$$C_k \quad = \quad \frac{\text{RSS}}{\hat{\sigma}^2} - (n-2) + 2\text{Trace}(\mathbf{XL})$$

where $\mathbf{L} = (\mathbf{X}'\mathbf{X} + k\mathbf{I})^{-1}\mathbf{X}'$. Therefore, minimizing C_k statistic will give us the best value of the ridge constant.

McDonald and Galarneau [11] suggested the following method. Let G be equal:

$$\text{G} \quad = \quad \hat{\boldsymbol{\beta}}'_{ols}\hat{\boldsymbol{\beta}}_{ols} - \hat{\sigma}^2 \sum_{j=1}^{p} \left(\frac{1}{\lambda_i}\right)$$

Then an estimator of the ridge constant is selected by solving the following equation:

$$\hat{\boldsymbol{\beta}}'_{ridge}\hat{\boldsymbol{\beta}}_{ridge} \quad = \quad \text{G} \qquad \text{if G} > 0$$

Otherwise if G < 0 choose $k = 0$ or if G $= 0$ choose $k = \infty$.

Khalaf and Shukur [8] proposed the following method:

$$k \quad = \quad (\lambda_{max}\hat{\sigma}^2)/((n-p-1)\hat{\sigma}^2 + \lambda_{max}\hat{\beta}^2_{max}$$

where λ_{max} is the maximum eigenvalue of the matrix $\mathbf{X}'\mathbf{X}$.

2.6 Variations and developments

Tibshirani [13] introduced the least absolute shrinkage and selection operator (LASSO) as a method of estimation following the work on ridge regression. The LASSO reduces the dramatic variation of the OLS estimator by shrinking some of the correlated coefficients to exactly zero. The LASSO estimator is obtained by minimizing the RSS subject to the constraint:

$$\sum_{j=1}^{p} |\beta_j| \quad \leq \quad c$$

The constraint shows that the LASSO produces some coefficient estimates that are exactly zero and hence improving both prediction accuracy and model interpretability.

The main difference between ridge regression and the LASSO is that in ridge regression, the OLS estimates are shrunk towards zero whereas with the LASSO some OLS estimates become exactly zero. Nevertheless, both ridge regression and LASSO introduce a small bias to improve the OLS estimates.

One of the limitations of the LASSO is that it fails to do a group selection when there are groups of strongly correlated independent variables. It only selects one variable from a group and ignores the other variables. Zou and Hastie [14] introduced the elastic net to overcome the limitations of the LASSO. The elastic net is a method used to group strongly correlated predictors. In addition, the elastic net estimator is obtained by minimizing the RSS subject to the ridge and LASSO constraints:

$$\beta_{enet} \quad = \quad ||y_i - \sum_{j=1}^{p} \beta_j x_j)^2||$$

subject to $\sum_{j=1}^{p} {\beta_j}^2 \leq c$ and $\sum_{j=1}^{p} |\beta_j| \leq c$.

The penalty part:

$$\sum_{j=1}^{p} |\beta_j| \quad \leq \quad c$$

performs variable selection by setting some coefficients to exactly zero.

The penalty part:

$$\sum_{j=1}^{p} {\beta_j}^2 \quad \leq \quad c$$

encourages the group selection by shrinking the coefficients of correlated variables towards zero.

3 Application

Suppose that a researcher is interested in determining the relationship between high blood pressure (Y in mm Hg) and weight (X_1 in kg), body surface area (X_2 in m^2) and age (X_3 in years) of 20 workers at Pick 'n Pay in Hatfield Plaza.

To detect the presence of multicollinearity in the data set, we first examine the correlation matrix by using the PROC CORR procedure in SAS [2] (see Appendix A) which yields the following output.

Pearson Correlation Coefficients, N = 20
Prob > |r| under H0: Rho=0

	Y	X1	X2	X3
Y	1.00000	0.84438 <.0001	0.87911 <.0001	0.14355 0.5512
X1	0.84438 <.0001	1.00000	0.92495 <.0001	0.45889 0.0435
X2	0.87911 <.0001	0.92495 <.0001	1.00000	0.08578 0.7337
X3	0.14355 0.5512	0.45889 0.0435	0.08578 0.7337	1.00000

From the output above it is very clear that multicollinearity is present, since the correlation between weight (X_1) and body surface area (X_2) is relatively high ($r = 0.92495$).

The following output below gives the OLS estimates by using the PROC REG procedure in SAS [2].

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	117.08571	99.78351	1.28	0.2789
X1	1	4.33511	3.01662	1.55	0.1711
X2	1	-2.85795	2.58313	-1.22	0.2951
X3	1	-2.18707	1.59661	-1.48	0.1917

Looking at the $p-$values of the output above we see that the all the OLS estimates are statistically insignificant at a 5% level of significance. In this example $\beta_2 = -2.86$, this means that high blood pressure (Y) is expected to decrease by 2.86 mm Hg when the body surface area (X_2) increases by one m^2, holding weight (X_1) and age (X_3) constant. Whereas one would be expecting a positive relationship between high blood pressure (Y) and body surface area (X_2). This shows that the presence of multicollinearity can result in highly unstable estimates and coefficients appear to have the wrong sign.

To apply ridge regression we use the PROC REG procedure in SAS [2] with the RIDGE option to get the ridge estimator and RIDGEPLOT option to plot the ridge trace (see Appendix B). The results are given below:

Parameter Estimates

Variable	DF	Parameter Estimate
Intercept	1	-7.403
X1	1	0.555
X2	1	0.368
X3	1	-0.192

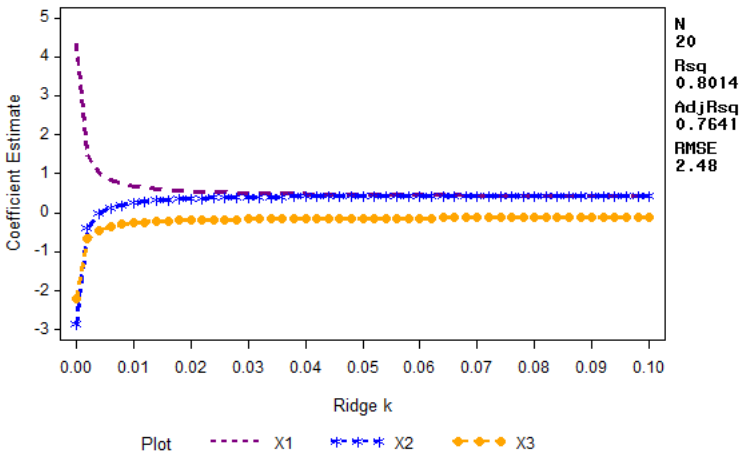


Figure 2: Ridge trace

The value of the ridge constant is chosen by looking at the variance inflation factors (VIF) that is very close to one. From the output in Appendix B we see that when k=0.020 with the VIF values 1.1031, 1.081 and 1.011 the ridge estimates obtained. The ridge estimates are $\beta_1 = 0.555$, $\beta_2 = 0.368$ and $\beta_3 = -0.192$. The OLS estimates have been shrunk towards zero and the negative sign on β_2 is removed. Therefore, applying ridge regression to the model results in more accurate and stable estimates.

Figure 2 demonstrates the ridge trace for this example with the coefficient estimates on the vertical axis and various values of the ridge constant along the horizontal axis. From the ridge trace we see that when $k = 0$ there is a huge variation between the OLS estimates. However, as the ridge constant increases slowly from zero, the coefficients seem to settle down and gradually drift towards zero. From this graph, the best choice of the ridge constant that can be selected is when $k = 0.02$, since it is the smallest value of the ridge constant where the coefficients are relatively stable.

The LASSO estimates are obtained by using the PROC GLMSELECT procedure in SAS [2] (see Appendix C).

Parameter Estimates

Variable	DF	Parameter Estimate
Intercept	1	6.565371
X1	1	0.982820
X2	1	0
X3	1	-0.406975

The results above show that applying the LASSO method to the observed data, sets some of the highly correlated explanatory variables to exactly zero as discussed earlier. In this example weight (X_1) and body surface area (X_2) are highly correlated and the LASSO has selected body surface area (X_2) as the only independent variable to be equal zero.

To do an application of the elastic net we use PROC GLMSELECT in SAS [2] with the SELECTION=ELASTICNET option (Appendix D).

Elastic Net Selection Summary

Step	Effect Entered	Effect Removed	Number Effects In	ASE	Validation ASE
0	Intercept		1	24.7695	24.7695
1	X2		2	10.1821	10.1821
2		X1	3	5.5018	5.5018
3	X3		4	5.3024	5.3024
4		X2	3	5.2972	5.2972
5	X2		4	5.0939	5.0939*

* Optimal Value of Criterion

The output above shows that the elastic net method groups the highly correlated variables, weight (X_1) and body surface area (X_2) as the variables to be removed from the model.

4 Conclusion

In this paper, we had a thorough discussion on ridge regression. Ridge regression introduces a small bias in coefficient estimation by continuously shrinking the least squares estimates to obtain improved parameter estimates. Properties of the ridge estimator and various methods suggested for selecting the optimal value of the ridge constant were presented. Furthermore, we also discussed alternative methods such as the LASSO and elastic net to improve the quality of prediction when multicollinearity is present. The results from the application have shown that ridge regression does very well in shrinking the OLS estimates to obtain better estimates and more interpretable results.

References

[1] STAT 897D, Applied Data Mining and Statistical Learning, digital image, The Pennsylvania State University, accessed 2 June 2016,<https://onlinecourses.science.psu.edu/stat857/node/155>.

[2] The data analysis for this essay was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA.

[3] D.N Gujarati and D.C Porter. *Basic Econometrics*. McGraw-Hill Education, 2009.

[4] R.R Hocking, F.M Speed, and M.J Lynn. A class of biased estimators in linear regression. *Technometrics*, 18(4):425–437, 1976.

[5] A.E Hoerl and R.W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[6] A.E Hoerl and R.W Kennard. Ridge regression iterative estimation of the biasing parameter. *Communications in Statistics-Theory and Methods*, 5(1):77–88, 1976.

[7] A.E Hoerl, R.W Kennard, and K.F Baldwin. Ridge regression: Some simulations. *Communications in Statistics-Theory and Methods*, 4(2):105–123, 1975.

[8] G. Khalaf and G. Shukur. Choosing ridge parameter for regression problems. *Communications in Statistics.-Theory and Methods*, 34:1177–1182, 2005.

[9] J.F Lawless and P. Wang. A simulation of ridge and other regression estimators. *Communications in Statistics-Theory and Methods*, AS(4):307–323, 1976.

[10] C.L Mallows. Some comments on Cp. *Technometrics*, 15(4):661–675, 1973.

[11] G.C McDonald and D.I Galarneau. A monte carlo evaluation of some ridge-type estimators. *Journal of the American Statistical Association*, 70(350):407–416, 1975.

[12] C. Samprit and A.S Hadi. *Regression Analysis by Example*. John Wiley & Sons, 2015.

[13] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[14] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Appendix

Appendix A

```
data Income;
infile 'C:\Users\LEBOGANG\Desktop\RAL 780\Ex 6\CH07TA01.txt';
input X1-X3 Y;
run;

proc corr data=Income;
var Y X1-X3;
run;
```

Appendix B

```
proc reg data=Income outest=ridge outvif ridge=0 to 0.1 by 0.002;
model Y = X1-X3;
plot / ridgeplot nomodel;
run;

proc print data=ridge;
run;

data new;
set ridge;
if _type_='RIDGEVIF';
run;

proc sort data=new;
by _type_;
run;

goptions reset=all i=join;
axis1 label=(angle=90 'Standardized coefficients') order = 0 to 60 by 5;
axis2 label=('Ridge constant k') minor=(number=4) order = 0 to 0.1 by 0.002;
legend1 label=('Plot:') value=('X1' 'X2' 'X3' );
symbol1 color=purple line=3 width=2;
symbol2 color=blue value=star width=2;
symbol3 color=orange value=dot width=2;

proc gplot data= new;
by _type_;
plot (X1-X3)*_RIDGE_/overlay legend=legend1 vaxis=axis1 haxis=axis2;
run;
```

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	396.98461	132.32820	21.52	<.0001
Error	16	98.40489	6.15031		
Corrected Total	19	495.38950			

Root MSE	2.47998	R-Square	0.8014
Dependent Mean	20.19500	Adj R-Sq	0.7641
Coeff Var	12.28017		

RIDGE	_RMSE_	Intercept	X1	X2	X3
PARMS	.	2.47998	117.086	4.335	-2.858	-2.187
RIDGEVIF	0.000	.	.	708.843	564.343	104.606
RIDGE	0.000	2.47998	117.086	4.335	-2.858	-2.187
RIDGEVIF	0.002	.	.	50.559	40.448	8.280
RIDGE	0.002	2.54921	22.277	1.464	-0.401	-0.674
RIDGEVIF	0.004	.	.	16.982	13.725	3.363
RIDGE	0.004	2.57173	7.725	1.023	-0.024	-0.441
RIDGEVIF	0.006	.	.	8.503	6.976	2.119
RIDGE	0.006	2.58174	1.842	0.844	0.128	-0.346
RIDGEVIF	0.008	.	.	5.147	4.305	1.624
RIDGE	0.008	2.58739	-1.331	0.746	0.210	-0.294
RIDGEVIF	0.010	.	.	3.486	2.981	1.377
RIDGE	0.010	2.59104	-3.312	0.685	0.262	-0.262
RIDGEVIF	0.012	.	.	2.543	2.231	1.236
RIDGE	0.012	2.59360	-4.661	0.643	0.297	-0.239
RIDGEVIF	0.014	.	.	1.958	1.764	1.146
RIDGE	0.014	2.59551	-5.637	0.612	0.322	-0.223
RIDGEVIF	0.016	.	.	1.570	1.454	1.086
RIDGE	0.016	2.59701	-6.373	0.589	0.341	-0.210
RIDGEVIF	0.018	.	.	1.299	1.238	1.043
RIDGE	0.018	2.59822	-6.946	0.570	0.356	-0.200
RIDGEVIF	0.020	.	.	1.103	1.081	1.011
RIDGE	0.020	2.59924	-7.403	0.555	0.368	-0.192
RIDGEVIF	0.022	.	.	0.956	0.963	0.986
RIDGE	0.022	2.60011	-7.776	0.543	0.378	-0.185
RIDGEVIF	0.024	.	.	0.843	0.872	0.966
RIDGE	0.024	2.60087	-8.083	0.532	0.386	-0.179

Appendix C

```
proc glmselect data=Income plots=coefficient(stepaxis=normb unpack);
model Y = X1-X3 / selection=LASSO(stop=4);
run;
```

```
                          Analysis of Variance

                                Sum of           Mean
Source                  DF      Squares         Square    F Value
Model                    3    389.29126      129.76375      19.57
Error                   16    106.09824        6.63114
Corrected Total         19    495.38950

Root MSE              2.57510
Dependent Mean       20.19500
R-Square              0.7858
Adj R-Sq              0.7457
AIC                  63.37266
AICC                 67.65838
SBC                  45.35559
```

Appendix D

```
proc glmselect data=Income valdata=Income plots=coefficients;
model Y = X1-X3/ selection=elasticnet(steps=140 L2=0.02 choose=validate);
run;
```

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value
Model	3	393.51198	131.17066	20.60
Error	16	101.87752	6.36735	
Corrected Total	19	495.38950		

Root MSE	2.52336
Dependent Mean	20.19500
R-Square	0.7943
Adj R-Sq	0.7558
AIC	62.56078
AICC	66.84650
SBC	44.54371
ASE (Train)	5.09388
ASE (Validate)	5.09388

Parameter Estimates

Parameter	DF	Estimate
Intercept	1	-7.403425
X1	1	0.555353
X2	1	0.368144
X3	1	-0.191627

Does overconfidence influence the performance of first level statistics students at the University of Pretoria?

Innocentia Thandiwe Konyana 10325965

STK795 Research Report

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

Supervisors: L Fletcher and F Reyneke

Department of Statistics, University of Pretoria

2 November 2016

Abstract

Overconfidence can affect how students perform regardless of their abilities. This can prolong the time required by students to complete their studies as it can result in students having to repeat courses over and over again due to failure. The objective of this study is to determine the correlation between first year statistics students' level of overconfidence and their academic performance (pass or fail); which is to find whether there exist a relationship between students' poor performance and overconfidence. Statistics is a challenge to many students and often educators mainly focus on the cognitive side which includes the acquisition of skills and knowledge; little attention is given to the non-cognitive side which also affects students' performance-this includes feelings, beliefs, attitude, expectations, perception as well as motivation. These factors may affect the students' ability to learn statistics and make it difficult for them to gain a deeper understanding of statistics.

It is of vital importance to determine whether or not overconfidence contributes to poor pass rates of first year level statistics students in order to prevent the effect thereof and to ensure that the field of statistics at the University of Pretoria gains continuous excellent performance. The sample used in this study was 1157 students enrolled at the University of Pretoria for first level statistics (STK 110) for the first time in 2014. A logistic regression was used to model the data. The results indicate that overconfidence has an impact on students' performance in first level statistics, whilst mathematics was found not to be a significant determinant of performance in statistics. English was a significantly predictor of students' performance at a 10 % level.

Declaration

I, *Innocentia Thandiwe Konyana,* 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.

_____
Innocentia Thandiwe Konyana

_____
Dr L Fletcher, F Reyneke

_____
Date 2/11/16

Acknowledgements

The compilation of this paper would not be possible without the active participation of the team of insightful and rigorous supervisors who devoted their precious time to making this paper a success. It has been a privilege and a pleasure to work with both of them.

To Dr L Fletcher and Mrs F Reyneke who made time to fit me in their weekly routine with dedication and willingness, I give grateful acknowledgement of their contribution to this research on the effect of over-confidence in first level statistics students of the University of Pretoria.

The author gives acknowledgement to the University of Pretoria; particularly the department of statistics for the opportunity of being part of their honours programme for the year 2016.

The author would also like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a post graduate bursary.

Contents

1 Introduction **6**

2 Background Theory **7**
2.1 Logistic regression . . . 7
2.2 Odds . . . 8
2.3 Odds ratio . . . 8
2.4 Logit function . . . 8
2.5 Interpretimg the logistic coefficients . . . 8
2.6 Tests and confidence intervals for the model and parameters . . . 9
2.6.1 Deviance and likelihood ratio test . . . 9
2.6.2 Wald test . . . 9
2.7 Goodness of fit of the model . . . 9
2.7.1 The Hosmer-Lemeshow test . . . 9
2.7.2 Pseudo-R^2 for logistic regression . . . 9
2.7.3 Classification tables . . . 10

3 Application **10**
3.1 Data source . . . 10
3.2 Organizing the data set . . . 10
3.3 Definition of variables . . . 11
3.3.1 Dependent variable . . . 11
3.3.2 Predictor/independent variables . . . 11
3.4 Data exploration . . . 11
3.5 Correlation of predictor variables with the dependent variable . . . 12
3.6 Logistic regression . . . 12
3.6.1 Statistical results . . . 12

4 Conclusion **13**

Appendix **17**

List of Figures

List of Tables

1 Classification table . . . 10
2 Confidence index for each question in the pre-test . . . 10
3 Frequency table for categorical variables . . . 11
4 Descriptive Statistics . . . 11
5 Test of the null hypothesis $H_0 : \beta$=0 . . . 12
6 Estimated model parameters . . . 13
7 Classification Table . . . 13

1 Introduction

Overconfidence is a controversial subject that has been studied repeatedly in the past years; it is a global subject which has attained attention across various disciplines and the world at large. Evidence indicates that this phenomenon is not only a problem in statistics, for instance it has been identified in various domains such as in chemistry [31], business studies [19], financial markets [34], [13], [17], politics [42] and in driving [18], [12]. Given the above information it is evident that overconfidence is a major problem which if not addressed may in the long-run affect performance, which is not only determined by one's ability or intelligence but also determined by individual's perceptions and expectations.

Overconfidence has been studied over the years as the difference between individuals' expected performance and their actual performance [27]. Overconfidence entails the overestimation of one's abilities, performance as well as chances of succeeding [26]. Furthermore overconfidence is defined as both over-optimism and over-precision, that is, individuals who are over-optimistic tend to overestimate their abilities either absolutely or comparing themselves with others [26]. Moreover individuals who are over-precise (i.e. people who often tend to be excessively certain regarding the accuracy of their beliefs) often do not consider uncertainty.

Overconfidence is also revealed when people tend to see themselves as better than others [6] and when people inaccurately access themselves, for instance people may overestimate their reasoning abilities, driving skills [18], [46] as well as their grammar knowledge [21]. In addition people also tend to underestimate the effort and time needed to complete tasks [3].

A study by [31] has shown that poor performance doesn't necessary mean inability; it can be dealt with if help is provided in time. Hence it is of vital importance to address the problem of overconfidence as this can help individuals to improve and do better in future. For the purpose of this study we will focus on the analysis of overconfidence particularly in statistics.

Previous studies have shown that overconfidence could to a certain degree be attributed to poor pass rates of first level students and can lead to changes in university courses [40], [35]. Students tend to be overconfident when it comes to their academic abilities [14], [28], [15], that is, students often fail to distinguish between what they know and what they do not know and often incorrectly assess themselves. Failure of students to accurately evaluate themselves often leads to poor academic performance [19]. There exist a negative relationship between students' perceived marks and actual marks [23]. That is, students who were highly confident about their marks actually performed poor and this is due to the fact that students who think highly of themselves tend to allocate less time to study. The study by [20] shows similar results, students were asked what marks they think they will get, then their actual marks were compared with the marks they expected and the study revealed that most of the students got marks lower than they had predicted.

In addressing the problem of students inaccurately evaluating themselves, [31] conducted a study to identify high risk students in a first year chemistry module. In this study it was found that the identification of high risk students at an early stage can to some extent reduce poor performance, as well as university drop-outs by giving the students appropriate assistance in time. In addition to the identification of high risk students [36] conducted a study to monitor accuracy, where he focused on relative accuracy. The study revealed that calibration (over/under confidence) had an effect in the total amount of time student allocate for studying. That is an overconfident student might feel prepared enough and stop studying prematurely and thus attain poor grades. The study by [36] focused on developing intervention measures that could help students better evaluate themselves, this study indicates that the intervention appears to assist students' in distinguishing between information they know and that which still needs additional study.

Different people exhibit different levels of overconfidence, [22] distinguished between high and low performing students and found that high performing students often exhibit lower levels of overconfidence, since they appear to be able to better distinguish between what they know and what they do not know as opposed to low performing students. Supporting this is the study by [14] which reveals that students who had high

Stochastic Achievement Test (SAT) scores were more accurate than those with low SAT.

Overconfidence further varies with gender; [2] has shown that gender differences in confidence are dependent on the context as well as the domain being tested, that is, it normally depends on the kind of questions asked. For instance [4] found that women appear to be less confident than men in subjects such as mathematics, solving complex problems and science. In terms of perceived performance males tend to be more overconfident when they are incorrect than when they are correct and the opposite holds for females [22]. Thus [4] and [22] concluded that there is no evidence that self-confidence can be translated to real academic achievement. Furthermore other studies reveal that gender is not related to performance [16].

To extend on the relationship between overconfidence and performance, [27] examined the relationship between self-efficacy (which refers to one's belief in their ability to succeed in a specific situation or task accomplishment) and performance. This study revealed a positive relationship between self-efficacy and performance, when considering the level of over-and-under confidence, overconfidence led to a negative relationship, which indicates that overconfidence results to poor performance. Supporting these literatures are, [44] who found that individuals who exhibited high self-efficacy tend to experience reduction in motivation which in turn has a negative effect on performance. These findings further supports the perpetual control theory by [32], which states that high self-efficacy, may cause a person to prematurely believe that their goals state has been reached, which may lead to a reduction in effort and thus performance.

Performance is not only affected by overconfidence, it can also be negatively affected by attitude and perception [45], [10]. Students' perception as well as their attitude towards statistical courses can affect their performance [38], these studies reveals that students' tend to have a negative attitude towards statistics or it tends to develop overtime with class attendance. Consequently they see statistics as a barrier between them and their qualifications [9], [29] and this affects their performance [30]. Students also tend to develop anxiety when they learn that they will attend statistical undergraduate courses [38], thus this anxiety affects how they perform in those courses. Students even go to the extreme of renaming the statistics course "sadistic" [37].

Expectations are also determinants of student's performance; in a study by [8] it was found that first year students may have un-realistic expectations about their academic performance which tend to reduce chances of them being successful in their studies. Evidence further indicates that students' performance can be affected by factors such as teaching strategies [1], their level of motivation [41], students approach to studying [25], students' balance between academic and social life [43], students' effort [39] and psychological factors [24]. The study by [24] further revealed that students' perception is also a factor affecting their performance, that is, students' perception of what will increase or decrease their chances of success have a strong influence on how they behave. For instance if a student believes that attending classes will help them pass they will attend classes regularly to increase their chances of success.

This paper will examine the relationship between first year statistics students' level of overconfidence and their academic performance (pass or fail), that is to find whether there exist a relationship between students' poor performance and overconfidence. Data from the University of Pretoria's 2014 first year statistics student will be utilised and a regression logistic model will be used to conduct the analysis.

2 Background Theory

2.1 Logistic regression

The logistic regression model is used to explain the relationship between a dependent binary variable and one or more continuous/categorical independent variables. The dependent variable takes the values 1 with probability ϑ or 0 with probability 1-ϑ. Logistic regression provides the odds of a successful event, i.e. the probability of success divided by the probability of failure and the results are provided in the form of

odds ratio. The logistic regressions goal is to find the best fitting model that will describe the relationship between the dependent binary variable and independent variables (predictors). Multinomial regression is not appropriate as it will violate the assumptions of normality of the responses and homoscedasticity of the residuals. Furthermore the discriminant analysis which can also be used to classify categories is also not appropriate since the assumption of multivariate normality of the independent variables cannot be satisfied with categorical prediction [7], [33].

2.2 Odds

The odds of an event defines how likely an event will occur and it's calculated as follows:

$$odds = \frac{p}{1-p}$$

where p repesents the probability of an event occurring and 1- p being the probability that the event doesn't occur. The odds of an event occurring are difficult to interpret hence the use of odds ratio is preferred when interpreting the estimated parameter coefficients.

2.3 Odds ratio

An odds ratio measures the relationship between an outcome and an exposure. It is a representation of an outcome occuring given a particular exposure and is denoted as follows,

$$\frac{P(\text{successs A})}{P(\text{failure A})} \div \frac{P(\text{success B})}{P(\text{failure B})}$$

which is the ratio of success for a patiular group divided by the success of another group.

2.4 Logit function

The logit function gives an estimation of the probability that a particular event occurs. It is usually referred to as a link function and it is defined as,

$$logit(p) = ln(\frac{p}{1-p})$$

which measures the log of the odds of the event occurring. Thus the resulting logistic regression model is as follows:

$$ln(\frac{p}{1-p}) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots\ldots\ldots + \beta_p x_{ip} + \varepsilon \tag{1}$$

The logistic regression model does not use ordinary least squares but rather uses the maximum likelihood estimation to solve for the parameter estimate that best fits the data.

2.5 Interpretimg the logistic coefficients

To interpret the parameter estimators (β's), equation 1 is converted back to odds by exponentiating both sides of equation (1) which yields the following:

$$\frac{\hat{p}}{1-\hat{p}} = e^{b_0} e^{b_1 x_{i1}} e^{b_2 x_{i2}} \ldots\ldots\ldots e^{b_p x_{ip}}$$

The odds of an event occurring are increased by a multiplicative factor of e^{b_1} for a one-unit change in x_{i1}. In order to get the odds ratios, the parameter estimates (β's) are exponentiated. Therefore the β' s are the log-odds, negative coefficient values implies that the odds ratios are less than 1 meaning that the outcome event is less likely to occur, whilst positive coefficient values implies that the odds ratios are more than 1 thus the outcome event is more likely to occur.

2.6 Tests and confidence intervals for the model and parameters

2.6.1 Deviance and likelihood ratio test

Maximum likelihood is used to find the best fitting line by finding the smallest possible deviance between the observed and predicted values. The deviance is a measure of the lack of fit of the data in a logistic regression model, once the maximum likelihood has identified the best solution, it assigns a value for the deviance which is referred to as "negative two log likelihood" (-2 log likelihood). The deviance statistic is calculated as follows:

$$D = -2ln(\frac{\text{likelihood of the fitted model}}{\text{likelihood of the saturated model}})$$

the equation above represents the likelihood-ratio test. D follows approximately a chi-squared distribution, with small values indicating better fit since the fitted model deviates less from the saturated model.

2.6.2 Wald test

A Wald test measures the statistical significance of each coefficient β's in the model. It then computes a Z statistic which is denoted as follows:

$$z = \frac{\hat{\beta}}{SE_{\hat{\beta}}}$$

with SE denoting some estimate of the standard error of $\hat{\beta}$ which may be the maximum likelihood estimator. The 95% confidence interval is

$$\hat{\beta} \pm 1.96SE(\hat{\beta})$$

2.7 Goodness of fit of the model

A goodness of fit measure is used to determine how well the model fits the data, that is, are the values predicted close to the values observed.

2.7.1 The Hosmer-Lemeshow test

The Hosmer-Lemeshow statistic provides an evaluation of how well the model fits the data by means of a chi-square statistic. The Hosmer-Lemeshow does not limit the number of explanatory variables (continuous or categorical). The Hosmer-Lemeshow is denoted as follows:

$$\sum_{i=1}^{g} \sum_{j=1}^{2} = \frac{(obs_{ij} - exp_{ij})^2}{exp_{ij}}$$

g= number of groups, obs_{ij}= observed values, exp_{ij}= expected values. The test uses a chi-square with g-2 degrees of freedom.

2.7.2 Pseudo-R^2 for logistic regression

Several statistics in logistic regression have been developed to measure whether the explanatory variables can successfully predict the dependent variable and these statistics are the same as the linear regression coefficient of determination R^2. The frequently used statistics are the Cox & Snell and the Nagelkerke R^2 which attains a maximum value that is less than 1. Furthermore the Nagelkerke R^2 is Cox & Snell R^2 adjusted version which has values that ranges between 0, thus it is often preferred.

2.7.3 Classification tables

The classification table indicates the comparison of the number of successes predicted by the logistic regression model compared to the number actually-observed and similarly the number of failures predicted by the logistic regression model to the number actually-observed.

	Success observed	Failure observed	
Success predicted	True Positive	False Positive	Predicted Positive
Failure predicted	False Negative	True Negative	Predicted Negative
	Observed Positive	Observed Negative	Total sample size

Table 1: Classification table

The overall accuracy of the logistic regression is a measure of the fit of the model and it is defined as:

$$Accuracy = (\frac{\text{True positive}}{\text{True negative}}) \div Totalsamplesize$$

3 Application

3.1 Data source

The data used in this study was collected during the commencement of STK 110 lectures in 2014, with the aim of understanding whether students are able to identify information that they know versus that which they do not know. The study focused on students who registered for STK 110 for the very first time and the data was collected by means of a pre-test. The pre-test consisted of 16 multiple choice questions that where purely based on basic mathematics operations and basic statistics understanding. Furthermore, students' had to give a rating on how confident they were in the answers they provided (Table 2), thus in total the test had 32 questions. Students' final matric results of mathematics and English, as well as information on gender, language and preferred language of instruction were obtained from the Bureau for Institutional Research and Planning (BIRAP). The actual performance of students was obtained by allocating a value 1 for the correct answer and 0 for an incorrect answer to the questions. The expected performance was obtained from the confidence data collected from the test (Table 2). This confidence data was scored as follows: 0 if a student chose A or B and 1 for C or D; it was based on the assumption that students either expected to be incorrect or correct.

Totally guessed answer	Almost a guess	Almost certain	Certain
A	B	C	D

Table 2: Confidence index for each question in the pre-test

3.2 Organizing the data set

For the purpose of this study the following students were considered: students who were taking STK 110 for the first time in the year 2014, students who had a minimum of 60% for their final grade 12 mathematics results and students who wrote the final STK 110 examination. The total number of students included in this study is 1157. The data set comprised of the following variables: home language; preferred language of instruction; mathematics and English final matric examination results; STK 110 final results; actual performance achieved for the pre-test, as well as the expected performance achieved for the pre-test.

3.3 Definition of variables

3.3.1 Dependent variable

The dependent variable STK110_bi was derived from students' final mark obtained for the module STK 110. This variable was created by assigning a value 1 (pass) for marks greater than or equal to 50 and a value 0 (fail) for marks less than 50.

3.3.2 Predictor/independent variables

The following predictor variables were considered:

- Mathematics: Final mathematics results achieved in matric
- English: Final English results achieved in matric
- Actual: Students' actual performance achieved for the pre-test (%)
- Expected: students expected performance achieved for the pre-test (%)
- Overconfidence: computed as the difference between the expected and actual performance, expressed as a percentage of the total number of questions in the test.
- Ratio: Computed as the expected test performance divided by the actual test performance
- Gender: Derived by assigning a value 1 if the student is a female and 0 if is a male
- Language: Derived as 1 if home language is the same as preferred language of instruction and 0 otherwise.

Variables	Categories	Frequency	Percentage
STK 110_bi	1= Pass	137	11.84
	0= Fail	1020	88.16
Gender	1=Female	535	46.24
	0= Male	622	53.76
Language	1= Home= Instruction	487	42.09
	0= Home≠Instruction	670	57.91

Table 3: Frequency table for categorical variables

3.4 Data exploration

The descriptive statistics results in Table 4 indicate that the average of students' expected performance is more than 10% higher than their average actual performance. The huge difference in the two means indicates the extent of overconfidence (misjudgement) observed amongst students. The mean of Ratio 1.25 indicates that on average students overestimated their true performance by 25%.

	Minimun	Maximum	Mean	Standard Deviation
English	45	99	74.94	7.55
Mathematics	60	99	75.37	9.29
STK 110	18	99	62.28	15.70
Actual	12.5	100	62.09	14.41
Expected	0	100	74.07	17.19
Overconfidence	-50	75	11.97	17.51
Ratio	0	6	1.25	0.41

Table 4: Descriptive Statistics

3.5 Correlation of predictor variables with the dependent variable

Since the dependent variable is binary, a point biserial correlation was calculated between the dependent variable and each of the explanatory variables in order to measure how strong is the relationships between them. The correlation ranges between -1 and 1, where -1 indicates a perfect negative correlation and 1 indicates a perfect positive correlation and 0 indicating no correlation at all. The correlations were conducted in order to determine which variables contribute significantly in predicting the dependent variable as well as to determine any correlations amongst the explanatory variables. Most importantly the correlation analysis was conducted in order to identify which overconfidence (Overconfidence or Ratio) index to use in the prediction model.
According to the correlation output both the confidence indices (Ratio and Overconfidence) have a negative relationship with the dependent variable (STK110_bi) as hypothesised. This indicates that higher levels of overconfidence can negatively affect performance. These indices are highly correlated therefore cannot be used together in the prediction model. Since Ratio has the strongest correlation with the dependent variable it will be used in the prediction model. The correlation between actual performance and the dependent variable is positive, indicating that the better the student performed in the test, the higher their chance of passing the module. Positive correlations also exist between English and STK110_bi and between mathematics and STK110_bi, indicating that better performance in both English and mathematics may increase the probability of passing STK 110.

3.6 Logistic regression

A binary logistic regression model was used to analyse the data since the underlying dependent variable, STK110_bi is dichotomous. The binary dependent variable follows a binomial distribution and the log of the odds of passing STK 110 will be modelled as a function of the explanatory variables, mathematics, English, language, gender and Ratio.
The binary logistic regression model is given by:

$$ln(\frac{p}{1-p}) = 0.3372 - 0.7592Ratio + 0.0112Mathematics + 0.0202English + 0.2989Language + 0.2881Gender$$

where p is the probability of passing STK110.

3.6.1 Statistical results

The data analysis for this essay was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA.
SAS output is attached in the appendix.

Statictic	DF	Chi-square	Pr ≻Chi-square
-2 Log (Likelihood)	5	20.063	≺ 0.0001
Score	5	31.814	≺ 0.0001
Wald	5	27.220	≺ 0.0001

Table 5: Test of the null hypothesis $H_0 : \beta$=0

The likelihood ratio chi-square of 20.063 with a p-value of ≺ 0.0001 indicates that the model as a whole fits significantly better than an intercept model. However, the pseudo -R^2 is very low at 0.0240, indicating that only a small portion of the variability in the odds of passing STK110 can be explained by this model.

Source	Coefficient	Standard error	Wald Chi-sq	Pr $\succ$Chi-sq	Odds ratio
Intercept	0.3372	1.1746	0.0824	0.7741	
Ratio	-0.7592	0.1969	14.8727	0.0001	0.468
Mathematics	0.0112	0.0102	1.2085	0.2716	1.011
English	0.0202	0.0122	2.7370	0.0980	1.020
Language	0.2989	0.1879	2.5303	0.1117	1.348
Gender	0.2881	0.1865	2.3871	0.1223	1.334

Table 6: Estimated model parameters

The coefficient for Ratio is highly significant ($p \prec 0.0001$) and the coefficient of English is statistically significant at a 10% level. None of the other coefficients are significant.

The odds ratio of the significant predictors can be interpreted as follows:

- For every unit increase in Ratio, the odds of passing STK 110 (versus failing) are less than 0.5 for students who are overconfident, compared to those who are not overconfident. In other words, Overconfident students' are more than twice as likely to fail.
- For every unit (one percent) increase in English, the odds of passing STK 110 (versus falling STK 110) increases by a factor of 0.02.

		Fail	Pass	% correct
STK110_bi	Fail	1	136	7
	Pass	1	1019	99.9
Overall %				88.2

Table 7: Classification Table

According to table 7, the overall accuracy of the model is 88.2%. The model is expected to correctly predict pass since the majority of students passed, but the model performs poorly in predicting students who failed.

4 Conclusion

In this study the effect of overconfidence on performance of first level statistics students was measured. It was expected that the grade 12 mathematics results would be a core contributor to predict the STK 110 pass rate. However according to the binary logistic model; it did not have a significant influence on the passing of the 2014 cohort of STK 110 students. The only two significant contributors were Ratio ($p \prec 0.0001$) and English ($p \prec 0.1$) while the other variables did not contribute significantly to students' performance. The non-significance of language is contrary to the finding by [11], which shows that language does have an association with students' performance. Although the results further indicates that gender does not have a significant effect on performance, it played a substantial role in overconfidence, showing that females were less confident than males ($p \prec 0.05$). Furthermore it was found that students with low pre-test scores appeared to be more overconfident than those with high pre-test scores. Some high-ability students showed under-confidence.

Self-motivation and confidence should not be misinterpreted for overconfidence, as students who are confident and motivated can actually perform well. Early intervention can help students to perform better and overcome the problem of overconfidence. Interventions such as the pre-test at the commencement of lectures and letting students to provide reasons for their answers may help students to prepare better for their assessments. Furthermore this will allow students to recognize their own level of skills and thus develop and improve those skills, as [5] has indicated that students overestimate their performance not because of cognitive competence. The problem of overconfidence can be conquered if students and lectures work together.

References

[1] David E Bartz and Laura K Miller. *12 Teaching Methods To Enhance Student Learning. What Research Says to the Teacher.* ERIC, 1991.

[2] Richard T Bliss and Mark E Potter. Mutual fund managers: does gender matter? *The Journal of Business and Economic Studies*, 8(1):1, 2002.

[3] Roger Buehler, Dale Griffin, and Michael Ross. Exploring the" planning fallacy": Why people underestimate their task completion times. *Journal of Personality and Social Psychology*, 67(3):366, 1994.

[4] Nancy K Campbell and Gail Hackett. The effects of mathematics task performance on math self-efficacy and task interest. *Journal of Vocational Behavior*, 28(2):149–162, 1986.

[5] Dennis E Clayson. Performance overconfidence: metacognitive effects or misplaced student expectations? *Journal of Marketing Education*, 27(2):122–129, 2005.

[6] David Dunning, Chip Heath, and Jerry M Suls. Flawed self-assessment implications for health, education, and the workplace. *Psychological Science in the Public Interest*, 5(3):69–106, 2004.

[7] Bradley Efron. The efficiency of logistic regression compared to normal discriminant analysis. *Journal of the American Statistical Association*, 70(352):892–898, 1975.

[8] William J Fraser and Roy Killen. Factors influencing academic success or failure of first-year and senior university students: do education students and lecturers perceive things differently? *South African Journal of Education*, 23(4):254–263, 2003.

[9] Iddo Gal and Lynda Ginsburg. The role of beliefs and attitudes in learning statistics: Towards an assessment framework. *Journal of Statistics Education*, 2(2):1–15, 1994.

[10] Joan Garfield. How students learn statistics. *International Statistical Review/Revue Internationale de Statistique*, pages 25–34, 1995.

[11] Ans Gerber, Johann Engelbrecht, Ansie Harding, and John Rogan. The influence of second language teaching on undergraduate mathematics performance. *Mathematics Education Research Journal*, 17(3):3–21, 2005.

[12] John A Groeger and ID Brown. Assessing one's own and others' driving ability: Influences of sex, age, and experience. *Accident Analysis & Prevention*, 21(2):155–168, 1989.

[13] Michael D Grubb. Overconfident consumers in the marketplace. *The Journal of Economic Perspectives*, 29(4):9–35, 2015.

[14] Douglas J Hacker, Linda Bol, Dianne D Horgan, and Ernest A Rakow. Test prediction and performance in a classroom context. *Journal of Educational Psychology*, 92(1):160, 2000.

[15] Gail Hackett and Nancy E Betz. An exploration of the mathematics self-efficacy/mathematics performance correspondence. *Journal for Research in Mathematics Education*, pages 261–273, 1989.

[16] Plake B Harvey A and Wise S. The validity of six beliefs about factors related to statistics achievement. 1985.

[17] David R Just, Ying Cao, David Zilberman, et al. Risk, overconfidence and production in a competitive equilibrium. In *2009 Annual Meeting, July 26-28, 2009, Milwaukee, Wisconsin*, number 49161. Agricultural and Applied Economics Association, 2009.

[18] Laura E Knouse, Catherine L Bagwell, Russell A Barkley, and Kevin R Murphy. Accuracy of self-evaluation in adults with adhd evidence from a driving study. *Journal of Attention Disorders*, 8(4):221–234, 2005.

[19] Paul Sergius Koku and Anique Ahmed Qureshi. Overconfidence and the performance of business students on examinations. *Journal of Education for Business*, 79(4):217–224, 2004.

[20] Joachim Krueger and Ross A Mueller. Unskilled, unaware, or both? the better-than-average heuristic and statistical regression predict errors in estimates of own performance. *Journal of Personality and Social Psychology*, 82(2):180, 2002.

[21] Justin Kruger and David Dunning. Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of Personality and Social Psychology*, 77(6):1121, 1999.

[22] Mary A Lundeberg, Paul W Fox, and Judith Punćcohaŕ. Highly confident but wrong: gender differences and similarities in confidence judgments. *Journal of Educational Psychology*, 86(1):114, 1994.

[23] Edmond Marks. Student perceptions of college persistence, and their intellective, personality and performance correlates. *Journal of Educational Psychology*, 58(4):210, 1967.

[24] Kirsten McKenzie and Robert Schweitzer. Who succeeds at university? factors predicting academic performance in first year australian university students. *Higher Education Research and Development*, 20(1):21–33, 2001.

[25] JHF Meyer, P Parsons, and TT Dunne. Individual study orchestrations and their association with learning outcome. *Higher Education*, 20(1):67–89, 1990.

[26] Don A Moore and Paul J Healy. The trouble with overconfidence. *Psychological review*, 115(2):502, 2008.

[27] Trevor T Moores and Jerry Cha-Jan Chang. Self-efficacy, overconfidence, and the negative effect on subsequent performance: A field study. *Information & Management*, 46(2):69–76, 2009.

[28] Frank Pajares and M David Miller. Role of self-efficacy and self-concept beliefs in mathematical problem solving: A path analysis. *Journal of Educational Psychology*, 86(2):193, 1994.

[29] Jan Perney and Ruth Ravid. The relationship between attitudes toward statistics, math self-concept, test anxiety and graduate students' achievement in an introductory statistics course. 1990.

[30] Ivars Peterson. Pick a sample. *Science News*, 140(4):56–58, 1991.

[31] Marietjie Potgieter, Mia Ackermann, and Lizelle Fletcher. Inaccuracy of self-evaluation as additional variable for prediction of students at risk of failing first-year chemistry. *Chemistry Education Research and Practice*, 11(1):17–24, 2010.

[32] William T Powers. *Behavior: The Control of Perception (rev. & exp.)*. Benchmark Press, 2005.

[33] S James Press and Sandra Wilson. Choosing between logistic regression and discriminant analysis. *Journal of the American Statistical Association*, 73(364):699–705, 1978.

[34] Paul C Price and Eric R Stone. Intuitive evaluation of likelihood judgment producers: Evidence for a confidence heuristic. *Journal of Behavioral Decision Making*, 17(1):39–57, 2004.

[35] Joseph R Radzevick and Don A Moore. Competing to be certain (but wrong): Social pressure and overprecision in judgment. In *Academy of Management Proceedings*, volume 2009, pages 1–6. Academy of Management, 2009.

[36] Joshua S Redford, Keith W Thiede, Jennifer Wiley, and Thomas D Griffin. Concept mapping improves metacomprehension accuracy among 7th graders. *Learning and Instruction*, 22(4):262–270, 2012.

[37] Bill Rosenthal. No more sadistics, no more sadists, no more victims. *UMAP Journal*, 13:281–290, 1992.

[38] Julian L Simon and Peter Bruce. Resampling: A tool for everyday statistical work. *Chance*, 4(1):22–32, 1991.

[39] Endya B Stewart. School structural characteristics, student effort, peer associations, and parental involvement the influence of school-and individual-level factors on academic achievement. *Education and Urban Society*, 40(2):179–204, 2008.

[40] Ralph Stinebrickner and Todd Stinebrickner. A major in science? initial beliefs and final outcomes for college major and dropout. *The Review of Economic Studies*, pages 1–25, 2013.

[41] Gilles L Talbot. Personality correlates and personal investment of college students who persist and achieve. *Journal of Research & Development in Education*, 1990.

[42] Philip Tetlock. *Expert Political Judgment: How good is it? How can we know?* Princeton University Press, 2005.

[43] Vincent Tinto. Dropout from higher education: A theoretical synthesis of recent research. *Review of Educational Research*, 45(1):89–125, 1975.

[44] Jeffrey B Vancouver, Charles M Thompson, and Amy A Williams. The changing signs in the relationships among self-efficacy, personal goals, and performance. *Journal of Applied Psychology*, 86(4):605, 2001.

[45] LK Waters, Theresa Martelli, Todd Zakrajsek, and Paula M Popovich. Measuring attitudes toward statistics in an introductory course on statistics. *Psychological Reports*, 64(1):113–114, 1989.

[46] Allan F Williams. Views of us drivers about driving safety. *Journal of Safety Research*, 34(5):491–494, 2003.

Appendix

Descriptive statistics
The MEANS Procedure

Variable	Label	N	Minimum	Maximum	Mean	Std Dev
STK110	STK110	1157	18.0000000	99.0000000	62.2765774	15.7028389
English	English	1157	45.0000000	99.0000000	74.9351772	7.5453725
Mathematics	Mathematics	1157	60.0000000	99.0000000	75.3716508	9.2923579
Actual	Actual	1157	12.5000000	100.0000000	62.0894555	14.4052541
Expected	Expected	1157	0	100.0000000	74.0654710	17.1942640
Overconfidence	Overconfidence	1157	-50.0000000	75.0000000	11.9760156	17.5075558
Ratio	Ratio	1157	0	6.0000000	1.2451576	0.4088745

Pearson Correlation Coefficients, N = 1157
Prob > |r| under H0: Rho=0

	STK_110 _bi	Overconfidence	Ratio	Actual	Expected	Mathematics	English
STK_110_bi	1.00000	-0.09218	-0.14131	0.16953	0.04817	0.04807	0.05928
STK 110 bi		0.0017	<.0001	<.0001	0.1015	0.1022	0.0438
Overconfidence	-0.09218	1.00000	0.89241	-0.43295	0.65549	-0.05304	-0.02260
Overconfidence	0.0017		<.0001	<.0001	<.0001	0.0713	0.4424
Ratio	-0.14131	0.89241	1.00000	-0.55153	0.44660	-0.07539	-0.03086
Ratio	<.0001	<.0001		<.0001	<.0001	0.0103	0.2942
Actual	0.16953	-0.43295	-0.55153	1.00000	0.39695	0.09541	0.08392
Actual	<.0001	<.0001	<.0001		<.0001	0.0012	0.0043
Expected	0.04817	0.65549	0.44660	0.39695	1.00000	0.02593	0.04729
Expected	0.1015	<.0001	<.0001	<.0001		0.3782	0.1079
Mathematics	0.04807	-0.05304	-0.07539	0.09541	0.02593	1.00000	0.10330
Mathematics	0.1022	0.0713	0.0103	0.0012	0.3782		0.0004
English	0.05928	-0.02260	-0.03086	0.08392	0.04729	0.10330	1.00000
English	0.0438	0.4424	0.2942	0.0043	0.1079	0.0004	

The FREQ Procedure

STK 110 bi

STK_110_bi	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	137	11.84	137	11.84
1	1020	88.16	1157	100.00

Language

0	487	42.09	487	42.09
1	670	57.91	1157	100.00

Gender

0	535	46.24	535	46.24
1	622	53.76	1157	100.00

Descriptive statistics

The LOGISTIC Procedure

Model Information

Data Set	WORK.STATS	
Response Variable	STK_110_bi	STK 110 bi
Number of Response Levels	2	
Model	binary logit	
Optimization Technique	Fisher's scoring	

Number of Observations Read	1157
Number of Observations Used	1157

Response Profile

Ordered Value	STK_110_bi	Total Frequency
1	0	137
2	1	1020

Probability modeled is STK_110_bi=1.

Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

	Intercept	Intercept and
AIC	843.704	825.642
SC	848.758	855.963
-2 Log L	841.704	813.642

R-Square 0.0240 Max-rescaled R-Square 0.0464

Descriptive statistics
The LOGISTIC Procedure

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	28.0629	5	<.0001
Score	31.8143	5	<.0001
Wald	27.2196	5	<.0001

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.3372	1.1746	0.0824	0.7741
Ratio	1	-0.7592	0.1969	14.8727	0.0001
Mathematics	1	0.0112	0.0102	1.2085	0.2716
English	1	0.0202	0.0122	2.7370	0.0980
Language	1	0.2989	0.1879	2.5303	0.1117
Gender	1	0.2881	0.1865	2.3871	0.1223

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
Ratio	0.468	0.318	0.688
Mathematics	1.011	0.991	1.032
English	1.020	0.996	1.045
Language	1.348	0.933	1.949
Gender	1.334	0.926	1.923

Association of Predicted Probabilities and Observed Responses

Percent Concordant	61.9	Somers' D	0.250
Percent Discordant	36.9	Gamma	0.253
Percent Tied	1.3	Tau-a	0.052
Pairs	139740	c	0.625

The LOGISTIC Procedure

Partition for the Hosmer and Lemeshow Test

		STK_110_bi = 1		STK_110_bi = 0	
Group	Total	Observed	Expected	Observed	Expected
1	116	89	88.93	27	27.07
2	116	101	97.65	15	18.35
3	116	94	99.97	22	16.03
4	116	102	101.46	14	14.54
5	116	102	102.86	14	13.14
6	116	108	104.03	8	11.97
7	116	103	105.16	13	10.84
8	116	109	106.22	7	9.78
9	116	105	107.46	11	8.54
10	113	107	106.26	6	6.74

Hosmer and Lemeshow Goodness-of-Fit Test

Chi-Square	DF	Pr > ChiSq
7.0470	8	0.5316

Classification Table

	Correct		Incorrect		Percentages			
Prob		Non-		Non-	Sensi-	Speci-	False	False

Level	Event	Event	Event	Event	Correct	tivity	ficity	POS	NEG
0.140	1020	0	137	0	88.2	100.0	0.0	11.8	.
0.160	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.180	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.200	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.220	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.240	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.260	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.280	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.300	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.320	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.340	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.360	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.380	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.400	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.420	1020	1	136	0	88.2	100.0	0.7	11.8	0.0
0.440	1019	1	136	1	88.2	99.9	0.7	11.8	50.0
0.460	1019	1	136	1	88.2	99.9	0.7	11.8	50.0
0.480	1019	1	136	1	88.2	99.9	0.7	11.8	50.0
0.500	1018	1	136	2	88.1	99.8	0.7	11.8	66.7
0.520	1018	1	136	2	88.1	99.8	0.7	11.8	66.7
0.540	1018	1	136	2	88.1	99.8	0.7	11.8	66.7
0.560	1018	2	135	2	88.2	99.8	1.5	11.7	50.0
0.580	1018	2	135	2	88.2	99.8	1.5	11.7	50.0
0.600	1017	2	135	3	88.1	99.7	1.5	11.7	60.0
0.620	1015	2	135	5	87.9	99.5	1.5	11.7	71.4
0.640	1015	2	135	5	87.9	99.5	1.5	11.7	71.4
0.660	1012	2	135	8	87.6	99.2	1.5	11.8	80.0
0.680	1012	2	135	8	87.6	99.2	1.5	11.8	80.0
0.700	1012	4	133	8	87.8	99.2	2.9	11.6	66.7
0.720	1008	6	131	12	87.6	98.8	4.4	11.5	66.7
0.740	1004	8	129	16	87.5	98.4	5.8	11.4	66.7
0.760	1000	9	128	20	87.2	98.0	6.6	11.3	69.0
0.780	990	15	122	30	86.9	97.1	10.9	11.0	66.7
0.800	973	16	121	47	85.5	95.4	11.7	11.1	74.6
0.820	943	22	115	77	83.4	92.5	16.1	10.9	77.8
0.840	892	31	106	128	79.8	87.5	22.6	10.6	80.5
0.860	794	44	93	226	72.4	77.8	32.1	10.5	83.7
0.880	637	72	65	383	61.3	62.5	52.6	9.3	84.2
0.900	442	94	43	578	46.3	43.3	68.6	8.9	86.0
0.920	207	117	20	813	28.0	20.3	85.4	8.8	87.4
0.940	46	132	5	974	15.4	4.5	96.4	9.8	88.1
0.960	1	137	0	1019	11.9	0.1	100.0	0.0	88.1
0.980	0	137	0	1020	11.8	0.0	100.0	.	88.2

Surviving the drop-out

Bianca Krüger 10144065

STK795 Research Report

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

Supervisor: Dr. L Fletcher

Department of Statistics, University of Pretoria

2 November 2016

Abstract

Survival analysis deals with analyzing the duration of time until one or more events occur. In this report, survival analysis will be applied to lifetime-type educational data in order to obtain a distribution of the duration of undergraduate studies of a complete sample of students for the period of 2010 to 2015 in the Department of Statistics. This distribution will be used to examine how many students finish their degree in the prescribed amount of time, how many students drop out, and how many students study longer than the prescribed period. We will investigate what students who drop out have in common with "perpetual students", as Kalamatianou & McClean referred to the censored observations [6]. Difference in features between perpetual students and students who finish their degree within the prescribed three years will also be investigated. It will be taken into account whether a student is male or female.

To evaluate and understand the problem of students who do not complete their degree in the prescribed period, we will make use of non-parametric survival analysis techniques, more specifically, the Kaplan-Meier product limit method.

We will construct one survival curve to examine the distribution of study time-duration for all students. We will search for common traits between perpetual students in order to pinpoint the problem of students not finishing their degree. For the same reason we will also draw comparison between the traits of students who finished their degree in the minimum amount of time and the traits of perpetual students. The logrank test to compare survival curves is the most appropriate method since some of the observations will still be censored at the end of the study.

Declaration

I, *Bianca Krüger,* 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.

_____
Bianca Krüger

_____
Dr.Lizelle Fletcher

_____
25 July 2016

Acknowledgements

The author would like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a postgraduate bursary

Contents

1 Introduction **6**

2 Background theory **7**
2.1 Terminology 7
2.2 Methodology 8

3 Application **9**
3.1 Description of study 9
3.2 Practical application 9
3.3 Results 10

4 Conclusion **14**

List of Figures

1 Graphical representation of the distribution of the duration of undergraduate studies with Statistics as a subject up to third year level 10
2 Percentage of students registered and degrees awarded per year 11
3 Graduates by gender 11
4 Graduates by school 12
5 Survival curve of all students 12
6 Survival curves of students by gender 13
7 Survival curve of students by school 14

List of Tables

1 Students registered vs. students passed 10

1 Introduction

It is important for the University of Pretoria and the Department of Statistics to estimate the likelihood and the time scale of graduation of students. It is also important for the university to better understand why students complete their degrees in the set amount of time and why other students only complete their degrees one or two years later than they should have, or not at all. There are a few main reasons why it is vitally important for the university to be able to examine the distribution of the duration of studies for students in the Department of Statistics. Firstly, the university can measure the performance of the Statistics Department by looking at the degree completion rates in the minimum time. If the Statistics Department is not efficient enough, these rates would typically be very low. Funding is the second reason for the university's interest in this topic. The longer a student takes to get their degree, the longer their studies have to be funded, albeit with a loan, a bursary or parents funding the student's studies. [6] A topic that overlaps in a certain sense with reason number two and which brings us to the third reason, is the fact that, as the duration of a student's study career increases, the time period before they can become an asset to the economy also increases. The fourth reason for concern is that these students take up the space of other students, since there are limited resources available. In effect, these "perpetual students" as Kalamatianou & McClean [6] labelled them, are putting a damper on the growth of the skilled labour force in the field of statistics, because new students, who possibly could have finished their degree in the prescribed time-limit are now being prevented to enter the system.

The aim of this research report is to identify the distribution of the duration of undergraduate studies for statistics students at the University of Pretoria in order to examine how many students finish their degree in the prescribed minimum time period of three years, how many students study longer than the prescribed minimum time period, and how many students drop out before finishing their degree. Furthermore, identifying similarities in students who study longer than the prescribed three years, making use of survival analysis techniques, may help the university to refine their admission requirements. Various survival curves will be constructed in order to look at the problem of students studying longer than the prescribed time period and students dropping out, from different angles. In order to construct these survival curves, the focus will be placed on a set of characteristics in individual students. A set of survival curves, separating students on gender will be constructed. The purpose of constructing more than one survival curve is to compare the traits of individual "perpetual students" in order to find similarities and diagnose the problem of students not finishing their degree in the prescribed minimum time period of three years, or dropping out altogether.

According to Allison, survival analysis is a family of statistical methods used to study the occurrence and timing of events. These methods are most often applied to the medical field to study the timing of patient deaths or the recurrence of disease in certain experiments. [1] Noda and colleagues (2002) in Dawson and Trapp did an experiment on small-cell cancer patients' reaction- and survival times on different combinations of cancer fighting medications. A death in this case would be the death of a patient. Borghi and colleagues (2002) in Dawson and Trapp compared different diets with recurrent formation of calcium oxolate stones. A death in this case would be the recurrence of a calcium oxolate stone. [5] Although survival analysis is historically most popular in the medical field, it is also applied and very useful in engineering (equipment failures), physics and science (earthquakes), econometrics (stock market crashes) and social sciences (revolutions, births, marriages and arrests) to name but a few [1],[6]. The name *survival analysis* and the terminology used in the field such as a *death*, have the unfortunate effect of narrowing the view of potential applications of these statistical methods. Survival analysis may take on different names in different fields of study. In sociology it is called *event history analysis*, while *reliability analysis* and *failure time analysis* is used in engineering, and *duration analysis* is used in econometrics. The difference in names does not imply a difference in methods, although approaches may differ from one discipline to another.

According to Allison (2010), survival analysis was designed for long-term data on the occurrence of certain events. He defines an event as a qualitative change (transformation from one discrete state to an additional discrete state) that can be situated in time. An arrest for example, is a transition from being a free person to being arrested. To apply survival analysis, it is important to know when the change occurred. The event

should be plotable in time. [1] In 2003 Kalamatianou & McClean conducted a study on 10 313 students at a Greek university. The goal was to examine the distribution of the duration from the start of a student's study career up until the moment that the student graduated, if the student graduated. Students that continued their studies indefinitely were referred to as perpetual students (right censored students) and students that dropped out or moved to another university were excluded from the study. Graduation was considered a death or in other words, the event of interest. In this case, a death was a positive outcome, which is often contrary to the usual survival analysis terminology, where a death is seen as a failure. The data were right censored, meaning that at the time of conclusion of study, there were still survivors (students that have still not finished studying). [6] These students were referred to as right censored students. Students with censored data presented only partial information about event occurrence. If a student's event time was censored, the researchers would only know, that if the person experienced the event, it would have been after the collection of the data has ended. Kalamatianou & McClean used parametric as well as nonparametric survival models to estimate and examine the distribution of duration of studies at this Greek university [6]. Parametric survival models are most commonly used in the engineering workspace, while its alternative, nonparametric survival models are more popular in the field of medicine [10]. Although Kaplan & Meier (1958) in their seminal article developed their nonparametric survival analysis models in the medical field [7], these models have been used by Kalamatianou & McClean in the field of lifetime-type educational data. In this article, we will also make use of nonparametric, rather than parametric survival models. When looking at graduation rates of women and men separately, it was found by Kalamatianou & McClean that there were significant differences between the graduation rates of the two gender groups. [6] To compare survival times for two or more groups, Dawson & Trapp explained two different methods. These methods are the actuarial method and the Kaplan-Meier product limit method. The actuarial method groups the data into small time intervals in order to keep the number of censored cases in each interval small. This method gives credit to participants who withdrew during the study. The Kaplan-Meier product limit method is homogenous to actuarial analysis, except that the time since entry is not partitioned into intervals. Depending on the number of events that occurred, Kaplan-Meier product limit method, also called Kaplan-Meier curves, involves fewer calculations than the actuarial method, mainly since survival is only estimated when an event occurs, so withdrawals are ignored in a certain sense. [5],[7]

Similar to Kalamatianou & McClean's study, Plank and colleagues conducted a study on high school dropout rates in America using data from the National Longitudinal Survey of Youth of 1997. The study was done to answer a few main questions of concern. Firstly, they wanted to estimate the amount of students dropping out of high school. Secondly they tried to identify reasons for students dropping out and thirdly they addressed solutions to this long run economical problem. [9]

2 Background theory

This chapter will be separated into two sub-sections: Terminology and Methodology.

2.1 Terminology

Some terms in survival analysis need some brief explanation, since survival analysis uses a very specific set of terms. Survival analysis techniques are described using common words which have a unique meaning in survival analysis context. These terms will be explained in this section.

Death:
An event of interest occurring. In the case of this study, a death will be the event of a student finishing their degree.

Survival:
The absence of the event of interest. In the case of this study, survival will be the event of a student continuing to study after the minimum time period to obtain a degree.

Censored:
There is only partial information available regarding the survival time of an individual. In the case of this study, all students who have not completed their degree at the conclusion of the study, will be categorised as censored cases, since we do not know whether they will obtain their degree or not.

Right censored:
With right censored data, the unobserved data lies to the right of the conclusion time of the study.

Left censored:
With left censored data, it is not possible to know when a subject entered the study, e.g. if a student has already been in the academic system when the study started, we have no way of knowing whether the student has failed or whether the student changed their academic plan.

Withdrawal:
A subject, for whatever reason (except for a death, i.e. obtaining their degree), is not part of the study anymore. A subject may move to another city, or he/she willingly resigns from the study. In the case of this study, a dropout will be seen as a withdrawal.

Number of students still at risk:
The amount of subjects still at risk, can be explained as the number of subjects left to still experience a death. Two factors can have an effect on the number of students still at risk. A death can decrease the amount at risk, while a withdrawal also decreases the amount at risk, since withdrawn subjects get excluded from the study. In the case of this study, the amount still at risk is the number of students who are yet to receive their degree. Thus, at the beginning of the study, the number of students at risk equals the total number of students entered into the study.

Right truncated:
Only observations who experience the event of interest by a specific time will be included in the sample.

Left truncated:
Only observations who survive past a certain time will be included in the sample.

2.2 Methodology

We will use the Kaplan-Meier method to estimate the survival function $\mathrm{S}(t)$, for t taking on the values 3, 4 and 5 (total amount of years before degree completion). This method takes into account the time that an event (death or a withdrawal) occurred t_i, the amount of events that have occurred at a certain time d_i and the amount of subjects that are still at risk just prior to the event occurring r_i. Each time an event occurs, we will calculate $\hat{S}_{t_i}$, which is the estimated survival function at time t. We will estimate the survival function $\mathrm{S}(t)$ with the following: [2]

$$\hat{S}_t = \prod_{t_i \leq t} \left[\frac{r_i - d_i}{r_i} \right] \tag{1}$$

where:
t_i= Time at which a student graduated
t = Time at which this study is concluded
d_i= Number of students that obtained a degree at time t_i
r_i= Number of registered students who are yet to receive a degree just prior to time t_i

To compare two survival functions (groups of students with different traits) with one another, we will make use of the logrank test. The logrank test will compare the number of observed graduations in each group

with the number of expected graduations based on the number of graduations in the combined groups. Using the null hypothesis that survival curves are equal in the two groups, we can use the following expression to test it: [5]

$$\chi^2 = \frac{(O_1 + E_1)^2}{E_1} + \frac{(O_2 + E_2)^2}{E_2} \tag{2}$$

where:
O_1= Total number of observed graduations in group 1
E_1= Total number of expected graduations in group 1
O_2= Total number of observed graduations in group 2
E_2= Total number of expected graduations in group 2

3 Application

The data analysis for this essay was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA.

3.1 Description of study

In this report, we will make use of nonparametric survival analysis methods, more specifically, the Kaplan-Meier method, in order to estimate the distribution of the duration of undergraduate studies in the Department of Statistics. Similar to the 2003 study of Kalamatianou & McClean, we will conduct a study on 20 000 students from the University of Pretoria. We will also consider graduation as a death, i.e. as the occurrence of an event. Since Kalamatianou & McClean found significant differences between the graduation times of men and women, this phenomenon will also be studied. [6] We will further compare groups of students who completed their degree on time with groups of students who studied longer than the prescribed minimum time period in order to identify significant differences. Since it goes beyond the scope of this research report, solutions to the problem of students dropping out or students studying longer than the prescribed minimum time period will not be addressed. The SAS procedure PROC LIFETEST will be used for the practical application; the data were obtained from the University of Pretoria's Bureau for Institutional Research and Planning (BIRAP) database.

3.2 Practical application

From the BIRAP dataset, we have only included students who were supposed to take Statistics up until their third year according to their academic plan. These academic plans are BCom: Economics and BCom: Statistics. Our study included 406 students. 89 of these students either dropped out, or changed their degree to something other than BCom:(Statistics) or BCom:(Economics) before they could complete their third year of study. This means that at the beginning of the third year of study for the students in our set, only 317 students remained.

The SAS procedure, PROC LIFETEST was used to conduct survival analysis on the data. Comparisons of survival between student genders and the school systems the student matriculated from were drawn using a logrank test statistic in the PROC LIFETEST procedure.

3.3 Results

Year	Degrees awarded	Students registered at beginning of year	Students left	Percentage of students passed in that year	Percentage passed of 317 students who made it to third year	Number of dropouts at end of year	Percentage dropped out
1	0	406	406	0,00%	0,00%	31	7,6%
2	0	375	375	0,00%	0,00%	58	15,5%
3	104	317	213	32,80%	32,80%	79	24,9%
4	58	134	76	43,28%	51,50%	40	29,9%
5	10	36	26	27,70%	54,25%	20	55,6%
6	4	6	2	66,60%	55,50%	2	33,3%

Table 1: Students registered vs. students passed

Table 1 summarises the amount of registered students at the beginning of each year, the amount of degrees awarded at the end of each year, the amount of students left at the end of each year, the percentage of students that passed and the percentage of students that did not return to complete their studies the following year. Only 55.5% of all students who made it up to their third year of studies completed their degree. Of the 406 students who started out in the first year, a mere 43.3% went on to obtain a degree. This number implies that less than half of the first year students hopeful to obtain a degree, actually see it through. It is interesting, but expected that the percentage of students who do not return to finish their studies, increase each year. This can be explained by the fact that students either give up, and feel that they will not be able to complete their degree, or the funds to continue studying are depleted, and these students are forced to give up.

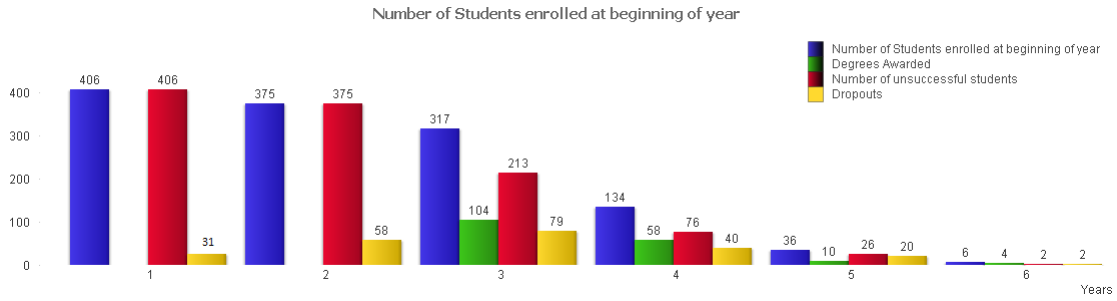


Figure 1: Graphical representation of the distribution of the duration of undergraduate studies with Statistics as a subject up to third year level

In Figure 1, the distribution of the duration of studies for undergraduate students at the University of Pretoria per year can be observed. The number of students unsuccessful at the end of year one and year two is equal to the number of students who registered, since the minimum time period to obtain an undergraduate degree in Statistics or Economics at the University of Pretoria is three years. The blue bar represents the number of students registered with the university at the beginning of each year; the green bar represents the amount of degrees awarded at the end of that year; the red bar represents the amount of censored (unsuccessful) students at the end of the year; the yellow bar represents the amount of students who dropped out at the end of the year, i.e. did not be return the next year. Thus, right censored students - number of students dropped out = number of registered students for the next year.

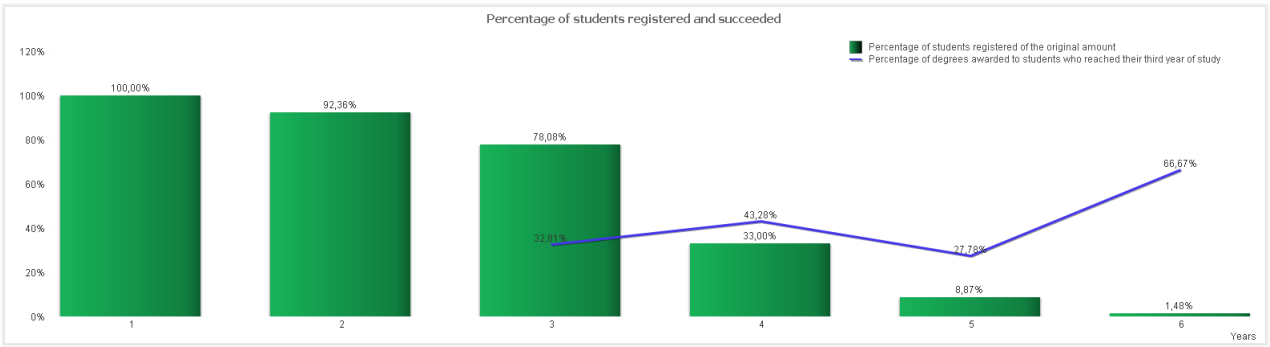


Figure 2: Percentage of students registered and degrees awarded per year

Figure 2 shows the percentage of students registered, dropped out and the percentage of degrees awarded. After 3 years of studies, only 25.62% of students who registered for a degree have been awarded with a degree, while 21.9% have dropped out. Of those who reached their third year, only 32.8% graduated within the prescribed minimum time period of three years. This is extremely costly to the university out of a financial and spacial perspective. Not only are the 67.2% of students who study longer than the prescribed 3 years costing the university money in subsidies but these students are also taking up the space of prospective students who are eager to join the university. These low figures also reflect badly on the university and warrants some initiative from the university to create an overall better experience for a student regarding his/her relationship with the university. Analysis should be done on the students who do not obtain their degree in the minimum time frame, to try and ascertain how they can be assisted to obtain their degree. Said analysis is beyond the scope of this article.

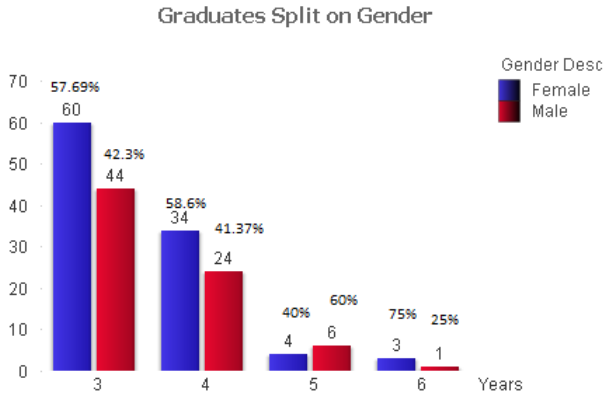


Figure 3: Graduates by gender

From Figure 3 it can be seen that females generally have a higher graduation rate than males. 60 out of the 104 degrees that were awarded after 3 years of studies were awarded to females. This equates to about 58% of the students who obtained a degree after 3 years. There can be various reasons for this occurrence, including field of study and responsibility levels at the age of 20 between males and females. With field of study, it is meant that if this analysis is done on students studying computer science, which is a mainly male dominated field of study, the graduation numbers would likely take on the opposite pattern of what we observe in our data. With responsibility levels, it is meant that male and female students between the ages of 18 and 20 do not have the same sense of responsibility. Females tend to generally be more tame, with a stronger sense of responsibility and duty.[3, 8]

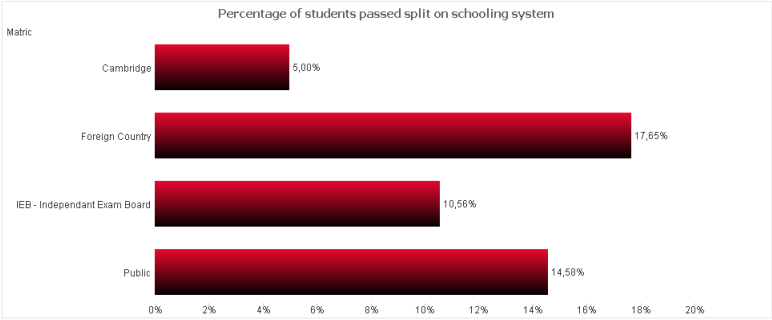


Figure 4: Graduates by school

From Figure 4 it can be seen that even though the largest number of students who graduate, are from public schools, the highest percentage of students who graduate obtained their matric certificate in a foreign country. A possible reason for this could be that students from other countries experience more pressure to graduate since they are here on a study permit.

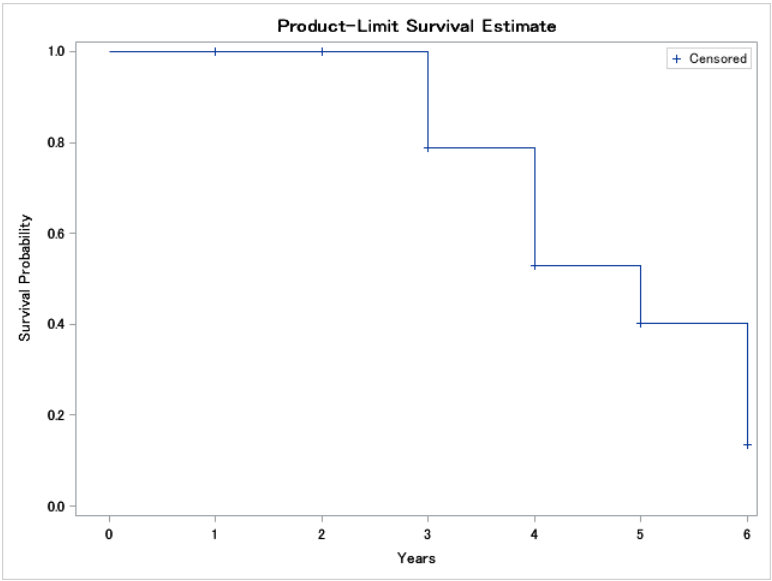


Figure 5: Survival curve of all students

Figure 5 displays a survival curve for the students who graduated in year 3, 4, 5 or 6 of their studies. From this curve, based on the data acquired from the BIRAP dataset, the estimated probability of a student graduating in their third year of study, is a mere +- 20%. As can be expected, the estimated probability of a student to graduate increases as the number of years studied increase. The probability of a student to be awarded with their degree after 4 years of study is +- 50% which is a 30% jump from year 3 to year 4. The probability of a student graduating after 5 years of study is +- 65%. This is only a 15% increase in probability from year 4 to year 5. After 5 years, students who have not finished their studies are requested to terminate their studies with the university. This means that these students are considered as right censored. We do not know if these students enrolled for a degree at another university, or if they dropped out completely. The reason for our study still containing students in a 6th year of study is that these students changed their degree at some point or took a gap year, and thus were given an extra year to finish their degree.

In the following two figures, Figure 6 and Figure 7, we will look at survival curves of students by gender and by schooling system, making use of the logrank test statistic to draw comparison between different groups:

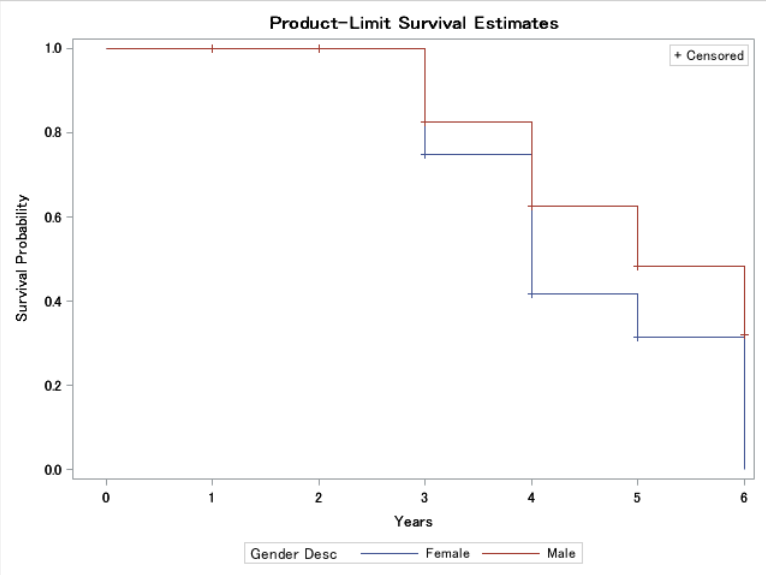


Figure 6: Survival curves of students by gender

The survival curve in Figure 6 shows that the probability of a female student obtaining her degree in years 3, 4 and 5 is higher than the probability of their male counterparts obtaining their degree. This could be ascribed to the fact that female students are more "dedicated" in a certain sense (a female student would rather stay at home to study, rather than attend a sports event on the eve of a test).

From the logrank test statistic, $\chi^2 = 11.6112$ with p= 0.0007, it can be seen that the graduation behaviour of males and females differ even on a 1% level of significance, which makes gender a very influential variable in the analysis of whether and when a student will in fact obtain their degree. Similar results were found in several other studies [3, 8].

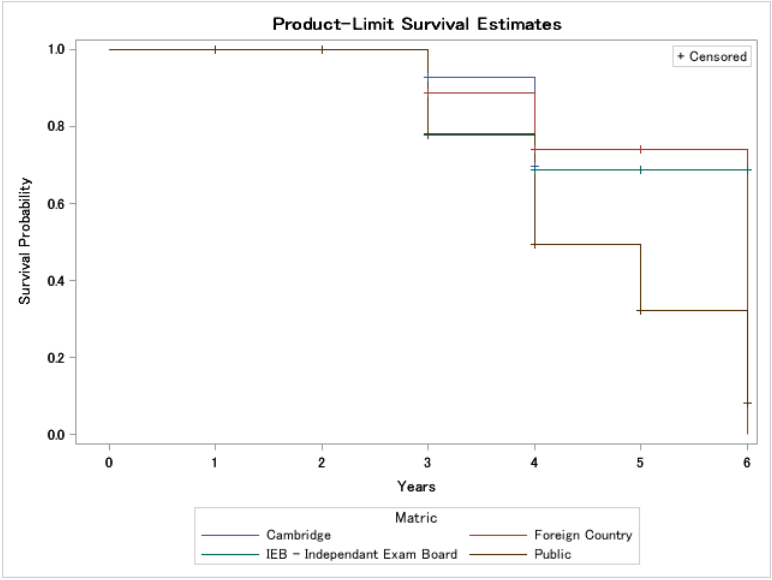


Figure 7: Survival curve of students by school

Figure 7 shows the different probabilities of students who matriculated from different schooling systems obtaining their degrees. The different schooling systems we looked at are public schools, IEB schools, schools in foreign countries and schools that follow Cambridge syllabus. It has to be noted that most of the students in our study received their secondary education from a public school in South Africa. There is a significant difference between the probability of a student who attended public school and the probability of a student who attended a school with a Cambridge schooling system to obtain their degrees, with a student who attended a public school having a higher probability to obtain a degree. Since there are large difference in the number of students attending public schools versus the other schooling systems, we can refer back to Figure 4, making use of percentages, for more context. Studies have already been done in 2015 on the significance of a student's grades in secondary school [4].

The logrank test statistic is $\chi^2 = 7.2806$ with a p-value of 0.0635. The different schooling systems are not significantly different on a 5% level of significance, but are significantly different on a 10% level of significance.

4 Conclusion

From our findings, it can be concluded that for the period of 2010 to 2015, only a quarter of the students who registered for a three year degree in BCom:(Statistics) or BCom:(Economics) actually obtained said degree within the minimum prescribed period. This information provides the university with the necessary information that the intake criteria for students applying to study towards a Statistics or Economics degree may be due for revision. In the monitored time period of 2010 to 2015, a significant amount of 230 students did not obtain their degree due to dropping out before finishing. This is a staggering number of 56.65% of students. Furthermore, 72 of the 176 students who graduated (40.9%), took longer than the prescribed minimum time period of three years. We analysed the graduation rates between males and females and observed that more females than males tend to finish their degree. Only 75 (42.6%) out of the 176 students who graduated were males. We have constructed three separate survival curves, of which two were used for logrank tests. From these survival curves, it can be seen that the model is a good fit for the data. The survival curves separating graduation times on gender confirms what we found in the data. Females are more likely to graduate than males. The survival curves separating graduation times on schooling systems also shows a moderately significant albeit smaller difference between public schooling graduates and graduates from other schooling systems.

To conclude, gender definitely plays a role in obtaining an undergraduate degree in statistics, while the student's schooling system also has a part to play. Going forward, it would be possible, with more research from the university, to start identifying the candidates at risk of struggling to obtain a degree even before the student starts their career at the University of Pretoria. This way, the university can start intervening to make sure that students from previously disadvantaged backgrounds get the same learning experience as students with an advantage and the university can ensure that students with the highest aptitude for statistics will get a chance before a student who might just drop out in second year. Much research, which is beyond the scope of this essay still needs to be done on this topic to fully understand and grasp the intricate workings of why one student would drop out while another would hold on and finish their degree.

References

[1] Paul D Allison. *Survival analysis using SAS: a practical guide*. SAS Institute, 2010.

[2] Alan B Cantor. *SAS Survival analysis techniques for medical research*. SAS Institute, 2003.

[3] Justin R Chimka, Teri Reed-Rhoads, and Kash Barker. Proportional hazards models of graduation. *College student retention*, 2007.

[4] Renata Clerici, Anna Giraldo, and Silvia Meggiolaro. The determinants of academic outcomes in a competing risks approach: evidence from italy. *Studies in Higher Education*, 2015.

[5] B Dawson and R.G. Trapp. *Basic & Clinical Biostatistics*. Lange, 2004.

[6] 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.

[7] Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481, 1958.

[8] Gillian M Nicholls. Analyszing time to student course withdrawal patterns for predictive modeling. *ASEE Southeast section conference*, 2013.

[9] Stephen B Plank, Stefanie DeLuca, and Angela Estacion. High school dropout and the role of career and technical education: A survival analysis of surviving high school. *Sociology of Education*, 81(4):345–370, 2008.

[10] G. Rupert and JR. Miller. *Survival Analysis*. Wiley Publications, 1998.

Appendix

SAS code to generate the survival curves:

Generating a survival curve using the whole set of data.

```
data SurvivalCurve_Student_Data;
set survival;
d=0;
if years>=3 and degree_status=1 then do;
d=1;
end;
run;

proc lifetest plots=(s);
time years*d(0);
run;
```

Generating survival curves to draw comparison between gender and schools by making use of the Logrank test statistic:

Drawing a comparison between gender using survival curves:

```
data Gender;
set survival;
d=0;
if years>=3 and degree_status=1 then do;
d=1;
end;
run;

proc lifetest plots=(s);
time years*d(0);
strata gender_desc;
run;
```

Drawing comparison between schools using survival curves.

```
data schools;
set ss;
d=0;
if years>=3 and degree_status=1 then do;
d=1;
end;
run;

proc lifetest plots=(s);
time years*d(0);
strata Matric;
run;
```

Econometric modelling: Model specification and diagnostic testing

Tokelo Iren Letshedi 13252276

STK795 Research Report

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

Supervisor: Dr J Kleyn

Department of Statistics, University of Pretoria

02 November 2016 (final)

Abstract

In this essay, we will be looking at one of the assumptions stated by the Classical Linear Regression Model (CLRM), namely, that the model must be correctly specified. The focus of the essay will be to distinguish between the two different types of model specification errors namely, over-fitting and under-fitting of models, then the consequences of these errors and different testing procedures which can be used to detect the model specification errors are discussed. A cubic cost function was considered as an application and all the test procedures were illustrated based on the cubic cost function and were put into practice.

Declaration

I, *Tokelo Iren Letshedi,* 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.

_____
Student's full name

_____
Supervisor's name

_____
Date

Acknowledgments

The author would like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a post-graduate bursary.

Contents

1 Introduction and Literature Review **6**

2 Background Theory

Model Specification Errors. **8**
2.1 Types of model specification errors 8
2.2 Omission of relevant variables (Under-fitting). 8
2.2.1 Ramsey's RESET test 9
2.2.2 Durbin-Watson D statistic 9
2.2.3 Lagrange Multiplier (LM) test 10
2.2.4 Examining residuals 10
2.3 Inclusion of irrelevant variables (Over-fitting). 11
2.3.1 Detecting the presence of unnecessary variables 11

3 Application **12**
3.1 The Ramsey's RESET test. 12
3.2 Lagrange Multiplier (LM) test 13
3.3 Examining residuals 14
3.4 Durbin-Watson d statistic 16
3.5 The $t-$test 16
3.6 The $F-$test 17

4 Conclusion **17**
Appendix20

List of Figures

1 The total cost curve. 7
2 Residuals v.s X_i, $X_i^2$and X_i^3 14
3 Residuals v.s $X_i$and X_i^2 15
4 Residual v.s X_i 15
5 Estimated d statistics for the linear, quadratic and cubic cost functions. 16
6 Cubic cost function 21
7 Cubic cost function 22
8 Quadraic cost functon 23
9 Quadratic cost function 24
10 Linear cost function 24
11 Linear cost function 25
12 Cubic model and adding an irrelevant variable 26
13 Cubic model and adding an irrelevant variable 27

List of Tables

1 Total Cost Function 12

1 Introduction and Literature Review

In this essay, we will distinguish between two types of model specification errors in depth and give highlights on what model mis-specification errors is. Model specification errors includes different types of errors namely: omission of relevant variables, inclusion of unnecessary variables, the use of incorrect functional form, incorrect specification of the stochastic error term, and errors of measurement in the dependent variable Y and the explanatory variable X [7]. To detect model specification errors, different tests are to be considered namely: the Durbin-Watson d statistic, Ramsey's RESET *test*, examination of residuals, Lagrange Multiplier (LM) *test*, $t-test$ and the $F-test$. In this essay, we will mainly focus on the omission and addition of relevant/irrelevant variables in a model.

The Ramsey RESET test [13] was discovered by Ramsey J.B in 1968 while doing his Ph.D thesis paper at the Wisconsis-Madison University. Ramsey argued and have shown that residuals are normally distributed with constant but not zero mean values under least squares distributions for different specification errors [13]. The RESET test is said to be a useful indicator when something is wrong [12] and it is generally for testing mis-specification as well as heteroscedasticity. This test has no power in detecting any results when the omitted variable(s) are linearly related to the variables that are included [7, 13].

The Durbin-Watson d statistic was developed by Geoffrey Watson and James Durbin and in 1950/1951 they developed the test's bounds to check for the existence of auto-correlation [7, 15, 14]. The Durbin-Watson d test is used to calculate residuals for the existence of auto-correlation [5]. A model is assumed and the residuals are calculated under Ordinary Least Squares (OLS). If there is a mis-specification of the assumed model, then there is an exclusion of a necessary independent variable. We calculate the d value and if it is significant, then we do not reject and conclude that the model is mis-specified [8]. The Durbin-Watson d statistic is calculated then compared to its d_{LOWER} and d_{UPPER} [7, 14] to check for any auto-correlation in the residuals [5, 15]. Observed auto-correlation will reflect that some variables are included in the error term instead of the model itself [5, 15].

The Lagrange Multiplier (LM) test was developed by Dr. C.R Rao in the University of Pennsylvania state and all results were published by Rao and Poti in 1946 [4, 7, 10]. This test let you compare a true model against a restricted model [7], then if the critical chi-square obtained is greater than the calculated value then we do not reject the restricted model, that is the restricted model is the true one [8, 9]. It is found that when two models are being compared to each other, the true model among the two models will reflect with the highest R^2 value or $\bar{R}^2$ value [5, 9].

There are many examples on econometric modelling under model specification and mis-specification errors that exists [3, 8]. Model specification errors originated from the correlation between the error term and independent variable which are caused by: the omission of important variables, including unnecessary variables, the use of incorrect functional form into a model, wrong specification of the stochastic error term,etc [1, 5, 7, 8, 9, 11].

In this research essay we will discuss the model specification errors throughout using a cost function example [7] from a cubic polynomial ,stated as

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \mu_i \tag{1}$$

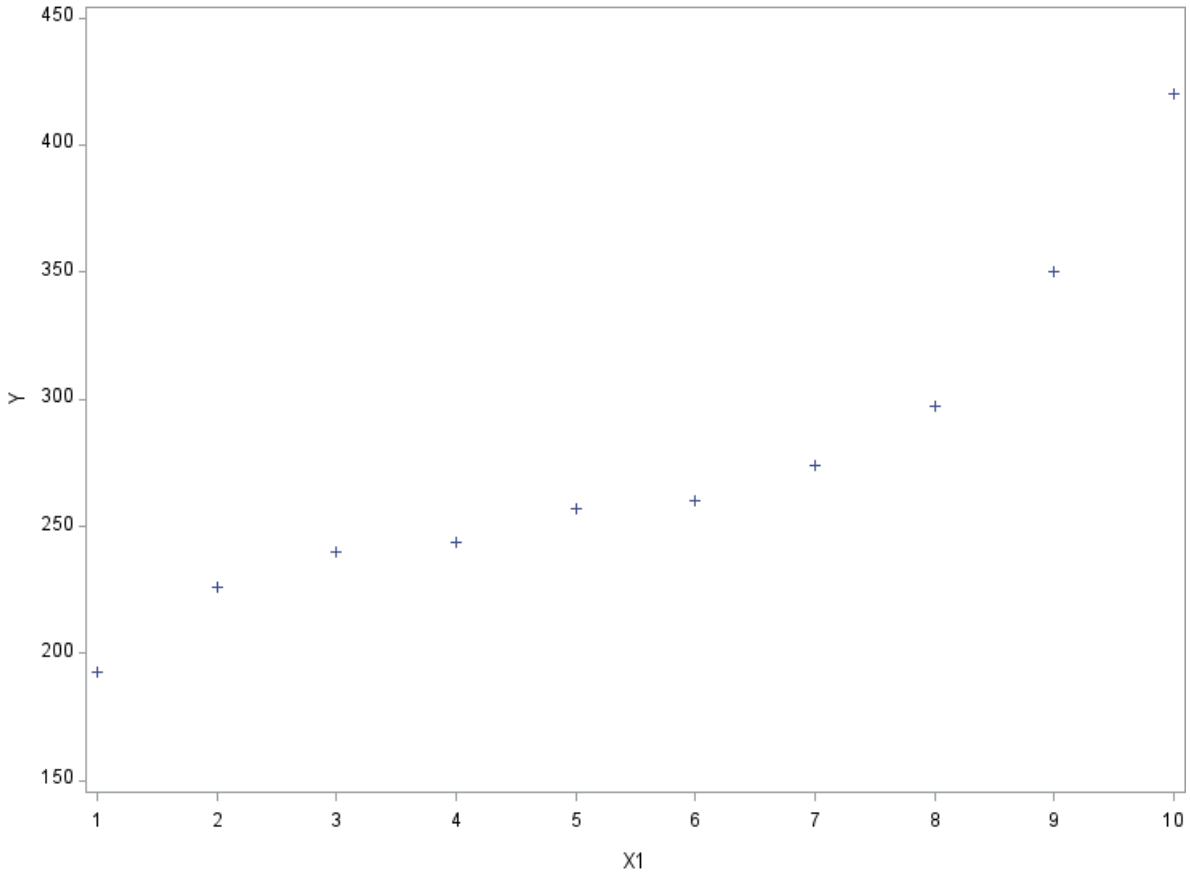


Figure 1: The total cost curve.

This cost function is a cubic polynomial since it resembles an S-shaped curve, this will be proven in the application section, with Y representing the total cost and X representing the output thereof. The law of diminishing returns dictated that the average and marginal cost curves are presented by U-shape curves. Both the AC and MC decreases as output rises but they tend to go up after a certain level of output leading to the effects of the law of diminishing marginal returns. We use the total cost curve to derive both the AC and MC cost curves and for that reason, parameters from the total cost function ($Eq.(1)$) have some restrictions on them because of the found U-shaped curves from the AC and MC curves, and those restrictions that must be satisfied by the parameters are as follows

1. β_0, β_1 and β_3 are greater than 0.
2. β_2 is less than 0.
3. β_2^2 is greater than $[3\times(\beta_1\times\ \beta_3)]$.

All these restrictions will be useful for our application section, since they will help checking if our model have some specification errors or not.

In section 2 we will consider the types of model specification errors but we will only discuss the first two types of model specification errors mentioned below in more detail. We will also discuss the test procedures and the detection of unnecessary variables under these two errors of specification. In section 3 we will apply the test procedures and the detection of unnecessary variables method in our cubic cost function to test for errors. In section 4 we conclude our findings.

2 Background Theory

Model Specification Errors.

2.1 Types of model specification errors

In this section, the different types of model specification errors will be considered. The main types of model specification errors are:

1. Omission of relevant variables (Under-fitting).
2. Inclusion of irrelevant variables (Over-fitting).
3. Using a wrong functional form.
4. Measurement errors (Dependent and independent variable).
5. Wrong specification of the stochastic error term.
6. Assuming a normal distribution for the error term.

We will only focus on the first two model specification errors above namely, the omission of relevant variables (under-fitting) and the inclusion of irrelevant variables (over-fitting). These model specification errors will be illustrated based on a practical example of the estimation of a total cost function.

2.2 Omission of relevant variables (Under-fitting).

In this section we consider the under-fitting of a model which is the omission of important variables in a model being used or observed. It is the leaving out of important variables that can assist in explaining the dependent variable. A systematic pattern will remain unexplained in the model.

Consider the following true model of a cost function, given in $Eq(1)$, as specified in [7]

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \mu_i$$

this equation is a cost function ,where X_i's represents output and Y represents the total cost of producing the output.

But then suppose that we use the following model instead of the true model

$$Y_i = \alpha_0 + \alpha_1 X_i + \alpha_2 X_i^2 + \varepsilon_i \tag{2}$$

equation (2) is our fitted model, then in $Eq.(2)$ we have omitted a relevant variable (X_i^3) which leads to a specification error.

The error term / residual of $Eq.(2)$ is

$$\varepsilon_i = \mu_{1i} + \beta_3 X_i^3$$

Such a model with missing relevant variable(s) can face consequences , where:

- Regression coefficients are biased and inconsistent, which implies that as the sample size specified increases the regression coefficient estimates , $\hat{\alpha}_0$ and $\hat{\alpha}_1$ from $Eq(2)$ still remain biased.
- The error variance σ^2 is estimated incorrectly.
- The estimated variance of $\hat{\alpha}_2$ is a biased estimator of the variance of $\hat{\beta}_2$.
- Measured confidence interval and hypotheses-testing statistics gives false results and those results are also misleading when conclusions are being made given by incorrectly estimated parameters from the model.

- Forecasting based on the incorrect model gives false results.
- The correlation coefficient (r_{23}) between the omitted variable (X_3) and the included variable(s) (X_2) is not zero.

There are various ways that can be used to test the omission of certain relevant variables in a model, consider the most important used tests below:

2.2.1 Ramsey's RESET test

The RESET (*Regression Equation Specification Error Test*) is generally proposed to detect specification errors. Specifying a model where variables have been omitted, we can detect a systematic pattern in residuals. Let us consider the following cost function but now we assume that our cost function is linear in output for simplicity

$$Y_i = \alpha_0 + \alpha_1 X_i + \nu_i \tag{3}$$

where Y is total cost and X is total output. The pattern in residuals can be detected by plotting the residuals ($\hat{\mu}_i$) obtained in $Eq.(3)$ against the estimated Y_i thereof. When using the RESET method, you will be required to follow four certain steps to get to a conclusion, which are:

1. From $Eq.(3)$, calculate the model's R^2 (denoted as R^2_{old}) and the estimated Y_i (denoted as $\hat{Y}_i$).
2. Observe the relationship between $\hat{\mu}_i$ and $\hat{Y}_i$ by rerunning $Eq.(3)$ and a new model will be specified (possibly correct), then we run

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 \hat{Y}_i^2 + \beta_3 \hat{Y}_i^{\,3} + \mu_i \tag{4}$$

therefore $\hat{Y}_i^2$ and $\hat{Y}_i^3$ (additional regressors) contains additional information that was left out in $Eq.(3)$.

3. We then obtain R^2 (denoted as R^2_{new}) from $Eq.(4)$. We then use the F test stated below to check whether the increase in R^2_{new} is statistically significant.

$$F = \frac{(R^2_{new} - R^2_{old})/(number\,of\,new\,regressors)}{(1 - R^2_{new})/(n - number\,of\,parameters\,in\,the\,new\,model)} \tag{5}$$

4. If found that our F value is statistically significant, for example at a 5% level of significance, then we can conclude that our model is mis-specified from $Eq.(3)$.

Using the RESET method has advantages and disadvantages. It is very easy to apply the F - test and it does not require us to state the alternative model but then if found that the initial model is mis-specified, we will still face a problem in choosing a better alternative model. [7, 13]

2.2.2 Durbin-Watson D statistic

The Durbin-Watson d statistics is given by

$$d = \frac{\sum_{i=2}^{n} (\hat{\mu}_i - \hat{\mu}_{i-1})^2}{\sum_{i=1}^{n} \hat{\mu}_i^2} \tag{6}$$

Note: *i is the index of observation and n is the total number of observations.*
Observing our cost function ($Eq.(1)$) again to use the Durbin-Watson d test to check for specification errors in a model. We compute a d-value for the cost function, and we compare it to the critical values (d_{lower} and d_{upper}) obtained at a certain percentage, then we can conclude our model using these critical values. If our d-value is below d_{lower}, it shows a positive auto correlation in the residuals , a value above d_{upper} indicate a negative auto correlation in the residuals and a value between d_{lower} and d_{upper} indicate indecision. The Durbin-Watson test always has results between 0 and 4. The correlation being observed reflects that some of the variable(s) included in the error term belongs in the model as part of the explanatory variable(s), if X_i^3 is excluded from the cost function, as $Eq.(2)$ shows, with the error term being mis-specified ($\mu_{1i} + \beta_3 X_i^3$) , and if X_i^3 affects Y_i significantly , then a systematic pattern will be present in the errors. Let us now consider few steps that can be used to detect specification errors in a model through the Durbin-Watson d test method:

1. From $Eq.(2)$, calculate the Ordinary Least Squares (OLS) error terms ($\hat{\mu}_i$).
2. Put all residuals / error terms ($\hat{\mu}_i$) obtained in step 1 in order according to the increasing values of the excluded variable, that is X_i^3.
3. Calculate the d statistic value, using $Eq.(6)$, from the ordered error terms
4. If the computed d statistic is statistically significant checked from the Durbin-Watson tables, then we can conclude that our hypothesis of the model is mis-specified. [1, 7]

2.2.3 Lagrange Multiplier (LM) test

The LM test is similar to the Ramsey's RESET test. Again, we are using our cost function example to illustrate the Lagrange Multiplier (LM) test. Comparing the cubic cost function ($Eq.(1)$) to the linear cost function ($Eq.(3)$), we see that $Eq.(3)$ is a restricted regression model with zero coefficients for the squared and cubed output variables. To put the LM test in use, one can follow its procedures:

1. Use OLS to estimate $Eq.(3)$ (the restricted regression) and get the residuals($\hat{\mu}_i$).
2. If $Eq.(1)$ (the unrestricted regression) is true, then the error terms ($\hat{\mu}_i$) under the restricted model (Eq.(3)) have close relationship to the outputs from the squared and cubed variables (X_i^2 and X_i^3).
3. We then do regression on all the residuals from the first step on the regressors from $Eq.(1)$,the unrestricted model, we then obtain:

$$\hat{\mu}_i = \alpha_0 + \alpha_1 X_i + \alpha_2 X_i^2 + \alpha_3 X_i^3 + \nu_i \tag{7}$$

   with ν_i being the error term and $\alpha_i' s$ are our parameter estimates.
4. Using a large sample, from Eq.(7) , the total number of the sample (n) multiplied by R^2 shows a chi-squared distribution with degrees of freedom (df) the same as the number of restricted variable(s):

$$nR^2 \sim \chi^2_{(number\ of\ restrictions)} \tag{8}$$

   and with the cost function case example it is two ($df = 2$) since only two variables are left out ($X_i^2$and X_i^3).
5. We will reject the restricted regression, if the chi-square value (in $Eq.(8)$) is greater than that value of the critical chi-squared at a given level of significance or otherwise, the restricted regression will not be rejected. [7, 10]

2.2.4 Examining residuals

The examination of residuals in a cross-sectional data set helps to detect model specification errors, by in fact checking errors that exists and plotting the errors will give a clear pattern and one can simply conclude from that pattern obtained. Reconsidering the cost function example, we assume that the cubic cost function is the true model

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \mu_{1i} \tag{9}$$

but one researcher decides to fit a quadratic cost function instead, that is

$$Y_i = \alpha_0 + \alpha_1 X_i + \alpha_3 X_i^2 + \mu_{2i} \tag{10}$$

and another researcher chooses to fit a linear cost function of

$$Y_i = \lambda_0 + \lambda_1 X_i + \mu_{3i} \tag{11}$$

Even though we know that both the quadratic and the linear cost functions have specification errors, we go ahead and estimate their residuals and then plot them in order to observe their patterns. If specification errors exists, the pattern of the residuals will be detected.[7]

2.3 Inclusion of irrelevant variables (Over-fitting).

In this section we will discuss the model specification errors of including irrelevant variables which is over-fitting of a model by using too many variables in a model and including some unnecessary and unrelated variables to the response variable. This is caused by errors in theory and improper variable selection procedures.

An example of a model with extra irrelevant variables:

We first consider a true model

$$Y = \theta_0 + \theta_1 X_i + \theta_2 X_i^2 + \theta_3 X_i^3 + \varepsilon_i$$

the fitted model with unnecessary variables is

$$Y = \alpha_0 + \alpha_1 X_i + \alpha_2 X_i^2 + \alpha_3 X_i^3 + \alpha_4 X_i^4 + \pi_i$$

The *consequences* of this type of model specification error are given below:

- Variance of regression coefficients are exaggerated.
- Fitted or measured model is not good for prediction of new data- Prediction is biased.
- The error variance of the model is estimated correctly.
- The measured confidence interval and hypotheses-tests are valid and true under the over-fitted model.
- The OLS estimated parameters of the over-fitted model are consistent and unbiased.

2.3.1 Detecting the presence of unnecessary variables

Consider a true cost function being developed

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \varepsilon_i \tag{12}$$

Then we specify a model with irrelevant variables as follows

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \beta_4 X_i^4 + \mu_i \tag{13}$$

and to find out if the variable X_i^4 belongs to the true model we can simply use the $t-$test. We then specify our null hypothesis and its alternative, that is

$$H_0 : \beta_4 = 0$$

$$H_a : \beta_4 \neq 0$$

The $t-$test [7] can be used to detect the existence of only one variable X_i^4 in a model by checking if the related estimated coefficient parameter β_4 is significant by using the following formula of the $t-$test:

$$t = \frac{\hat{\beta}_4}{se\,(\hat{\beta}_4)}$$

where $\hat{\beta}_4$ is the estimated parameter of β_4 and $se\,(\hat{\beta}_4)$ is the standard error of the estimated coefficient parameter. We conclude that our hypothesis (H_0) is not rejected if the calculated $t-$value is less than the given critical $t-$value. However, if we need to check if more than one variable belongs to true cost function, for example variables X_i^2 and X_i^3 , then we need to use the $f-$test [7]. We then first specify our null hypothesis and its alternative of

$$H_0 : \beta_2 = \beta_3 = 0$$

$$H_a : at\,least\,one\,\beta \neq 0$$

Consider $Eq.(12)$ as the unrestricted cost function and must be compared to our restricted cost function of

$$Y_i = \beta_0 + \beta_1 X_i + \upsilon_i \tag{14}$$

The $F-$test is used to detect the existence of more than one variable in a model by checking its significance using the restricted and unrestricted models from their residual sum of squares (denoted as RSS) and their degrees of freedom (df) as follows:

$$F = \frac{(\,RSS_R \,-\, RSS_{UR}\,)\,/\,m}{(\,RSS_{UR}\,)\,/\,(\,n \,-\, k\,)}$$

$Note: RSS_R = \sum \hat{\mu}_R^2$, is the residual sum of squares of the restricted model.
$RSS_{UR} = \sum \hat{\mu}_{UR}^2$, is the residual sum of squares of the unrestricted model.
m =restrictions number
n =total number of observations.
k =the number of coefficient parameters in the unrestricted model.

We conclude that there are no extra or unnecessary variables in a model and we have a good model when our tests ($F-test$ and $t-test$) shows that our estimated coefficients parameters are statistically significant and proved that the model's R^2-value is high.

3 Application

In this section we will consider the following test procedures for under-fitting the model namely, the RESET test, the LM restricted, examining the residuals ,and the Durbin-Watson d statistic and over-fitting the model namely, $t-$test and $F-$test.

We now consider the cost function's applications and empirical results, the true cost function given in [7] is a third(cubic)-degree polynomial as follows:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \mu_i \tag{15}$$

with the given following data set that will be used below with 10 observations.

Total cost	Output		
	X_i	X_i^2	X_i^3
193	1	1	1
226	2	4	8
240	3	9	27
244	4	16	64
257	5	25	125
260	6	36	216
274	7	49	343
297	8	64	512
350	9	81	729
420	10	100	1000

Table 1: Total Cost Function

Looking at the empirical results illustrated under the omission of relevant variables as follows:

3.1 The Ramsey's RESET test.

As stated before, for simplicity we will use the linear cost function (denoted as the old function) as described in section 2, as follow

$$Y_i = \alpha_0 + \alpha_1 X_i + \nu_i \tag{16}$$

we get the following estimated regression model:

$$\hat{Y}_i = 166.46667 + 19.93333X_i$$

$$(19.021) \qquad (3.066)$$

with R^2_{old}= 0.8409 [16]. Then the new cost function specified (possibly correct) is

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 \hat{Y}_i^2 + \beta_3 \hat{Y}_i^{\,3} + \mu_i \tag{17}$$

estimated as

$$\hat{Y}_i = 2140.7223 + 476.6557X_i - 0.09187Y_i^2 + 0.000119Y_i^3$$

$$(6.375) \qquad (4.778) \qquad (0.9856) \qquad (0.0592)$$

with R^2_{new} =0.9983 [16], regressors = 2 and parameters = 4 (new model) . We specify the hypothesis

$$H_0 : Restricted\, model$$

$$H_a : Unrestricted\, model$$

The F-test is

$$F = \frac{(R^2_{new} - R^2_{old})/(\#\, of\, new\, regressors)}{(1 - R^2_{new})/(n - \#\, of\, parameters\, in\, the\, new\, model)}$$

$$= \frac{(\,0.9983\, -\, 0.8409\,)/2}{(\,1\, -\, 0.9983\,)/(\,10\, -\, 4\,)}$$

$$= 277.76$$

and given the critical value of $F_{(\,0.05,\,2,\,6\,)} = 5.14$ from the F-distribution tables. Since our F-value is very large and larger than the given critical F-test [$F = 277.76 > F_{(0.05,2,6)} = 5.14$], the null hypothesis is rejected at a 5% level of significance and concluding that the unrestricted model Eq.(15) is the correct model.

3.2 Lagrange Multiplier (LM) test

Consider the restricted regression Eq.(16) :

$$\hat{Y}_i = 166.46667 + 19.93333X_i$$

$$(19.021) \qquad (3.066)$$

where Y is our total cost and X is output. Then regressing the residuals of the restricted regression we get the following results

$$\hat{\mu}_i = \alpha_0 + \alpha_1 X_i + \alpha_2 X_i^2 + \alpha_3 X_i^3 + \nu_i \tag{18}$$

$$\hat{\mu}_i = -24.7 + 43.5443X_i - 12.9615X_i^2 + 0.9396X_i^3$$

$$(6.375) \quad (4.779) \qquad (0.0986) \qquad (0.059)$$

with R^2= 0.9896 [16], then we calculate our

$$nR^2 = 10(0.9896) = 9.896$$

The R^2 suggests that 98.96% of the residuals of the restricted model are explained by the omitted variables which shows that there is still a systematic pattern in the residuals which was left unexplained. The following value is the critical chi-square

$$\chi^2_{(0.05)}(2) = 5.99147$$

The null hypothesis is rejected at a 5% level of significance and we conclude that the unrestricted model is the correct model, since the critical chi-square value is smaller than that of the estimated nR^2 [$nR^2 > \chi^2_{(0.05)} = 5.9914$].

3.3 Examining residuals

Examining the residuals gives a visual diagnostic to test for specification errors. If we detect a certain pattern of the residuals then we have specification errors.

Looking at our unrestricted model or true cost function which is the cubic polynomial of

$$\hat{Y}_i = 141.7667 + 63.4776X_i - 12.9615X_i^2 + 0.9396X_i^3$$

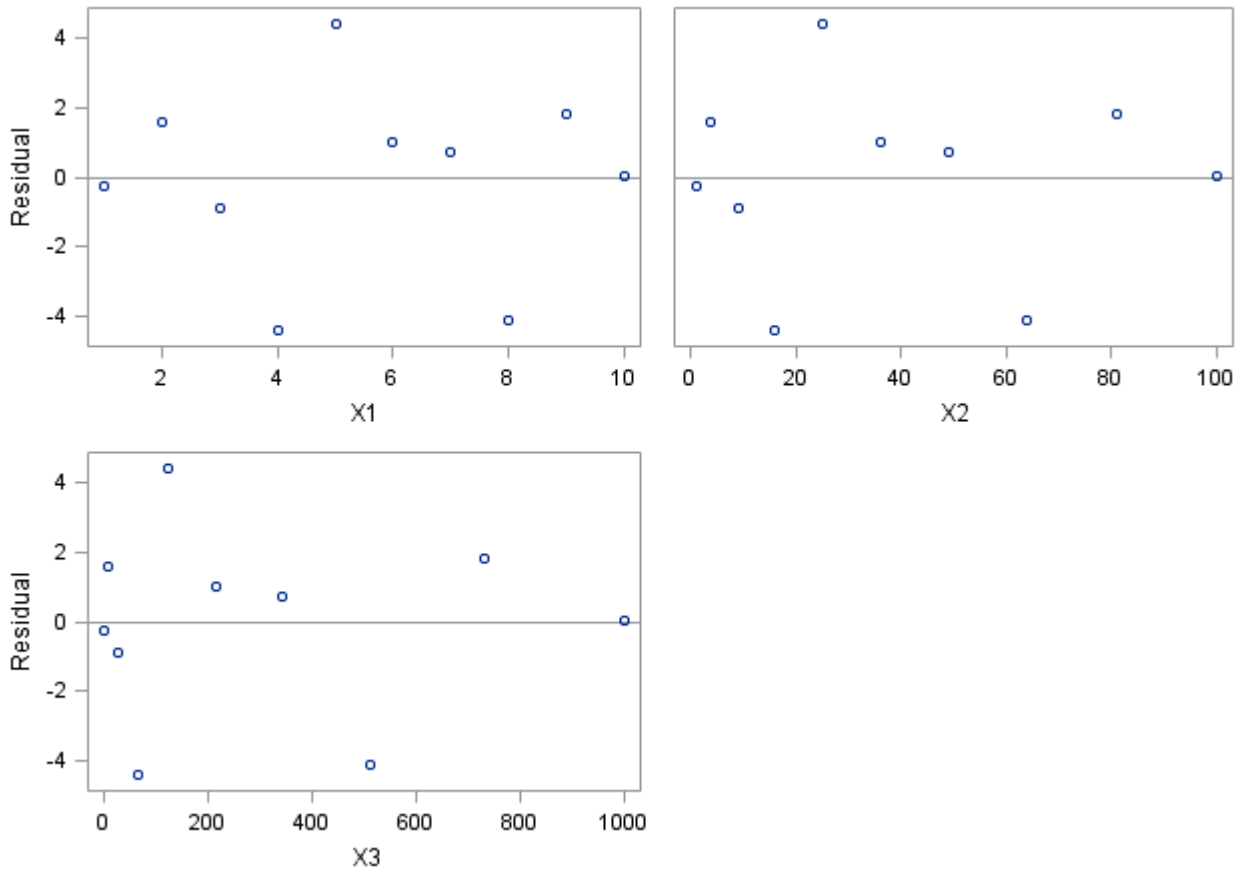


Figure 2: Residuals v.s X_i, $X_i^2$and X_i^3

When observing the pattern of the residuals from the true cost function, it can be seen that they do not have a noticeable pattern forming a certain shape, the residuals are very small and randomly distributed and is a good indication that nothing is left unexplained in the model.

Consider the following quadratic cost function of

$$\hat{Y}_i = 222.38333 - 8.02500X_i + 2.54167X_i^2$$

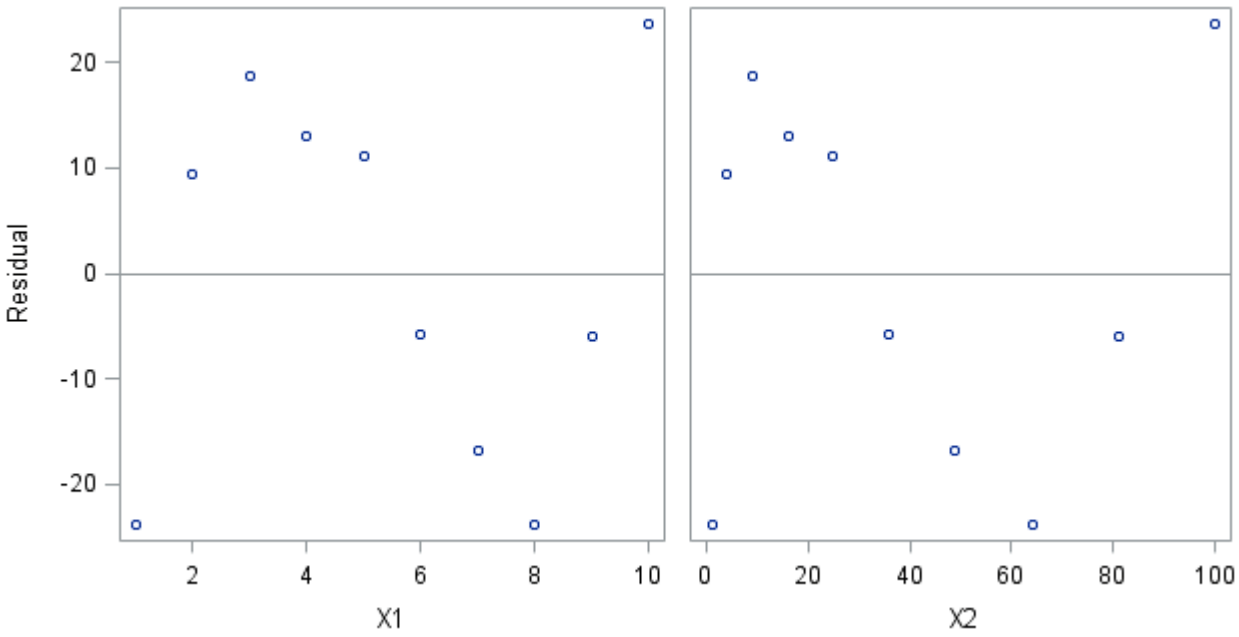


Figure 3: Residuals v.s $X_i$and X_i^2

When using this quadratic cost function we get a noticeable pattern of an almost S-shaped curve but not clear enough, meaning the residuals are larger than that from the cubic cost function discussed above. This is a clear indication that there is an unexplained pattern in the residuals which should prompt you to reconsider the model specification.

Lastly consider the linear cost function of

$$\hat{Y}_i = 166.467 + 19.933X_i$$

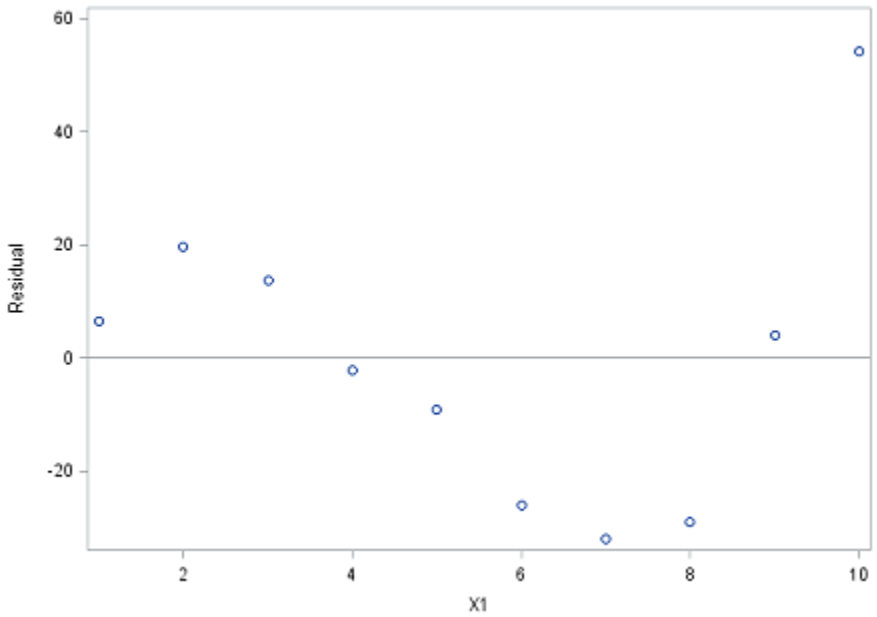


Figure 4: Residual v.s X_i

Under the linear cost function we see a very clear noticeable pattern of an S-shape. This linear cost function is rejected simply because the residuals are very large compared to the other two models with a clear systematic pattern in the residual term.

3.4 Durbin-Watson d statistic

We now examine the calculated d statistic for linear, quadratic and cubic cost functions. Figure 4 below shows the estimated cost functions with their standard errors, R^2 ,adjusted R^2 and the estimated d statistics thereof.

- $\hat{Y}_i = 166.467 + 19.933X_i$ $\quad R^2 = 0.8409$

  (19.021) (3.066) $\quad \bar{R}^2 = 0.8210$

  $d = 0.716$

- $\hat{Y}_i = 222.383 - 8.0250X_i + 2.542X_i^2$ $\quad R^2 = 0.9284$

  (23.488) (9.809) (0.869) $\quad \bar{R}^2 = 0.9079$

  $d = 1.038$

- $\hat{Y}_i = 141.767 + 63.478X_i - 12.962X_i^2 + 0.939X_i^3$ $\quad R^2 = 0.9983$

  (6.375) (4.778) (0.9856) (0.0592) $\quad \bar{R}^2 = 0.9975$

  $d = 2.70$

Figure 5: Estimated d statistics for the linear, quadratic and cubic cost functions.

Under the linear function the estimated $d = 0.716$ comparing it to $d_U = 1.320$ and $d_L = 0.879$, we find a positive auto correlation in the estimated residuals. For the quadratic cost function $d = 1.038$ is compared to the critical values of $d_U = 1.641$ and $d_L = 0.697$,this shows an indecision results but if the modified statistic test is used then we find a positive auto correlation in the estimated residuals. Under the cubic function $d = 2.70$ and we know from Durbin-Watson table that a value of 2 suggests no correlation and simply means no model specification errors, this is the correctly specified model.

We now consider the empirical results illustrated under the inclusion of irrelevant variables as follows:

3.5 The $t-$test

Considering that $Eq.(19)$ below is our true cost function

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \beta_4 X_i^4 + \mu_i \tag{19}$$

$\hat{Y}_i = 146.4167 + 57.5161X_i - 10.7802X_i^2 + 0.6415X_i^3 + 0.0136X_i^4$ $\quad R^2 = 0.9984$

(11.6084) (13.0840) (4.5315) (0.6055) (0.0274) $\quad \bar{R}^2 = 0.9972$

To test if β_4 does belong in our cost function we use the $t-$test. Specifying the hypothesis

$$H_0 : \beta_4 = 0$$

$$H_a : \beta_4 \neq 0$$

Test statistic: $t = \frac{\hat{\beta}_4 - 0}{se(\hat{\beta}_4)} = \frac{0.0136}{0.0274} = 0.4964$

with p-value=0.6416.

By comparing this value to our critical value of $t_{(0.05/2,6)} = 1.943 > t$. We do not reject H_0 at a 5% level of significance and we therefore conclude that β_4 is an irrelevant variable which is included in the model and it should therefore be removed.

3.6 The $F-$test

Treating $Eq.(19)$ as our unrestricted cost function and comparing it to a fitted restricted cost function below, that is $Eq.(20)$

$$Y_i = \beta_0 + \beta_1 X_i + \upsilon_i \tag{20}$$

$$\begin{array}{ll} \hat{Y}_i = 166.467 + 19.933 X_i & R^2 = 0.8409 \\ \quad\quad (19.021) \quad (3.066) & \bar{R}^2 = 0.8210 \end{array}$$

In order to determine whether more than one variable, that is X_i^2 and X_i^3, belong in the true cost function , we use the F test. We specify the null hypothesis

$$H_0 : \beta_2 = \beta_3 = 0 \quad (Restricted\, model)$$

$$H_a : at\, least\, one\, \beta \neq 0 \quad (Unrestricted\, model)$$

Our F-value is: $$F = \frac{(RSS_R - RSS_{UR})/m}{(RSS_{UR})/(n-k)} = \frac{(6202.53-64.74)/2}{64.74/(10-6)} = 284.42$$

By comparing this value to a critical value of $F_{(0.05,2,6)} = 5.14$, at least one $\beta \neq 0$ and therefore we reject the null hypothesis at a 5% level of significance, since $F = 284.42 > F_{(0.05,2,6)} = 5.14$. We therefore conclude that the unrestricted model is the correct model.

All results found and stated in this section shows either a certain model is in fact correctly specified or model specification errors took place and if we did encounter specification errors, we can simply apply remedies to those errors, for example, we found that $Eq.(16)$ is under-fitted, we just have to add the missing variables (X_i^2 and X_i^3) and $Eq.(19)$ is over-fitted, we then remove the extra irrelevant variables, which is X_i^4.

4 Conclusion

In this essay we considered two types of model specification errors, under-fitting and over-fitting. We discussed the consequences of the specification errors and also different test procedures which could be used to detect mis-specification of a model namely, Ramsey's RESET test, Durbin-Watson d statistic , Lagrange Multiplier , examination of residuals , $t-$test and the F- test. The test procedures were illustrated in a practical example of a cubic cost function being estimated and the following results below were obtained:

- Under the omission of relevant variable(s):

Firstly, the Ramsey's RESET test was used to detect model mis-specification and the linear cost function Eq.(16) was rejected based on the results found, where the F test statistic was greater than that of the critical F- value.

Secondly, the Lagrange Multiplier test showed that the restricted regression was rejected at a 5% level of significance, since we found that the χ^2 statistic was greater than that of the critical chi-square $\chi^2_{(0.05)}(2)$.

Thirdly, three figures were depicted for examining residuals. Figure 2 resembled a random distribution of residuals and no pattern was depicted. Figure 3 and figure 4 formed S-shaped curves, meaning residuals are large and we have a specification error, then the functions of figure 3 and figure 4 were rejected as they were not the true cost functions and figure 2 was the true cost function.

Lastly, under the Durbin-Watson d statistic, we calculated the d statistics for the linear and quadratic cost functions and both gave a positive auto correlation in residuals and this suggests that we have model specification errors.

- Under the inclusion of irrelevant variable(s):

Firstly, the t -test was considered. Since the critical t - value was greater than the t test statistic , we concluded that $\beta_4 = 0$ and the null hypothesis was not rejected at a 5% level of significance.

Secondly, the F -test was considered. We concluded that at least one $\beta \neq 0$ and therefore we rejected the null hypothesis since the critical F - value was greater than that of the F test statistics. The restricted cost function suggests specification errors and the unrestricted cost function is found to be the correctly specified cost function.

In this research essay we distinguished the difference between different model specification errors and we found that if a specification error occurred or was found in a model then it means that model is mis-specified. We mainly focused on the two types of model specification errors mentioned above. These two types were illustrated practically and some results were obtained and simple remedies thereof were also stated. Model specification errors can always be avoided simply by using the correctly specified model.

In this essay we only considered the discussion of the two types of model specification errors fully and could include a comprehensive discussion of the other types of model specification errors namely, using a wrong functional form, measurement errors (dependent and independent variables), wrong specification of the stochastic error term and assuming a normal distribution for the error term, in future research. Model mis-specification errors were not introduced in this essay and could also be considered for future research.

References

[1] Anil K Bera and Carlos M Jarque. Model specification tests: A simultaneous approach. *Journal of Econometrics*, 20(1):59–82, 1982.

[2] Alok Bhargava, Luisa Franzini, and Wiji Narendranathan. Serial correlation and the fixed effects model. *The Review of Economic Studies*, 49(4):533–549, 1982.

[3] Thomas R Buckley. Model misspecification and probabilistic tests of topology: evidence from empirical data sets. *Systematic Biology*, 51(3):509–523, 2002.

[4] Thomas D Cook and David L DeMets. *Introduction to Statistical Methods for Clinical Trials*. CRC Press, 2007.

[5] Keith Cuthbertson, Stephen G Hall, and Mark P Taylor. *Applied Econometric Techniques*, volume 274. Harvester Wheatsheaf New York, 1992.

[6] Robert F Engle. A general approach to lagrange multiplier model diagnostics. *Journal of Econometrics*, 20(1):83–104, 1982.

[7] Damoder N Gujarati. *Basic Econometrics*. Tata McGraw-Hill Education, 2009.

[8] Jerry A Hausman. Specification tests in econometrics. *Econometrica: Journal of the Econometric Society*, pages 1251–1271, 1978.

[9] David F Hendry. *Dynamic Econometrics*. Oxford University Press on Demand, 1995.

[10] David Kaplan. The impact of specification error on the estimation, testing, and improvement of structural equation models. *Multivariate Behavioral Research*, 23(1):69–86, 1988.

[11] Peter Kennedy. *A Guide to Econometrics*. MIT press, 2003.

[12] Scott B MacKenzie, Philip M Podsakoff, and Cheryl Burke Jarvis. The problem of measurement model misspecification in behavioral and organizational research and some recommended solutions. *Journal of Applied Psychology*, 90(4):710, 2005.

[13] James B Ramsey. Classical model selection through specification error tests. *Frontiers in Econometrics*, pages 13–47, 1974.

[14] James Bernard Ramsey. Tests for specification errors in classical linear least-squares regression analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 350–371, 1969.

[15] John D Sargan and Alok Bhargava. Testing residuals from least squares regression for being generated by the gaussian random walk. *Econometrica: Journal of the Econometric Society*, pages 153–174, 1983.

[16] The data analysis for this essay was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA.

Appendix

- SAS PROGRAM

```
data a;
input Y X1 X2 X3;
cards;
193 1 1 1
226 2 4 8
240 3 9 27
244 4 16 64
257 5 25 125
260 6 36 216
274 7 49 343
297 8 64 512
350 9 81 729
420 10 100 1000
;
run;
data b; set a;
nx2=x1**2;
nx2=x1**3;
x4=x1**4;
run;
proc reg data=a;
model y=x1 x2 x3/xpx i alpha=0.05 covb clb clm cli;
run;
proc reg data=a;
model y=x1 x2/xpx i alpha=0.05 covb clb clm cli;
run;
proc reg data=a;
model y=x1/xpx i alpha=0.05 covb clb clm cli;
run;
data c; set a;
y2=y**2;
y3=y**3;
run;
proc reg data=a;
model y=x1 y2 y3/xpx i alpha=0.05 covb clb clm cli;
run;
proc reg data=b;
model y=x1 x2 x3 x4/xpx i alpha=0.05 covb clb clm cli;
run;
ods graphics on;
proc reg data=a plots=(fit(nolimits));
model y=x1 x2 x3;
id x1 x2 x3;
plot y*x1x2x3;
plot y*x1x2;
plot y*x1;
run;
ods graphics off;
```

- SAS OUTPUT

The REG Procedure
Model: MODEL1
Dependent Variable: Y

Number of Observations Read	10
Number of Observations Used	10

X'X Inverse, Parameter Estimates, and SSE					
Variable	Intercept	X1	X2	X3	Y
Intercept	3.7666666667	-2.638888889	0.5	-0.027777778	141.76666667
X1	-2.638888889	2.1161939912	-0.427350427	0.0246373996	63.477661228
X2	0.5	-0.427350427	0.090034965	-0.00534188	-12.96153846
X3	-0.027777778	0.0246373996	-0.00534188	0.0003237503	0.9395881896
Y	141.76666667	63.477661228	-12.96153846	0.9395881896	64.743822844

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	38918	12973	1202.22	<.0001
Error	6	64.74382	10.79064		
Corrected Total	9	38983			

Root MSE	3.28491	R-Square	0.9983
Dependent Mean	276.10000	Adj R-Sq	0.9975
Coeff Var	1.18975		

Figure 6: Cubic cost function

Parameter Estimates							
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
Intercept	1	141.76667	6.37532	22.24	<.0001	126.16682	157.36652
X1	1	63.47766	4.77861	13.28	<.0001	51.78483	75.17049
X2	1	-12.96154	0.98566	-13.15	<.0001	-15.37337	-10.54970
X3	1	0.93959	0.05911	15.90	<.0001	0.79496	1.08421

Covariance of Estimates				
Variable	Intercept	X1	X2	X3
Intercept	40.64473323	-28.47529245	5.3953185703	-0.299739921
X1	-28.47529245	22.835081478	-4.611383393	0.2658532396
X2	5.3953185703	-4.611383393	0.9715346377	-0.057642292
X3	-0.299739921	0.2658532396	-0.057642292	0.0034934723

Figure 7: Cubic cost function

X'X Inverse, Parameter Estimates, and SSE				
Variable	Intercept	X1	X2	Y
Intercept	1.3833333333	-0.525	0.0416666667	222.38333333
X1	-0.525	0.2412878788	-0.020833333	-8.025
X2	0.0416666667	-0.020833333	0.0018939394	2.5416666667
Y	222.38333333	-8.025	2.5416666667	2791.6166667

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	36191	18096	45.37	<.0001
Error	7	2791.61667	398.80238		
Corrected Total	9	38983			

Root MSE	19.97004	R-Square	0.9284
Dependent Mean	276.10000	Adj R-Sq	0.9079
Coeff Var	7.23290		

Figure 8: Quadraic cost functon

Parameter Estimates							
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
Intercept	1	222.38333	23.48780	9.47	<.0001	166.84352	277.92315
X1	1	-8.02500	9.80949	-0.82	0.4403	-31.22077	15.17077
X2	1	2.54167	0.86908	2.92	0.0222	0.48661	4.59672

Covariance of Estimates			
Variable	Intercept	X1	X2
Intercept	551.67662698	-209.37125	16.616765873
X1	-209.37125	96.226180556	-8.308382937
X2	16.616765873	-8.308382937	0.7553075397

Figure 9: Quadratic cost function

X'X Inverse, Parameter Estimates, and SSE			
Variable	Intercept	X1	Y
Intercept	0.4666666667	-0.066666667	166.46666667
X1	-0.066666667	0.0121212121	19.933333333
Y	166.46666667	19.933333333	6202.5333333

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	32780	32780	42.28	0.0002
Error	8	6202.53333	775.31667		
Corrected Total	9	38983			

Root MSE	27.84451	R-Square	0.8409
Dependent Mean	276.10000	Adj R-Sq	0.8210
Coeff Var	10.08494		

Figure 10: Linear cost function

Parameter Estimates							
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
Intercept	1	166.46667	19.02142	8.75	<.0001	122.60319	210.33014
X1	1	19.93333	3.06558	6.50	0.0002	12.86409	27.00257

Covariance of Estimates		
Variable	Intercept	X1
Intercept	361.81444444	-51.68777778
X1	-51.68777778	9.3977777778

Figure 11: Linear cost function

X'X Inverse, Parameter Estimates, and SSE						
Variable	Intercept	X1	X2	X3	x4	Y
Intercept	10.916666667	-11.80555556	3.8541666667	-0.486111111	0.0208333333	146.41666667
X1	-11.80555556	13.868330743	-4.727564103	0.6122442372	-0.026709402	57.516122766
X2	3.8541666667	-4.727564103	1.6635222417	-0.220352564	0.0097732129	-10.78015734
X3	-0.486111111	0.6122442372	-0.220352564	0.0297040922	-0.00133547	0.6415112665
x4	0.0208333333	-0.026709402	0.0097732129	-0.00133547	0.0000607032	0.013548951
Y	146.41666667	57.516122766	-10.78015734	0.6415112665	0.013548951	61.71969697

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	38921	9730.29508	788.26	<.0001
Error	5	61.71970	12.34394		
Corrected Total	9	38983			

Root MSE	3.51339	R-Square	0.9984
Dependent Mean	276.10000	Adj R-Sq	0.9972
Coeff Var	1.27251		

Figure 12: Cubic model and adding an irrelevant variable

Parameter Estimates							
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
Intercept	1	146.41667	11.60839	12.61	<.0001	116.57636	176.25698
X1	1	57.51612	13.08395	4.40	0.0070	23.88275	91.14950
X2	1	-10.78016	4.53149	-2.38	0.0632	-22.42873	0.86841
X3	1	0.64151	0.60553	1.06	0.3379	-0.91505	2.19807
x4	1	0.01355	0.02737	0.49	0.6416	-0.05682	0.08392

Covariance of Estimates					
Variable	Intercept	X1	X2	X3	x4
Intercept	134.75467172	-145.7270623	47.575599748	-6.000526094	0.257165404
X1	-145.7270623	171.18983419	-58.35676476	7.5575057589	-0.329699236
X2	47.575599748	-58.35676476	20.534417731	-2.720018697	0.1206399477
X3	-6.000526094	7.5575057589	-2.720018697	0.3666655139	-0.016484962
x4	0.257165404	-0.329699236	0.1206399477	-0.016484962	0.0007493164

Figure 13: Cubic model and adding an irrelevant variable

Exploring robust regression

Ilke Lewis 12065103

STK795 Research Report

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

Supervisor: FHJ Kanfer

Department of Statistics, University of Pretoria

11 October 2016

Abstract

Ordinary least squares (OLS) estimators for a linear model is sensitive to unusual data. Even one extreme observation for example may have a major effect on the estimated parameters of the regression. Robust estimation regression can be used as an alternative estimation procedure to ordinary least squares regression in the case of unusual data. Robust estimation regression procedures are less influenced by unusual data and uses methods that are resistant to the possibility that one or several unknown outliers may occur in the data and will therefor provide more useful estimated models. M-estimation and bounded-influence estimation as robust estimation regression procedures is presented. In addition, comparisons of these robust estimates based on their robustness and efficiency will be done through a simulation study. A real data application is provided to compare robust estimation regression procedures with ordinary least squares regression.

Declaration

I, *Ilke Lewis,* 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.

_____
Ilke Lewis

_____
FHJ Kanfer

_____
Date

Acknowledgements

First the author would like to articulate her appreciation to Dr Frans Kanfer for all his valuable insight, guidance and suggestions. The author is also grateful to Dr. Inger for offering generous help and support. The authors would like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a post graduate bursary.

Contents

1 Introduction **6**

2 Background theory **8**
2.1 Breakdown and measuring robustness 8
2.2 M-estimation 9
2.2.1 Computing M-estimates 10
2.2.2 Objective functions 11
2.3 Bound-influence regression 12
2.3.1 Least-trimmed squares (LTS) 13

3 Application **13**
3.1 Simulation study 13
3.2 Real data application 14

4 Conclusion **18**

References **19**

Appendix **20**

List of Figures

1 Objective functions [5] 12

List of Tables

1 Comparison of estimation methods for data with 10% contamination 13
2 Comparison of estimation methods for data with 40% contamination 13
3 Comparison of estimation methods under data with 1% bad high leverage points 14
4 Comparison of estimated models for real data 18

1 Introduction

Ordinary least squares (OLS) regression [11] is used to study how the response variables $y_i's$ is related to a set of regressors $x_i's$ were (x_i, y_i) for $i = 1, 2, 3, ..., n$ is independent and identically distributed (i.i.d). The linear regression model is given by,

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where Y is a $(n \times 1)$ response variable vector, $\boldsymbol{X}$ is a $(n \times p)$ design matrix, $\boldsymbol{\beta}$ is an unknown $(p \times 1)$ parameter vector, and $\boldsymbol{\varepsilon}$ is a random error vector and is independent and identically distributed (iid) and independent of $\mathbf{X}$ with $E(\varepsilon_i|x_i) = 0$. The size of the residuals are an indication of the regression models performance. An universally used estimate for β is the OLS estimator which minimize the quantity with respect to $\boldsymbol{\beta}$,

$$\begin{aligned} \boldsymbol{Q} &= (\boldsymbol{Y} - \boldsymbol{X\beta})' (\boldsymbol{Y} - \boldsymbol{X\beta}) \\ &= \boldsymbol{Y'Y} - 2\boldsymbol{\beta' X'Y} + \boldsymbol{\beta' X' X\beta} \end{aligned}$$

Differentiating with respect to β and setting the partial derivatives equal to zero yields,

$$-2\boldsymbol{X'Y} + 2\boldsymbol{X'Xb} = 0$$

to obtain the normal equations,

$$\boldsymbol{X'Xb} = \boldsymbol{X'Y}$$

Permultiply with $(X'X)^{-1}$ to obtain the ordinary least square estimator,

$$\boldsymbol{b} = (\boldsymbol{X'X})^{-1} \boldsymbol{X'Y}$$

The OLS estimate is ideal among the class of linear unbiased estimates and it is also the most efficient unbiased estimate under the assumption that the residuals in regression models are normally distributed with $\epsilon_i \sim N\left(0, \sigma^2\right)$ and independent distributed where $i = 1, ..., n$. On the other hand, in application, residuals do not always precisely follow a normal distribution. In any given data set there might be outliers or the residuals may follow another distribution for example a t-distribution which has a heavier tail than the normal distribution. In both these cases, the ordinary least squares estimator will deviate strongly from the true value of the estimates. [9]

Outlier data can be seen as unusual data. There are different types of outliers that can influence the ordinary least squares estimates. Vertical outliers, good leverage points and bad leverage points will be discussed. [10] Vertical outlier are those observations that have unusual values in the y-dimension but not in the x-dimension. A vertical outlier has an effect on the ordinary least squares estimates, especially on the intercept estimator. A good leverage point can be seen as observation that have unusual values in the region of the explanatory variable but are still close to the regression line. A good leverage point will not have an effect on the the ordinary least squares estimates but rather on the statistical inference seeing that they inflate the estimated standard errors. A bad leverage point can be seen as the observations that have unusual values in the region of the explanatory variable but does not lay close to the regression line. A bad leverage point will effect both the estimation of the intercept and the slope of the OLS estimates.

A common method to improve the estimate sensitivity to outliers, is by transforming the data, and applying OLSs regression to the transformed data. Standard methods for outlier detection in a data set are based on initial ordinary least squares fit and using numerical or graphical procedures called regression diagnostics to detect unusual observations. But seeing that all of the above mentioned methods used by regression diagnostics are based on the initial ordinary least squares fit, the parameters and leverages may be largely influenced by extreme observations. These methods can therefore be misled by the combined action of several outliers, and may even fail to identify a single outlier. [4]

Therefor, unusual observation may have a substantial effect on the parameter estimates of the regression model. Robust estimation regression can be used as an alternative to OLS regression. [11]

Robust estimation regression is originated to avoid some boundaries of established parametric methods. OLS models make strong assumptions about the structure of data, assumptions that often do not hold in applications. This report will review some standard robust estimation regression methods and discuss their properties. Usually the properties of efficiency, the breakdown point, and the influence function is used to measure the performance of regression estimates. In this report the breakdown point of an estimator, more specifically the finite sample breakdown point of an estimator, known as the smallest fraction of *bad* data a data set may contain before estimates turn *bad*, is used to illustrate the impact of unusual data. The efficiency indicates how well the robust method works compared to OLS estimation when data precisely follow a normal distribution. Since OLS estimation is the best estimation method when the data set are normally distributed, the aim is to get the robust estimators to perform as closely to ordinary least squares estimation as possible. Therefor, high efficiency is desired for robust estimation. Robust regression methods mainly deal with the following problems:

1. Outliers only in the response domain (y-domain)
2. High leverage outliers (outliers in both the x-domain and y-domain)
3. Distribution with heavier tail than normal distribution [2]

The M-estimation, a generalization of maximum likelihood estimates (MLE) from principle, is one of the most commonly used estimation method to address problems with outliers. The three most commonly used M-estimators is known as the OLS, Huber and Tukey bisquare estimators. Sometimes the M-estimator may be vulnerable to high leverage points and therefor bounded-influence regression is used. Very high breakdown estimates do not permit for diagnosis of model misspecification, and should be avoided.[3] One of the bound-influence estimators is least-trimmed squares (LTS) regression and will be presented in the report. A comparisons of the M-estimation and least trimmed squares estimation based on their robustness and efficiency will be done through a simulation study. A real data application is also provided to compare robust regression procedures with ordinary least squares regression.

2 Background theory

2.1 Breakdown and measuring robustness

With robust methods the aim is to develop models which have beneficial behavior in an approximately normally distributed model. Fox and Weisberg [5] defined the finite sample breakdown point of an estimator to be the smallest fraction (α) of the data such that if $[n\alpha]$ tends to infinity then the value of the estimator also trends to infinite. Therefor an estimator with a high breakdown point is more robust. Consider the observed random sample $\{x_1, x_2, ..., x_n\}$ with average,

$$\overline{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

Geyer [6] rewrite (1) ,

$$\begin{aligned} \overline{x}_n &= \frac{1}{n}\left[\sum_{i=1}^{n-1} x_i + x_n\right] \\ &= \frac{n-1}{n}\bar{x}_{n-1} + \frac{1}{n}x_n \end{aligned} \tag{2}$$

with $x_1, x_2, ..., x_{n-1}$ fixed and $x_n \to \infty$, then $\overline{x}_n \to \infty$. Fox and Weisberg [5] then suggested $\bar{x}_n$ can be made as large as possible, by increasing the value of x_n, regardless of the other $n-1$ values. Geyer [6] implied that the finite sample breakdown point of an estimator is some function of n, thus for calculation purposes, the asymptotic breakdown point of an estimator is used to get a single number. Geyer [6] defined the asymptotic breakdown point to be the limit of the finite sample breakdown point as n tends to infinity.

Considering the above, the breakdown point of a sample mean and ordinary least square estimates is $\frac{1}{n}$ with a asymptotic breakdown point of 0, since even one outlier may involve in a substantial change in the estimation. The median stays within the majority data, if the minority of data trend to infinity. Although the median changes, it does not become subjectively *bad*. The breakdown point of the sample median is $\frac{1}{2}$. If a breakdown point exceeds $\frac{1}{2}$ then more than half of the data are outliers, which will make it is impossible to distinguish between the *good* and *bad* distributions. Although the sample median can achieve the finest breakdown point value, its efficiency is very low. Geyer [6] then concluded that from the breakdown point characterisation of robustness, the sample mean is the worst estimator that can be used, for the reason that it is only suitable for ideal data with no outliers and that the median is the better one of the two.

The breakdown point is very influenced by extreme values, and therefor a trimmed mean is suggested, a robust estimator less influenced by outliers. The trimmed mean is calculated by trimming $[\alpha n]$ observations on both sides, after all observations has been ordered from smallest to largest so that,

$$X_{(1)} \leq X_{([n\alpha]+1)} \leq .... \leq X_{(n-[n\alpha])} \leq X_{(n)}$$

According to Stigler [8] the trimmed mean can be written as,

$$\overline{X}_\alpha = \frac{1}{n-2n\alpha}\sum_{i=[n\alpha]+1}^{n-[n\alpha]} X_{(i)}$$

where n is the number of observations in the dataset, and $[n\alpha]$ is the proportions of the observations trimmed on both sides, with $n-2n\alpha$ as the remaining observations.

Another standard measurement of robustness is the influence function ψ. Let b be the estimate of β based on the original data and b_0 be the estimate based on the modified data which has no outliers. The sensitivity curve of b is then known as $b-b_0$. The influence function is an asymptotic version of the sensitivity curve, and measures the

rate at which β responds to a small amount of contamination at $x_0$and therefor showing the influence of a single outlier on the estimate. For robust estimators, the influence function should not trend to infinity as x becomes subjectively large. Therefore, a bounded influence function is preferred.

2.2 M-estimation

As stated previously, the ordinary least square (OLS) estimates are influenced by unusual data, therefor Fox and Weisberg [5] suggested the M-estimate and implied that the breakdown of the sample mean is equivalent to the breakdown of the OLS estimator.

Consider the follow linear model,

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_n x_{in} + \varepsilon_i \\ &= \boldsymbol{x_i'\beta} + \varepsilon_i \end{aligned} \tag{3}$$

for $i = 1, 2, ..., n$. Assuming that the model is not the problem and that $E(y \mid x) = x_i^{'}\beta$, the distribution of the residuals are unknown and can therefor be heavy-tailed with outliers. By estimating β with the estimator $\boldsymbol{b}$, we have

$$\begin{aligned} \hat{y}_i &= b_0 + b_1 x_{i1} + b_2 x_{i2} + ... + b_p x_{in} + e_i \\ &= \boldsymbol{x_i'b} \end{aligned}$$

where $e_i = y_i - \hat{y}_i$, $\boldsymbol{b} = \begin{bmatrix} b_0 \\ \vdots \\ b_p \end{bmatrix}$ and $x_i^{'} = \begin{bmatrix} \mathbf{1} & x_{i1} & \cdots & x_{in} \end{bmatrix}$.

In M-estimation the parameter $\boldsymbol{b}$ are determined by minimizing an objective function,

$$\begin{aligned} \sum_{i=1}^{n} \rho\left(e_i\right) &= \sum_{i=1}^{n} \rho\left(y_i - \hat{y}_i\right) \\ &= \sum_{i=1}^{n} \rho\left(y_i - \boldsymbol{x_i'b}\right) \end{aligned} \tag{4}$$

where $\rho\left(\cdot\right)$ is a loss function, defined as a cost for any given error e_i. For the loss function $\rho\left(\cdot\right)$ to be reasonable, $\rho\left(\cdot\right)$ should have the following properties:

- $\rho\left(e\right) \geq 0$ (positive)
- $\rho\left(0\right) = 0$
- $\rho\left(e\right) = \rho\left(-e\right)$(symmetric)
- $\rho\left(e_i\right) \geq \rho\left(e_i'\right)$ for $|e_i| > |e_i'|$.

These properties hold for the OLS estimator. Consider $\rho_{OLS}\left(e_i\right) = e^2$ for the OLS. It is clear that this complies to the properties of a lost function.[5]

- $e^2 \geq 0$ (positive)
- $\rho_{OLS}\left(0\right) = 0$
- $\rho_{OLS}\left(e\right) = \rho_{OLS}\left(-e\right)$(symmetric)
- $\rho_{OLS}\left(e_i\right) \geq \rho_{OLS}\left(e_i'\right)$ for $|e_i| > |e_i'|$.

2.2.1 Computing M-estimates

Fox and Weisberg [5] further implied that the M-estimates can be calculated by minimizing (4) since,

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{b}} \sum_{i=1}^{n} \rho\left(e_i\right) &= \frac{\partial}{\partial \boldsymbol{b}} \sum_{i=1}^{n} \rho\left(y_i - \boldsymbol{x_i'b}\right) \\ &= \sum_{i=1}^{n} \psi\left(y_i - \boldsymbol{x_i'b}\right) \boldsymbol{x}_i' \end{aligned} \tag{5}$$

where the influence function $\psi\left(e_i\right) = \frac{\partial}{\partial b}\rho\left(e_i\right)$. The influence function $\psi\left(\cdot\right)$ can be defined as the the overall sensitivity of the estimate. Re-writing (5) as a robust estimator in a form familiar to the problem like weighted least squares, the weight function can be defined as,

$$\begin{aligned} w_i &= w\left(e_i\right) \\ &= \frac{1}{e_i}\left(\frac{\partial}{\partial e}\rho\left(e_i\right)\right) \\ &= \frac{1}{e_i}\psi\left(e_i\right) \end{aligned} \tag{6}$$

From (4) we have,

$$\sum_{i=1}^{n} w_i \left(y_i - \boldsymbol{x_i'b}\right) \boldsymbol{x}_i' \tag{7}$$

and can be solved by setting equal to zero. Fox and Weisberg [5] proposed an iterative solution known as iteratively reweighted least square (IRLS) and is required for (7), considering that the weights w_i depends on the residuals, and therefor the weight in unknown until after the regression is completed. Iteratively reweighted least squares is calculated in three steps,

1. Initial estimates of $\boldsymbol{b}^{(0)}$are selected.
2. Calculate an estimate of the scale of the residuals $e_i^{(t-1)}$ together with the weights $w_i^{(t-1)} = w\left[e_i^{(t-1)}\right]$ for each repetition of t.
3. Obtain the new weighted least squares estimates by solving:

$$\boldsymbol{b}^{(t)} = \left[\boldsymbol{X'W}^{(t-1)}\boldsymbol{X}\right]^{-1}\boldsymbol{X'W}^{(t-1)}\boldsymbol{y}$$

Where $\boldsymbol{X}$ is the a matrix and $\boldsymbol{W}^{(t-1)} = diag\left\{w_i^{(t-1)}\right\}$ is the current weight matrix. Repeating the last two steps will allow the model to converge. The asymptotic co-variance matrix of $\boldsymbol{b}$ is then,

$$\nu\left(\boldsymbol{b}\right) = \frac{E\left(\psi^2\right)}{\left[E\left(\psi'\right)\right]^2}\left(\mathbf{X'X}\right)^{-1} \tag{8}$$

and the estimate $\hat{\nu}\left(\boldsymbol{b}\right)$ for $\nu\left(\boldsymbol{b}\right)$ in (8) can be written as,

$$\hat{\nu}\left(\boldsymbol{b}\right) = \frac{\sum\left[\psi\left(e_i\right)\right]^2}{\left[\sum\psi'\left(e_i\right)/n\right]^2}\left(\mathbf{X'X}\right) \tag{9}$$

Note that the estimate $\hat{\nu}\left(\boldsymbol{b}\right)$ in (9) is not reliable in small samples. [5]

2.2.2 Objective functions

Fox and Weisberg [5] further presented the three types of M-estimators with each estimators objective function and weighted function. The three methods is known as the ordinary least squares (OLS), Huber and Tukey bisquare estimators. Several choices of ρ have been proposed. If $\rho_{OLS}(e) = \frac{1}{2}e^2$ then the influence function becomes $\psi_{OLS}(e) = e$ with weight function, $w_{OLS}(e) = 1$.

From this it can be seen that an outlier will have an influence on the model, and therefor Fox and Weisberg [5] suggested the Huber estimator instead, which choose $\rho(\cdot)$ to be the loss function. For the Huber function,

$$\rho_H(e) \begin{cases} \frac{1}{2}e^2 & for|e| \leq k \\ k|e| - \frac{1}{2}k^2 & for|e| > k \end{cases}$$

The corresponding influence function for the Huber loss function is,

$$\psi_H(e) \begin{cases} e & for|e| \leq k \\ k & for|e| > k \end{cases}$$

With weight function,

$$w_H(e) = \begin{cases} 1 & for|e| \leq k \\ \frac{k}{|e|} & for|e| > k \end{cases}$$

where k is the tuning constant. The tuning constant k is selected to give reasonably high efficiency if the residuals are normally distributed. If the tuning constant k is small, there will be more resistance to the outliers and the efficiency will be lower when the residuals are normally distributed.

For the Tukey bisquare estimators,

$$\rho_B(e) = \begin{cases} \frac{k^2}{6} & \left\{1 - \left[1 - \left(\frac{e}{k}\right)^2\right]^3\right\} \quad for|e| \leq k \\ \frac{k^2}{6} & for|e| > k \end{cases}$$

The corresponding influence function is,

$$\psi_B(e) \begin{cases} e\left[1 - \left(\frac{e}{k}\right)^2\right]^2 & for|e| \leq k \\ 0 & for|e| > k \end{cases}$$

With weight function,

$$w_B(e) = \begin{cases} \left[1 - \left(\frac{e}{k}\right)^2\right]^2 & for|e| \leq k \\ 0 & for|e| > k \end{cases}$$

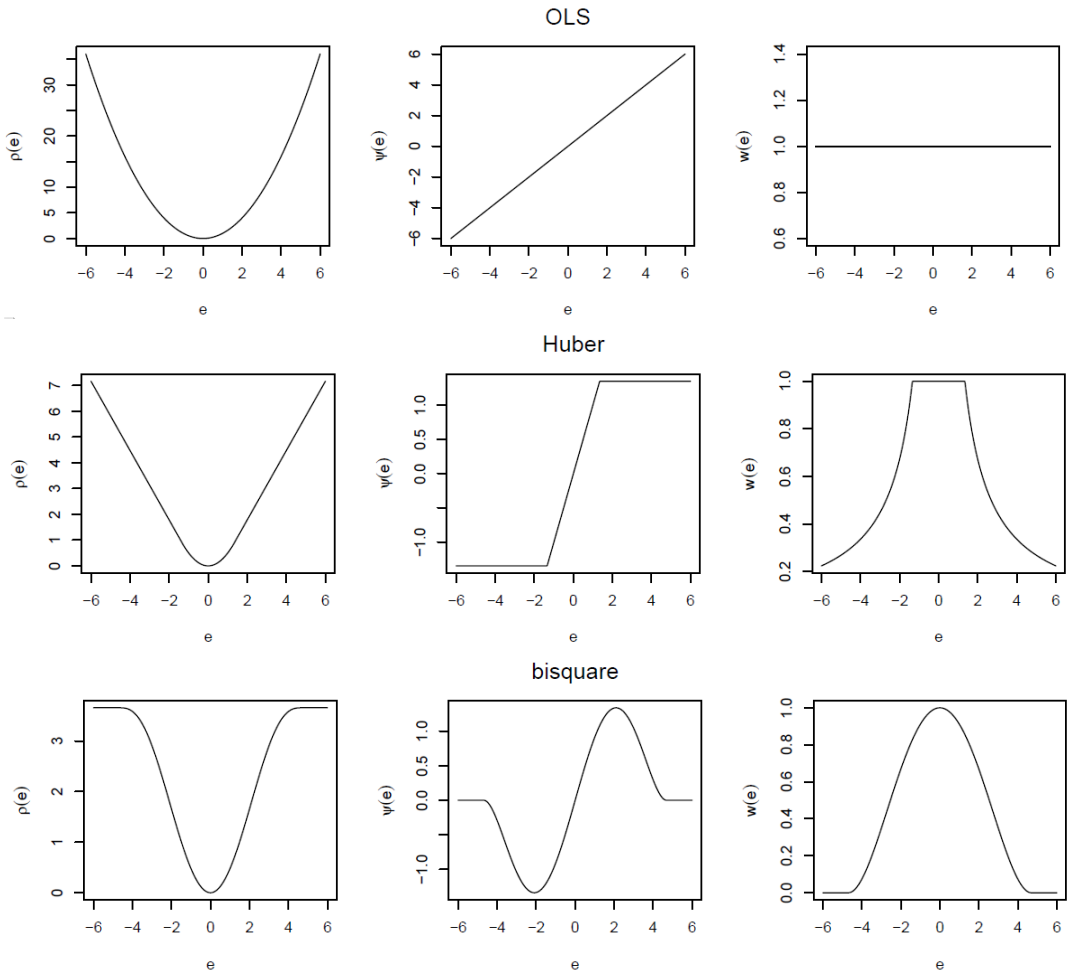


Figure 1: Objective functions [5]

In Figure 1 comparisons of the different functions for the OLS, Huber, and Tukey bisquare estimator is presented. For the Huber estimator in Figure 1 a tuning constant of $k = 1,345$ was used and $k = 4,685$ was used for the Tukey bisquare estimator. [5]

The ordinary least squares function increases rapidly as the residuals deviates from zero, with equal assigned weights to each observation. The Huber function also increase as the residuals deviates from zero, but not as quickly as the ordinary least squares function. The Huber function assigns equal weights to each observation, but declines for $|e| > k$. The Tukey bisquare function increases as the residuals deviates from zero, but levels of for $|e| > k$. The weights assigned to each observation declines as soon as the residuals departs from zero and are zero for $|e| < k$. [5]

2.3 Bound-influence regression

M-estimators with Huber function or Tukey bisquare function are robust to outliers in the response variable with high efficiency. However, the M-estimator may be just as vulnerable as ordinary least squares estimates to high leverage points and therefor bounded-influence regression is used. Very high breakdown estimates do not permit for diagnosis of model misspecication, and should be avoided, unless one is positive that the fitted model is correct.[3]

2.3.1 Least-trimmed squares (LTS)

Least trimmed squares is one of the bound-influence estimators with a breakdown point of nearly 50%. By ranking the squared error terms acceding,

$$\left(e^2\right)_{(1)}, \left(e^2\right)_{(2)}, ..., \left(e^2\right)_{(n)} \tag{10}$$

To minimize the sum of the smallest possible m of 10, least-trimmed squares estimator uses the regression coefficient $\boldsymbol{b}$, were $m = \lfloor \frac{n}{2} \rfloor + \lfloor \frac{(k+2)}{n} \rfloor$, and $\lfloor \rfloor$denotes rounding down to the next smallest integer,

$$LTS\left(\boldsymbol{b}\right) = \sum_{i=1}^{m} \left(e^2\right)_{(i)}$$

Bounded-Influence estimators can be given an output of unreasonable results, and there's no straightforward formula to be used for coefficient standard errors [7].

3 Application

3.1 Simulation study

Many methods have been developed in response to the problems unusual observations causes. This report aims to demonstration the advantages of the different objective functions and some available robust techniques. This report will explore three different regression estimates namely the M-estimate using Huber weights, M-estimate using Tukey weights, and LTS estimation. A simulation study will be preformed on these three robust estimation method, together with ordinary least squares (OLS) estimation, and comparisons will be made based on their robustness and efficiency under different scenarios. The data analysis for this essay was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA.

In order to obtain simulated data for the comparison, a 1000 random observation where generated. The first 900 observations are from a linear model, and to allow 10% contamination in the data, the last 100 observations are significantly biased in the y-direction. The parameter estimates for M-estimation and LTS estimation were generated with the ROBUSTREG procedure. The OLS estimates were generated with the REG procedure. These estimates are shown in Table 1.

Estimation methods	Intercept	β_1	β_2
OLS estimates	19.06712	3.55485	2.12341
Robust estimates (M-estimation, Huber)	10.1054	4.9972	3.0088
Robust estimates (M-estimation, bisquare)	10.0024	5.0077	3.0161
Robust Estimates (LTS estimation)	10.0083	5.0316	3.0396

Table 1: Comparison of estimation methods for data with 10% contamination

While the OLS estimate did not correctly estimate the regression coefficients for the main model for data with 10% contamination the M-estimation and LTS estimation did. The OLS analysis with 10% contaminated data indicates that X1 and X2 have a significant influence on y at a 5% level of significant.

The next scenario demonstrates the estimations under 40% contaminated data.

Estimation methods	Intercept	β_1	β_2
M-estimates with default settings for 40% contaminated data	44.8991	2.4309	1.3742
M-estimates (tuned) for 40% contaminated data	10.0137	4.9905	3.0399
LTS estimation (default setting) for data with 40% contamination	24.0106	18.0792	4.7076
LTS estimation (tuned) for data with 40% contamination	10.0276	4.9970	3.0656

Table 2: Comparison of estimation methods for data with 40% contamination

It can be seen in Table 2 that the M-estimation method with default options, did not correctly estimate the regression coefficients for the main model. Therefor the constant c was modified for the M-estimation method since the breakdown values of the estimates can be increased and thus one can capture the correct model. Similarly, the constant H can also be modified for the LTS method and can then correctly estimate the main model with these methods.

Regardless of the value of constant c used, the M-estimator may be just as vulnerable as ordinary least squares estimates to high leverage points and can therefor fail to correctly estimate the main model. For this reason bounded-influence regression methods is used instead. The LTS in PROC ROBUSTREG are robust to bad leverage points, and will correctly estimate the main model. The next scenario demonstrates the estimations under data with 1% bad high leverage points.

Estimation methods	Intercept	β_1	β_2
M-estimates for data with 1% leverage points	44.8991	2.4309	1.3742
LTS estimation for data with 1% leverage points	9.9742	5.0010	3.0219

Table 3: Comparison of estimation methods under data with 1% bad high leverage points

3.2 Real data application

For the real data application the data set crime will be used with 51 observation and can be found in [1]. The data set contains the following variables: state id (**sid**), state name (**state**), violent crimes per 100,000 people (**crime**), murders per 1,000,000 (**murder**), percent of population living in metropolitan areas (**pctmetro**), percent of population that is white (**pctwhite**), percent of population with a high school education or above (**pcths**), percent of population living under poverty line (**poverty**), and percent of population that are single parents (**single**). The variable pctmetro, pctwhite, pcths, poverty and single will be used in the regression procedure to predict crime.

Output 1: Ordinary least squares (OLS) estimates

The REG Procedure

Model: MODEL1
Dependent Variable: crime

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-1795.90448	668.78846	-2.69	0.0101
pctmetro	1	7.60881	1.29527	5.87	<.0001
pctwhite	1	-4.48291	2.77907	-1.61	0.1137
pcths	1	8.64644	7.82602	1.10	0.2751
poverty	1	26.24416	11.08327	2.37	0.0222
single	1	109.46660	20.35989	5.38	<.0001

The Ordinary least squares (OLS) analysis shown in Output 1 indicates that pctmetro, poverty and single have a significant influence on crime at the 5% level.

Output 2: M-estimation summary statistics information

The ROBUSTREG Procedure

Model Information

Data Set	WORK.CRIME
Dependent Variable	crime
Number of Independent Variables	5
Number of Observations	51
Method	M Estimation

Summary Statistics

Variable	Q1	Median	Q3	Mean	Standard Deviation	MAD
pctmetro	48.5000	69.8000	84.0000	67.3902	21.9571	22.2390
pctwhite	79.3000	87.6000	92.6000	84.1078	13.2528	9.7852
pcths	73.1000	76.7000	80.1000	76.2235	5.5921	5.0408
poverty	10.7000	13.1000	17.4000	14.2588	4.5842	4.2995
single	10.0000	10.9000	12.1000	11.3255	2.1215	1.4826
crime	326.0	515.0	780.0	612.8	441.1	345.4

Output 3: M-estimation model fitting information

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-1258.80	580.3849	-2396.33	-121.262	4.70	0.0301
pctmetro	1	5.6854	1.1241	3.4823	7.8886	25.58	<.0001
pctwhite	1	-6.5785	2.4117	-11.3053	-1.8516	7.44	0.0064
pcths	1	4.2256	6.7915	-9.0855	17.5368	0.39	0.5338
poverty	1	26.4461	9.6182	7.5947	45.2975	7.56	0.0060
single	1	119.7424	17.6686	85.1125	154.3722	45.93	<.0001
Scale	1	145.0494					

The M-estimates can be seen in Output 2 and Output 3. Besides the individuals living in metropolitan areas (**pctmetro**), the individuals living under poverty line (**poverty**) as well as the individuals that are single parents (**single**), the robust analysis also indicates that the individuals that is white (**pctwhite**) is significant. This new finding is explained in Output 4, where the outliers and leverage points are identified with asterisks. These unusual observations are defined by the standardized robust residuals and robust MCD distances that exceed the corresponding cutoff values displayed in the diagnostics summary.

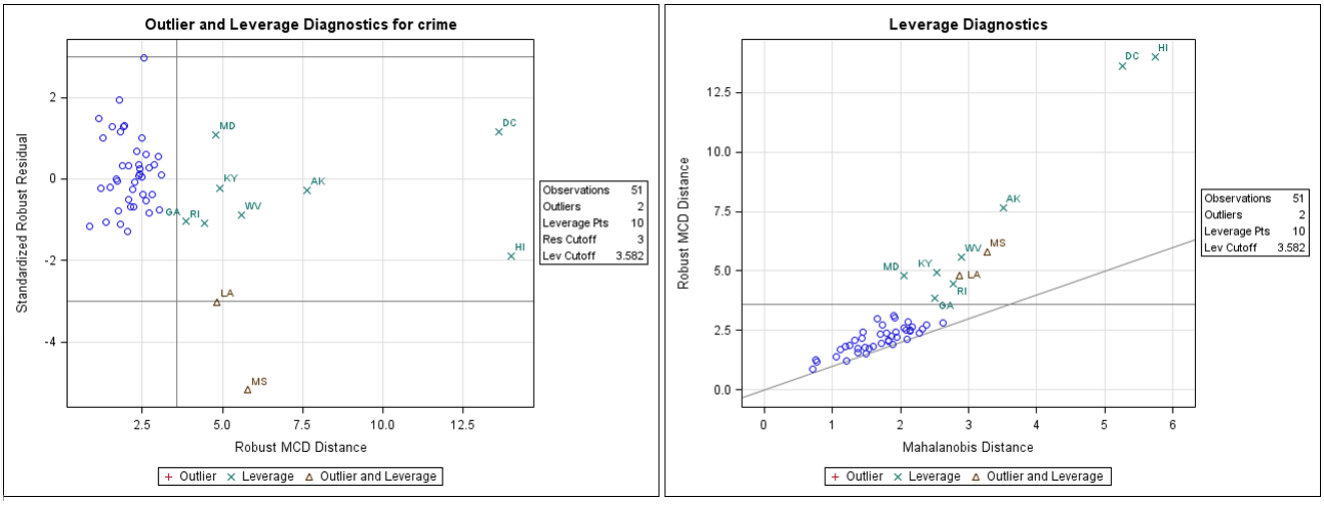
Output 4: Diagnostics

Diagnostics

Obs	state	Mahalanobis Distance	Robust MCD Distance	Leverage	Standardized Robust Residual	Outlier
1	AK	3.5116	7.6496	*	-0.2900	
10	GA	2.4987	3.8528	*	-1.0384	
11	HI	5.7436	14.0050	*	-1.8997	
17	KY	2.5270	4.9202	*	-0.2192	
18	LA	2.8690	4.8222	*	-3.0181	*
20	MD	2.0455	4.7898	*	1.0873	
25	MS	3.2725	5.7852	*	-5.1739	*
39	RI	2.7804	4.4392	*	-1.0745	
49	WV	2.8921	5.5983	*	-0.8888	
51	DC	5.2537	13.6224	*	1.1587	

Output 4 indicates that LA, the 18th state in the data and MS, the 25th state in the data, are outliers. Output 4 also identifies leverage points based on the robust MCD distances, however, there are two high-leverage points in this data set, HI, the 11th state and DC the 51th state. There are two valuable scatter plots for illustrating these outliers and leverage points identified in Output 4, namely the standardized robust residuals against the robust distances and the robust distances against the classical Mahalanobis distances. These corresponding graphs can be seen in Output 5.

Output 5: Residuals by distance plot and distance by distance plot for crime data



As mentioned previously M-estimator may be just as vulnerable as ordinary least squares estimates to high leverage points and can therefor fail to correctly estimate the main model. For this reason bounded-influence regression methods is used instead. The LTS in PROC ROBUSTREG are robust to bad leverage points, and will correctly estimate the main model.

Output 6: LTS estimates

The ROBUSTREG Procedure
LTS Parameter Estimates

Parameter	DF	Estimate
Intercept	1	-84.7267
pctmetro	1	3.7482
pctwhite	1	-17.0630
pcths	1	6.7194
poverty	1	21.6148
single	1	95.4139
Scale (sLTS)	0	95.3879
Scale (Wscale)	0	122.9121

Output 7: Diagnostics and LTS R Square

Diagnostics

Obs	state	Mahalanobis Distance	Robust MCD Distance	Leverage	Standardized Robust Residual	Outlier
1	AK	3.5116	7.6496	*	-1.3898	
9	FL	2.3285	2.5589		3.8308	*
10	GA	2.4987	3.8528	*	-2.0059	
11	HI	5.7436	14.0050	*	-6.6372	*
17	KY	2.5270	4.9202	*	0.3736	
18	LA	2.8690	4.8222	*	-3.6410	*
20	MD	2.0455	4.7898	*	0.2368	
25	MS	3.2725	5.7852	*	-7.1985	*
39	RI	2.7804	4.4392	*	-0.3289	
49	WV	2.8921	5.5983	*	-0.3336	
51	DC	5.2537	13.6224	*	0.0329	

R-Square for LTS Estimation

R-Square	0.9228

Output 7 indicates that FL, HI, LA and MS are outliers. Output 4 also identifies leverage points based on the robust MCD distances, however, there are two high-leverage points in this data set, HI, the 11th state and DC the 51th state.

Output 8: Final Weighted LS Estimates

Parameter Estimates for Final Weighted Least Squares Fit

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-635.428	496.6754	-1608.89	338.0376	1.64	0.2008
pctmetro	1	4.8578	0.9324	3.0304	6.6853	27.14	<.0001
pctwhite	1	-14.5091	3.1080	-20.6006	-8.4175	21.79	<.0001
pcths	1	10.1539	5.5917	-0.8056	21.1135	3.30	0.0694
poverty	1	28.8548	7.8996	13.3719	44.3376	13.34	0.0003

Output 8 shows the least squares estimates calculated after deleting the detected outliers showed in Output 7. In table 4 a comparison can be made for the different estimation methods used for the real data application .

Method	Estimated model
OLS	$Y = -1795.90 + 7.61X_1 - 4.48X_2 + 8.65X_3 + 26.24X_4 + 109.47X_5$
M-estimation	$Y = -1258.80 + 5.69X_1 - 6.58X_2 + 4.23X_3 + 26.45X_4 + 119.74X_5$
LTS estimation	$Y = -84.73 + 3.75X_1 - 17.06X + 6.72X_3 + 21.61X_4 + 95.41X_5$
Weighted OLS	$Y = -635.43 + 4.86X_1 - 14.51X_2 + 10.15X_3 + 28.85X_4$

Table 4: Comparison of estimated models for real data

4 Conclusion

Ordinary least squares models make strong assumptions about the structure of data, assumptions that often do not hold in application. Robust regression was used as an alternative estimation procedure to ordinary least squares regression in the case of unusual data as in application. This report reviewed some standard robust regression methods and discuss their properties. In application the conclusion was made in both the simulated data and real data that the robust estimation methods estimated the models more correctly than OLS estimation methods. Therefor robust regression procedures are proven to be less influenced by unusual data by making use of methods that are resistant to the possibility that one or several unknown outliers may occur in the data and therefor provided more useful estimated models than OLS regression.

References

[1] Alan Agresti and Barbara Finlay. *Statistical Models for the Social Sciences*. Upper Saddle River, NJ: Prentice-Hall, 1997.

[2] C Chen. Robust regression and outlier detection with the robustreg procedure. *SAS Institute Inc. Cary, NC*, 2002.

[3] RD Cook, DM Hawkins, and S Weisberg. Comparison of model misspecification diagnostics using residuals from least mean of squares and least median of squares fits. *Journal of the American Statistical Association*, 87(418):419–424, 1992.

[4] John Fox. *Regression Diagnostics: An Introduction*, volume 79. Sage, 1991.

[5] John Fox and Sanford Weisberg. *An R Companion to Applied Regression*. Sage, 2010.

[6] Charles J Geyer. Breakdown point theory notes. 2006.

[7] Leonard A Stefanski. A note on high-breakdown estimators. *Statistics & Probability Letters*, 11(4):353–358, 1991.

[8] Stephen M Stigler. Do robust estimators work with real data? *The Annals of Statistics*, pages 1055–1098, 1977.

[9] Y Susanti, H Pratiwi, et al. M estimation, s estimation, and mm estimation in robust regression. *International Journal of Pure and Applied Mathematics*, 91(3):349–360, 2014.

[10] Vincenzo Verardi and Christophe Croux. Robust regression in stata. *The Stata Journal, pages 439-453*, 2008.

[11] Chun Yu, Weixin Yao, and Xue Bai. Robust linear regression: A review and comparison. *Kansas State University, Manhattan, Kansas, USA 66506-0802.*, 2014.

Appendix

Simulation study

```
data a (drop=i);
      do i=1 to 1000;
            x1=rannor(1234);
            x2=rannor(1234);
            e=rannor(1234);
            if i > 900 then y=100 + e;
            else y=10+5*x1+3*x2+0.5*e;
            output;
      end;
run;

proc reg data=a;
model y = x1 x2;
output out=t student=res cookd=cookd h=lev;
run;

proc robustreg data=a method=m (wf=huber);
model y = x1 x2;
run;

proc robustreg data=a method=m (wf=bisquare);
model y = x1 x2;
run;

proc robustreg data=a method=lts ;
model y = x1 x2;
run;
```

```
data b (drop=i);
      do i=1 to 1000;
            x1=rannor(1234);
            x2=rannor(1234);
            e=rannor(1234);
            if i > 600 then y=100 + e;
            else y=10 + 5*x1 + 3*x2 + .5 * e;
            output;
      end;
run;

proc robustreg data=b method=m ;
model y = x1 x2;
run;

proc robustreg data=b method=m(wf=bisquare(c=2));
model y = x1 x2;
run;

proc robustreg data=b method=lts;
model y = x1 x2;
run;

proc robustreg data=b method=lts(h=450);
model y = x1 x2;
run;
```

```
data c (drop=i);
      do i=1 to 1000;
            x1=rannor(1234);
            x2=rannor(1234);
            e=rannor(1234);
            if i > 600 then y=100 + e;
            else y=10 + 5*x1 + 3*x2 + .5 * e;
            if i < 11 then x1=200 * rannor(1234);
            if i < 11 then x2=200 * rannor(1234);
            if i < 11 then y= 100*e;
      output;
end;
run;

proc robustreg data=b method=m ;
model y = x1 x2;
run;

proc robustreg data=c method=lts(h=502);
model y = x1 x2;
run;
```

Real data application

```
data crime;
input state$ crime murder pctmetro pctwhite pcths poverty single;
datalines;
 AK   761   9.0   41.8   75.2   86.6    9.1     14.3
 AL   780  11.6   67.4   73.5   66.9   17.4     11.5
                       .
                       .
                       .
WY   286   3.4   29.7   95.9   83.0   13.3     10.8
 DC  2922  78.5  100.0   31.8   73.1   26.4     22.1
run;

proc reg data=crime;
model crime=pctmetro pctwhite pcths poverty single;
run;

ods graphics on;
proc robustreg data=crime plots=all;
model crime=pctmetro pctwhite pcths poverty single / diagnostics leverage;
id state;
run;
ods graphics off;

proc robustreg method=lts(h=33) fwls data=crime;
model crime=pctmetro pctwhite pcths poverty single / diagnostics leverage ;
id state;
run;
```

Generalized logistic distributions

Magomarele Malapane 12023052

WST795 Research Report

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

Supervisor: Ms B.V Omachar, Co-supervisor: Dr P.J van Staden

Department of Statistics, University of Pretoria

02 November 2016

Abstract

The logistic distribution is frequently used as a growth model in many problems. It has often been selected as an alternative to the normal distribution because of its higher kurtosis and longer tails. This paper discusses three different generalizations of the logistic distribution, each with two shape parameters. The three different types of generalizations are the quantile-based generalized logistic (GLO) distribution developed by [7], quantile-based GLO distribution possessing skewness-invariant measures of kurtosis, developed by [11], and the quantile-based skew GLO studied by [2].The L-moment ratio diagrams for each distribution are used to compare their flexibility with regards to distributional shape.

Declaration

I, *Magomarele Tlhologelo Cliford Malapane,* 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.

_ _

Magomarele Tlhologelo Cliford Malapane

_ _

Brenda V. Omachar

_ _

Paul J. van Staden

_ _

02 November 2016

Acknowledgements

The author would like to thank the Centre for Artificial Intelligence Research (CAIR) and PSG konsult for the financial support in the form of a postgraduate bursary.

Contents

1 Introduction **6**

2 Background history/Literature Review **6**

3 The report objective **7**

4 Generalizations of the GLO distributions **8**
4.1 Quantile-based generalized logistic distribution(GLO_{vsk}) 8
4.2 Generalized skew logistic model($GSLO_{QB}$) 10
4.3 The quantile-based generalized logistic distribution (GLO_{QB}) 12

5 The $L-$moment ratio diagrams **14**

6 Conclusion **16**

Appendix **18**

List of Figures

1 Graph of logistic and normal cumulative distribution functions 6
2 The density curves for the quantile-based generalized logistic distribution(GLO_{vsk}). 10
3 The probability density curves for the generalized skew logistic model($GSLO_{QB}$). 12
4 The density curves for the quantile-based generalized logistic distribution (GLO_{QB}). 13
5 The $L-$ moment ratio diagram for the GLO_{vsk}. 14
6 The $L-$ moment ratio diagram for the $GSLO_{QB}$. 15
7 The $L-$ moment ratio diagram for the GLO_{QB}. 15
8 Comparing the flexibility of the different generalizations via the $L-$moment ratio diagrams. 16

1 Introduction

The logistic distribution is a continuous symmetric distribution. It has a similar appearance to the normal distribution in shape, but it has longer tails and thus a higher level of kurtosis - measure of the distribution tail (heavy or light tailed) relative to the normal. Consider the following cumulative distribution function(CDF) of the standardized normal and the logistic distributions respectively:

$$G_1(x) = \frac{1}{2} erf\left(\frac{x}{\sqrt{2}}\right) \tag{1}$$

and

$$G_2(x) = \left[1 + e^{\left(-\frac{\pi x}{\sqrt{3}}\right)}\right]^{-1}, \tag{2}$$

where $erf(x)$ is the error function defined as:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_{-\infty}^{x} e^{-t^2} dt.$$

From Figure 1, both $G_1(x)$ and $G_2(x)$ are symmetric about $x = 0$, so it makes sense where suitable to replace the normal distribution with the logistic distribution, in order to simplify theoretical analysis.

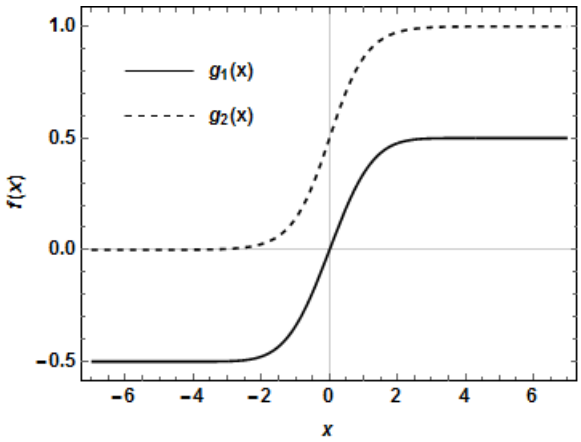


Figure 1: Graph of logistic and normal cumulative distribution functions .

The logistic distribution can be defined in terms of its CDF, probability density function, PDF, and quantile function, $Q(p)$. The book by [1] discusses in great detail of its application and various properties. Sometimes distributions tend to have no closed-form expressions for their cumulative distribution functions, $F(x)$ or their probability density functions, $f(x)$. In such cases, a quantile-based distribution comes into play and it is defined in terms of its quantile function, $Q(p)$, and quantile density function, $q(p)$. Examples of such distributions are Tukey's lambda distribution [9], various types of generalized lambda distributions [8, 10] and the Davies distribution [4].

2 Background history/Literature Review

The simple form of the logistic distribution enabled researchers to propose several generalizations of this distribution. The generalized distributions are quite flexible in distributional shape. They are indexed by

one or two shape parameters, to introduce skewness (measure of symmetry) and to alter their tail weights. In common practice, the data is not always symmetric. The need for the generalized distributions is essential to model such data.

The new quantile-based generalized logistic distribution with two shape parameters is proposed by [7]. The quantile function of the GLO, proposed by [5], is used as the building block to create the new quantile-based distribution. The quantile-based GLO is highly flexible with regards to distributional shape, in which it displays extensive levels of skewness and kurtosis. [7] also discusses distributional properties through L-moments. An estimation algorithm for estimating the distribution's parameters, with the method of L-moments estimation is also proposed.

[6] defines the L-moment as an expectation of linear combinations of order statistics. The first order L-moment, L_1,is the L-location and second order L-moment, L_2 is the L-scale. The r^{th} order $L-$ moments are rarely used as their variability and increases when $r > 2$, hence the moment-ratios are used to give skewness and kurtosis measures. The r^{th}order L-moment ratio is given by $\tau_r = \frac{L_r}{L_2}$, with τ_3 and τ_4 as the L-skewness and $L-$kurtosis moment ratios respectively.

The generalization of the quantile-based skew logistic distribution is proposed by [11] using the quantile-based approach, which was originally introduced by [3] . [11] extended his work by investigating further properties of the skew logistic distribution, and provides closed-form estimators for the parameters. The work was further discussed in detail by [2], where he presents and provides a generalization of this model. In addition, he discusses its properties as well as a method of estimation for its parameters.

A quantile-based generalized logistic distribution possessing skewness-invariant measures of kurtosis is presented by [11]. They showed that the distribution possesses kurtosis measures based on L-moments and quantiles, which are skewness invariant.

Distributions that have no closed-form expressions for their either cumulative distribution functions, $F(x)$, or their probability density functions, $f(x)$, generate moments that are complex in nature. Subsequently, the method of moments estimation is complicated to use in the estimation of the parameters. Alternative methods such as the method of percentiles and method of $L-$Moments can be implemented to simplify the theoretical analysis.

3 The report objective

The report discusses three different generalizations of the logistic distribution, each with two shape parameters. The main idea of including the two shape parameters is to provide greater flexibility of the distributions and in modeling skewed data. The three different types of generalizations mentioned above will be compared in terms of their flexibility. The L-moment ratio diagrams for each distribution is used to make the comparison.

The structure of the report is outlined in the following manner.

- In Section 4, the three different generalizations of the GLO are defined. Their quantile functions, $Q(p)$, quantile density functions, $q(p)$, density quantile functions, $f_p(p)$, and $L-$Moments are discussed.
- Section 5 presents the application to compare the flexibility of the different generalization via the $L-$moment ratio diagrams.

4 Generalizations of the GLO distributions

The random variable X is said to have a logistic distribution if its cumulative distribution is given by

$$F_X(x) = \frac{1}{1+e^{-\frac{x-\mu}{\sigma}}}, -\infty < x < \infty.$$

The corresponding probability density function is given by

$$f_X(x) = \frac{-\frac{x-\mu}{\sigma}}{\sigma\left(1+e^{-\frac{x-\mu}{\sigma}}\right)^2}, -\infty < x < \infty$$

where $\mu(-\infty < \mu < \infty)$ is the location parameter and $\sigma > 0$ is the scale parameter. The generalization of logistic distribution is obtained through the addition of one or more shape parameters to provide greater flexibility ot the distribution. All the generalizations of the logistic distribution in this paper have two shape parameters. The case where the distributions tend to have no closed-form expression, for either their cumulative distribution or their probability density function, the quantile-based distribution is defined in terms of its quantile function. In this paper, generalizations of the logistic distribution are defined in terms of their quantile function and are therefore known as quantile-based distribution.

4.1 Quantile-based generalized logistic distribution(GLO_{vsk})

A quantile-based generalized logistic distribution possessing skewness-invariant measures of kurtosis is studied by [11]. The quantile function, $Q(p)$, quantile density function, $q(p)$, and the L-Moments of the GLO_{vsk} are presented.

Definition 1. Let X be a real-valued random variable. X is said to have a GLO_{vsk} i.e $X \sim GLO_{vsk}(\alpha, \beta, \lambda, \delta)$, if the quantile function is defined as

$$Q(p) = \alpha + \beta\left(\frac{1-\delta}{\lambda}\left(\left(\frac{p}{1-p}\right)^{\lambda} - 1\right) + \frac{\delta}{\lambda}\left(1 - \left(\frac{1-p}{p}\right)^{\lambda}\right)\right), 0 \le p \le 1 \tag{3}$$

where α and $\beta(>0)$ are location and scale parameters respectively, and where λ and $0 \le \delta \le 1$ are the shape parameters.

Theorem 2. *The quantile density and density quantile functions of the GLO_{vsk} are*

$$q(p) = \frac{\left(\frac{p}{1-p}\right)^{\lambda}\beta(\delta-1) - \left(\frac{1}{p}-1\right)^{\lambda}\beta\delta}{(p-1)p}, 0 < p < 1, \tag{4}$$

and

$$f_p(p) = \frac{(p-1)p}{\left(\frac{p}{1-p}\right)^{\lambda}\beta(\delta-1) - \left(\frac{1}{p}-1\right)^{\lambda}\beta\delta}, 0 < p < 1$$

respectively.

Proof. $q(p)$ is obtained by differentiating Eq 3 with respect to p.

The result are shown as follows:

$$Q(p) = \alpha + \beta\left\{\frac{(1-\delta)}{\lambda}\left(\frac{p}{1-p}\right)^{\lambda} - \frac{(1-\delta)}{\lambda} + \frac{\delta}{\lambda} - \frac{\delta}{\lambda}\left(\frac{1-p}{p}\right)^{\lambda}\right\}$$

then

$$\begin{aligned} q(p) &= \frac{dQ(p)}{dp} \\ &= \beta\left\{\frac{(1-\delta)}{\lambda}\lambda\left(\frac{p}{1-p}\right)^{\lambda-1}\frac{d}{dp}\left(\frac{p}{1-p}\right)-\frac{\delta}{\lambda}\lambda\left(\frac{1-p}{p}\right)^{\lambda-1}\frac{d}{dp}\left(\frac{1-p}{p}\right)\right\} \\ &= \beta\left\{(1-\delta)\left(\frac{p}{1-p}\right)^{\lambda-1}\left(\frac{1}{1-p}\right)^{2}+\delta\left(\frac{1-p}{p}\right)^{\lambda-1}\left(\frac{1}{p}\right)^{2}\right\} \\ &= \beta\left\{(1-\delta)\,p^{\lambda-1}\left(\frac{1}{1-p}\right)^{\lambda+1}+\delta\,(1-p)^{\lambda-1}\left(\frac{1}{p}\right)^{\lambda+1}\right\} \\ &= \frac{\left(\frac{p}{1-p}\right)^{\lambda}\beta\,(\delta-1)-\left(\frac{1}{p}-1\right)^{\lambda}\beta\delta}{(p-1)\,p} \end{aligned}$$

In order to obtain $f_p(p)$, the reciprocal of $q(p)$ in Eq 4 is taken and the following result is obtained.

$$\begin{aligned} f_p(p) &= \frac{1}{q(p)} \\ &= \frac{1}{\beta\left\{(1-\delta)\,p^{\lambda-1}\left(\frac{1}{1-p}\right)^{\lambda+1}+\delta\,(1-p)^{\lambda-1}\left(\frac{1}{p}\right)^{\lambda+1}\right\}} \\ &= \left(\beta\left\{(1-\delta)\,p^{\lambda-1}\left(\frac{1}{1-p}\right)^{\lambda+1}+\delta\,(1-p)^{\lambda-1}\left(\frac{1}{p}\right)^{\lambda+1}\right\}\right)^{-1} \\ &= \frac{(p-1)\,p}{\left(\frac{p}{1-p}\right)^{\lambda}\beta\,(\delta-1)-\left(\frac{1}{p}-1\right)^{\lambda}\beta\delta} \end{aligned}$$

□

Theorem 3. *Let $X \sim GLO_{vsk}(\alpha,\beta,\lambda,\delta)$. The L-Moments of the GLO_{vsk}, L- location(L_1), L− scale(L_2), L−*skewness(τ_3) and *L- kurtosis(τ_4) respectively exist for $-1<\lambda<1$ and are presented as follows:*

$$L_1 = \alpha+\beta(1-2\delta)\left(\pi Csc\left[\pi\lambda\right]-\frac{1}{\lambda}\right),$$

$$L_2 = \pi\beta\lambda Csc\left[\pi\lambda\right],$$

$$\tau_3 = \lambda(1-2\delta),$$

and

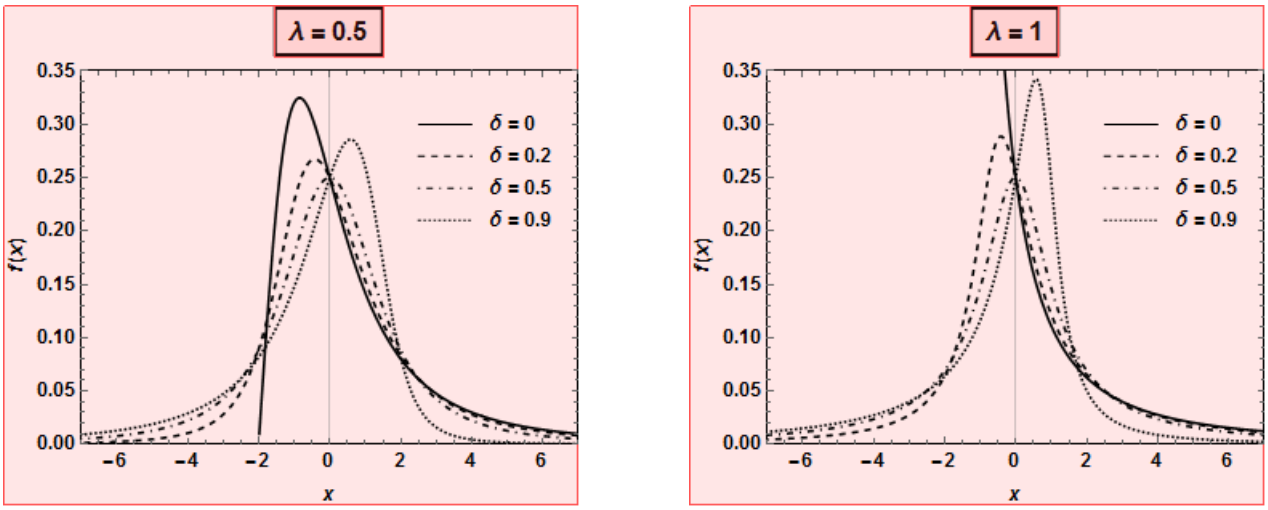
$$\tau_4 = \frac{1}{6}(1+5\lambda^2),$$

where Csc[·] is a cosec function.

Proof. See [11] for the results. □

The $L-$kurtosis moment ratio is invariant to any values of δ(scale parameter). Meaning that, for a given value of δ the $L-$kurtosis remains unchanged [11].

Figure 2 shows the probability density curves for the GLO_{vsk} for selected values of λ and δ. Without loss of generality, $\alpha = 0$ and $\beta = 1$.



(a) (b)

Figure 2: The density curves for the quantile-based generalized logistic distribution(GLO_{vsk}).

The basic properties of the quantile-based generalized logistic model(GLO_{vsk}) are listed below:

- The probability density curve of the GLO_{vsk} reduces to the logistic distribution when $\lambda = 0$.
- When $\delta = 0$ or $\delta = 1$, the GLO_{vsk} is $J-$shaped for $|\lambda| \geq 1$, unimodal for all other combinations of values of λ and δ. See Figure 2b.
- The distribution is symmetric when $\delta = 0.5$. See Figure 2a.
- The quantile-based generalized logistic model is negatively skewed for values of $\delta > 0.5$ (See figure 2b) and positively skewed for $\delta < 0.5$ (See Figure 2a).

4.2 Generalized skew logistic model($GSLO_{QB}$)

A generalization of the quantile-based skew logistic distribution of [11] was studied further by [2]. He presents a generalization of this model and discusses its properties, as well as a method of estimation of its parameters.

Definition 4. Let X be a real-valued random variable. X is said to have a generalized quantile based skew logistic distribution denoted by $X \sim GSLO_{QB}(\alpha, \beta, \delta, \kappa)$, if it has the quantile function:

$$Q_{GSLO}(p) = \alpha + \beta\{(1-\delta)log(p^{\kappa}) - \delta log(1-p^{\kappa})\}, 0 \leq p \leq 1, \tag{5}$$

where α and $\beta(> 0)$ are the location and scale parameters respectively, and $\delta(0 \leq \delta \leq 1)$ and $\kappa(> 0)$ are shape the parameters.
Using the reflection rule for quantile functions by [3], the reflected $Q_{GSLO}(p)$ is equivalent to $-Q_{GSLO}(1-p)$.

Theorem 5. *Let $X \sim GSLO_{QB}(\alpha, \beta, \delta, \kappa)$, the quantile density and density quantile functions of the $GSLO_{QB}$ are given as*

$$q(p) = \beta\left(\frac{\kappa(1-\delta)}{p} + \frac{\delta\kappa p^{\kappa-1}}{1-p^{\kappa}}\right), 0 \leq p \leq 1, \tag{6}$$

and

$$f_p(p) = \frac{p\left(1-p^{\kappa}\right)}{\beta\kappa\left(1-\delta-p^{\kappa}+2\delta p\kappa\right)}, 0 \leq p \leq 1$$

respectively.

Proof. The same idea as in Theorem 2 is used. $q(p)$ is derived by differentiating 5 with respect to p. $f_p(p)$ by taking the reciprocal of $q(p)$. □

Theorem 6. *Suppose* $X \sim GSLO_{QB}(\alpha, \beta, \delta, \kappa)$ *then the L-location*(L_1), $L-$*scale*(L_2), $L-$skewness(τ_3) and L- *kurtosis*(τ_4) *of* $GSLO_{QB}$ *are respectively given as*

$$L_1 = \alpha + \beta\left(-\delta\psi(1) + (1-\delta)\,\psi\left(\frac{1}{\kappa}\right) - (1-2\delta)\,\psi\left(\frac{1}{\kappa}+1\right)\right),$$

$$L_2 = \frac{1}{2}\beta(1-2\delta)\kappa + \beta\delta(-\psi(\frac{1}{\kappa}) + \psi(\frac{2}{\kappa})),$$

$$\tau_3 = \frac{-\frac{1}{6}\beta\left(1-2\delta\right)\kappa + \beta\delta\left(\psi\left(\frac{1}{\kappa}\right) - 3\psi\left(\frac{2}{\kappa}\right) + 2\psi\left(\frac{3}{\kappa}\right)\right)}{\frac{1}{2}\beta(1-2\delta)\kappa + \beta\delta\left(-\psi\left(\frac{1}{\kappa}\right) + \psi\left(\frac{2}{\kappa}\right)\right)},$$

and

$$\tau_4 = \frac{\frac{1}{12}\beta\left(1-2\delta\right)\kappa + \beta\delta\left(-\psi\left(\frac{1}{\kappa}\right) + 6\psi\left(\frac{2}{\kappa}\right) - 10\psi\left(\frac{3}{\kappa}\right) + 5\psi\left(\frac{4}{\kappa}\right)\right)}{\frac{1}{2}\beta(1-2\delta)\kappa + \beta\delta\left(-\psi\left(\frac{1}{\kappa}\right) + \psi\left(\frac{2}{\kappa}\right)\right)},$$

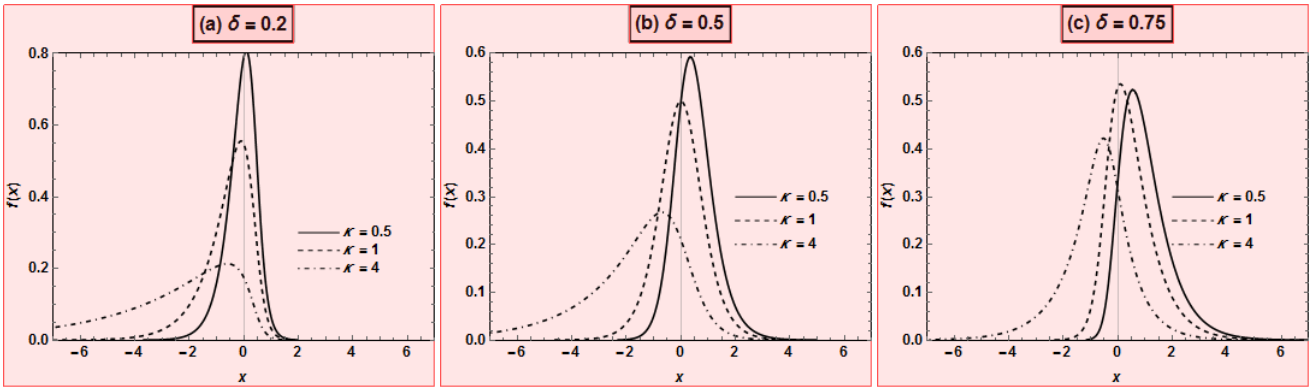
where $\psi(t) = \frac{d}{dt}\ln\Gamma(t) = \frac{\Gamma^{'}(t)}{\Gamma(t)}$ is the digamma function.

Proof. Refer to [2] for detailed proof of the results. □

[2] plotted two plots of the $L-$skewness and the $L-$kurtosis against κ for different fixed values of δ. From each plot he observed that the $L-$skewness increases beyond $\frac{1}{3}$ for large δ, which results in the upper bound for SLD_{QB} of [11]. For $\delta = 0$, $0 < \delta < 1$ and $\delta = 1$, τ_3 is constant at $-\frac{1}{3}$, τ_3 increases up to some point and then decreases eventually to $-\frac{1}{3}$ respectively as κ increases. The limit of τ_3 when $\kappa \to \infty$ is 1.
It follows that all together the $GSLO_{QB}$ and the reflected $GSLO_{QB}$ have all theoretically possible values of $\tau_{3,}$ in effect from -1 to 1.

The τ_4 plots pass through $\frac{1}{6}$ when $\kappa = 1$, which is in agreement with [11]. For $\delta = 0$, $0 < \delta < 1$ and $\delta = 1$, τ_4 is a constant at $\frac{1}{6}$, increases up to some point and then decreases to $\frac{1}{6}$ respectively as κ increases. The limit of τ_4 when $\kappa \to \infty$ is 1. For the summary of τ_3 and τ_4 refer to [2].

In Figure 3 the probability density curves for the $GSLO_{QB}$ are illustrated with selected values of δ and κ. Without loss of generality, assume $\alpha = 0$ and $\beta = 1$.



(a) (b) (c)

Figure 3: The probability density curves for the generalized skew logistic model($GSLO_{QB}$).

From Figure 3, some basic properties of the generalized skew logistic model $GSLO_{QB}$ are listed below:

- The distribution is symmetric around point zero when $\delta = 0.5$ and $\kappa = 1$.
- The distribution is positively skewed if $\delta > 0.5$ and negatively skewed if $\delta < 0.5$.
- When $\kappa = 1$, the generalized skew logistic model $GSLO_{QB}$ reduces to the quantile-based generalized logistic distribution(GLO_{vsk}) of [11].

4.3 The quantile-based generalized logistic distribution (GLO_{QB})

The new quantile-based generalized logistic distribution was proposed by [7]. It has two shape parameters. In this subsection, their findings are presented.

Definition 7. Let X be a real-valued random variable. X is said to have a quantile-based generalized logistic distribution, denoted by $X \sim GLO_{QB}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$,if it has a quantile function, defined as:

$$Q(p) = \lambda_1 + \frac{1}{\lambda_2}\left(\frac{1}{\lambda_3}\left(\left(\frac{p}{1-p}\right)^{\lambda_3} - 1\right) - \frac{1}{\lambda_4}\left(\left(\frac{1-p}{p}\right)^{\lambda_4} - 1\right)\right), 0 < p < 1, \tag{7}$$

where λ_1 and λ_2 are location and scale parameters respectively, whilst λ_3 and λ_4 are the shape parameters.

Theorem 8. *Suppose* $X \sim GLO_{QB}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$. *The quantile density and density quantile functions of the* GLO_{QB} *are*

$$q(p) = \frac{1}{\lambda_2 p(1-p)}\left(\left(\frac{p}{1-p}\right)^{\lambda_3} + \left(\frac{1-p}{p}\right)^{\lambda_4}\right), 0 < p < 1$$

and

$$f_p(p) = \lambda_2 p(1-p)\left(\left(\frac{p}{1-p}\right)^{\lambda_3+1} + \left(\frac{1-p}{p}\right)^{\lambda_4-1}\right)$$

respectively.

Proof. The same methodology as in Theorem 5 is applied, where $q(p)$ is derived by differentiating Eq 7 with respect to p and $f_p(p)$ by taking the reciprocal of $q(p)$. □

Theorem 9. *Let* $X \sim GLO_{QB}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$. *The L-Moments of the* GLO_{QB}, *L- location*(L_1), $L-$*scale*(L_2), $L-$skewness(τ_3) and *L- kurtosis*(τ_4) *respectively exists for* $-1 < \lambda_3 < 1$ *and* $-1 < \lambda_4 < 1$ *are given as*

$$L_1 = \lambda_1 + \frac{1}{\lambda_2}\left(\pi\left(Csc\left[\pi\lambda_3\right] - Csc\left[\pi\lambda_4\right]\right) - \frac{1}{\lambda_3} + \frac{1}{\lambda_4}\right),$$

$$L_2 = \frac{1}{\lambda_2}\left(\pi\left(Csc\left[\pi\lambda_3\right]\lambda_3 + Csc\left[\pi\lambda_4\right]\lambda_4\right)\right),$$

$$\tau_3 = \frac{Csc\left[\pi\lambda_3\right]\lambda_3^2 + Csc\left[\pi\lambda_4\right]\lambda_4^2}{Csc\left[\pi\lambda_3\right]\lambda_3 + Csc\left[\pi\lambda_4\right]\lambda_4},$$

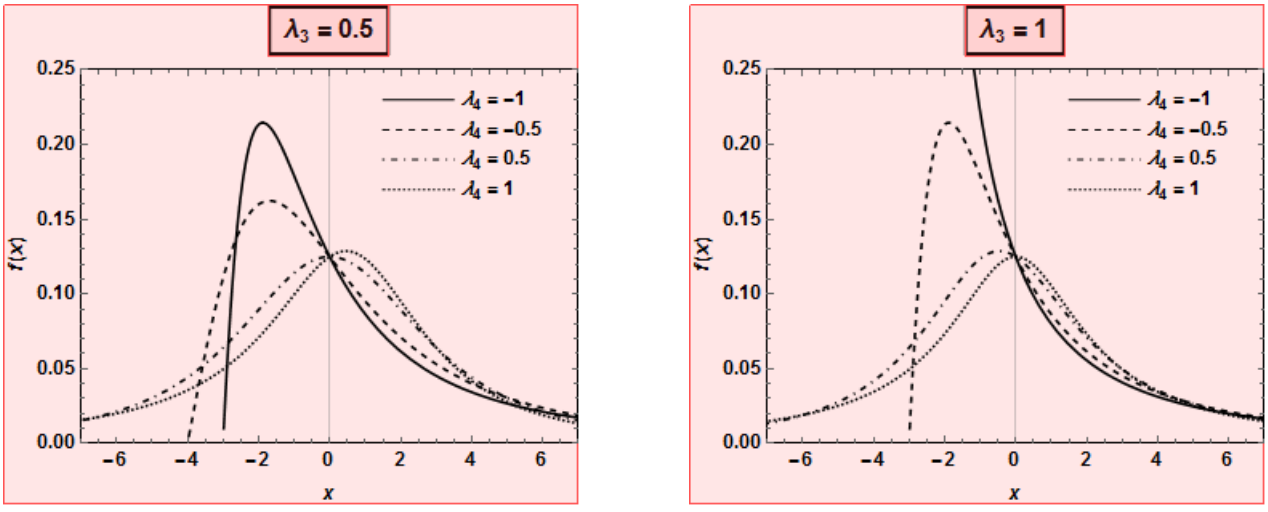
and

$$\tau_4 = \frac{Csc\left[\pi\lambda_3\right]\lambda_3\left(1+5\lambda_3^2\right) + Csc\left[\pi\lambda_4\right]\lambda_4\left(1+5\lambda_4^2\right)}{6\left(Csc\left[\pi\lambda_3\right]\lambda_3 + Csc\left[\pi\lambda_4\right]\lambda_4\right)},$$

where $Csc\left[\cdot\right]$ is the cosec function.

Proof. Refer to [7] for the detailed proof of the result. □

Figure 4 plots the probability density function of the GLO_{QB}, with selected values of the shape parameter λ_4 and λ_3. Without loss of generality, assume $\lambda_1 = 0$ and $\lambda_2 = 1$.



(a) (b)

Figure 4: The density curves for the quantile-based generalized logistic distribution (GLO_{QB}).

The following basic descriptions are observered from Figure 4:

- The GLO_{QB} is symmetric around point 0 when $\lambda_3 = \lambda_4$. See Figure 4a.

- When $\lambda_3 > \lambda_4$, the GLO_{QB} is negatively skewed (See Figure 4b) and positively skewed when $\lambda_3 < \lambda_4$ (See Figure 4a).
- The GLO_{QB} is $J-$shaped when $\lambda_3 \geq 1\,(\lambda_3 \leq -1)$ and $\lambda_4 \leq -1\,(\lambda_4 \geq 1)$. For all other combinations of the values of λ_3 and λ_4, the GLO_{QB} is unimodal. See Figure 4b.
- The probablity density function the GLO_{QB} reduces to logistic distribution when $\lambda_3 = \lambda_4 = 0$.
- If $X \sim GLO_{QB}\,(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$, then $X \sim GLO_{QB}\,(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = X \sim GLO_{QB}\,(\lambda_1, \lambda_2, -\lambda_3, -\lambda_4)$.

5 The $L-$moment ratio diagrams

The $L-$ moment ratio diagrams(plot of $L-$kurtosis against $L-$skewness) helps to decide on which probability distribution function to be chosen for regional frequency analysis. The $L-$moment ratio diagrams for the three generalization of the logistic distributions are illustrated in Figure 5-7. Figure **??** compares the flexibility of the different generalization. The dotted graph is the boundary for all distributions and the coloured region is the generalized logistic distribution. It is expected that all the three generalizations of the logistic distribution to plot within the dotted graph. The more area covered by the generalized logistic, the flexible it is with respect to distributional shape.

Figure 5 plots the $L-$ moment ratio diagram for the GLO_{vsk}. The shaded area(green) is the spaced covered by the GLO_{vsk}. The logistic distribution is obtained when $\lambda = 0$ and has the point $\left(0, \frac{1}{6}\right)$. From Figure 7, the symmetry is obtained when $\delta = \frac{1}{2}(\tau_3 = 0)$, positively skewed for $\delta < \frac{1}{2}\,(\tau_3 > 0)$ when $\lambda > 0$ and negatively skewed for $\delta > \frac{1}{2}\,(\tau_3 < 0)$ when $\lambda > 0$. It follows from Figure 5 that the minimum value for τ_4 for the GLO_{vsk} is the same for that of logistic distribution.

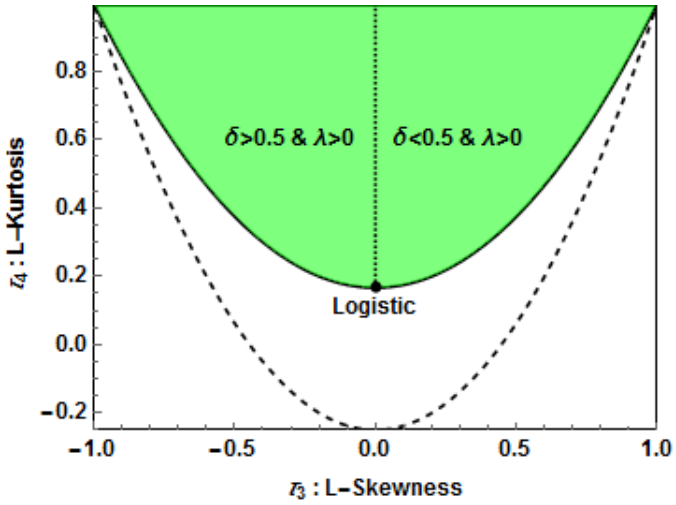


Figure 5: The $L-$ moment ratio diagram for the GLO_{vsk}.

The shaded area(Blue) in Figure 6 is the area covered by the generalized skew logistic($GSLO_{QB}$). The logistic distribution is obtained at the point $\left(0; \frac{1}{6}\right)$ and exponential distribution(E) at the point $\left(\frac{1}{3}; \frac{1}{6}\right)$ and the reflected exponential(RE) at the point $\left(-\frac{1}{3}; \frac{1}{6}\right)$.

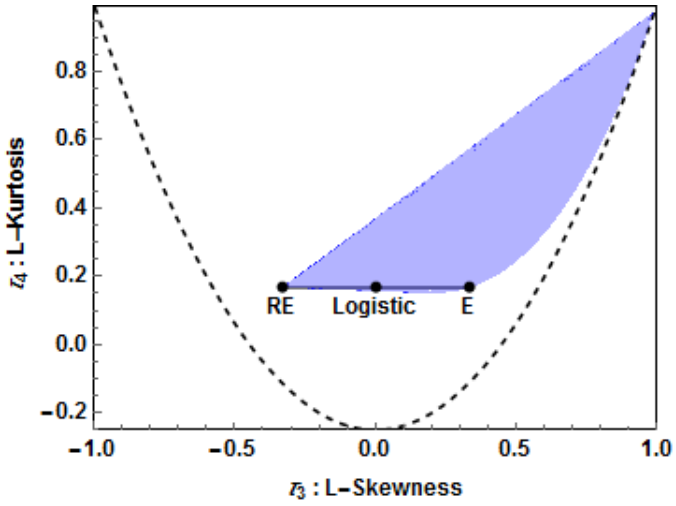


Figure 6: The $L-$ moment ratio diagram for the $GSLO_{QB}$.

Figure 7 shows the $L-$ moment ratio diagram for the GLO_{QB}. The shaded area(Orange) in Figure 7 is the space covered by the GLO_{QB}. The logistic distribution is obtained when $\lambda_3 = \lambda_4 = 0$ and has the point $\left(0, \frac{1}{6}\right)$. The symmetry is obtained when $\lambda_3 = \lambda_4 \, (\tau_3 = 0)$, positively skewed when $\lambda_3 > \lambda_4 \, (\tau_3 > 0)$ and negatively skewed when $\lambda_3 < \lambda_4 \, (\tau_3 > 0)$. The minimum value for τ_4 from the GLO_{QB} is $\frac{1}{6}$ which is the same as τ_4 for the logistic distribution.

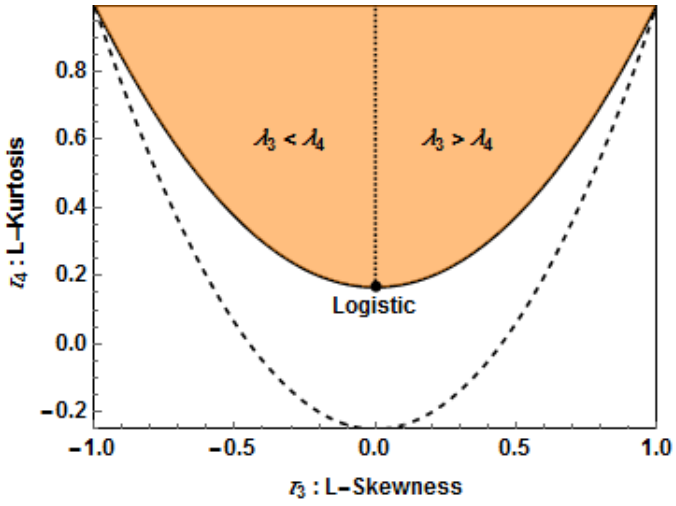
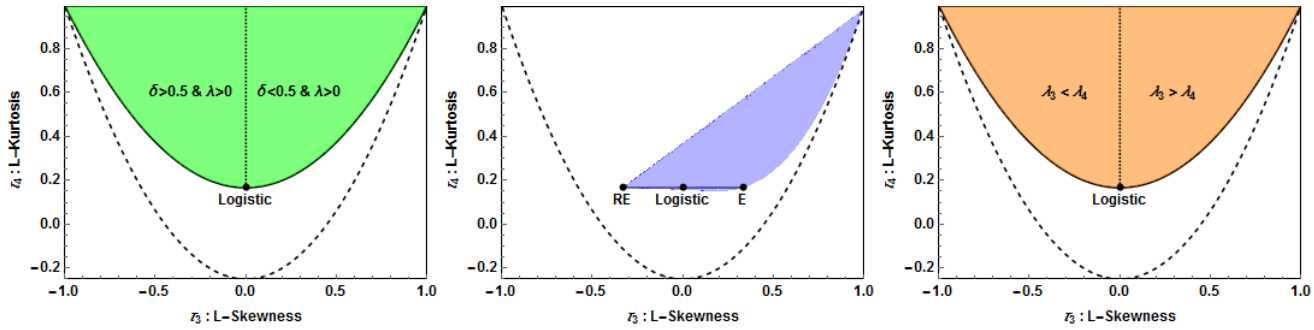


Figure 7: The $L-$ moment ratio diagram for the GLO_{QB}.

The GLO_{vsk} and GLO_{QB} covers equal amount of area see Figure 8. Note that the $GSLO_{QB}$ covers a small amount of area shaded by red. Hence the GLO_{vsk} and GLO_{QB} are more flexible as compared to $GSLO_{QB}$ with regard to distributional shape since large amount of area is covered.



(a) (b) (c)

Figure 8: Comparing the flexibility of the different generalizations via the $L-$moment ratio diagrams.

6 Conclusion

The paper discusses three different generalizations of the logistic distribution, each with two shape parameters. The three different types of generalizations are the quantile-based generalized logistic (GLO) distribution developed by [7], quantile-based GLO distribution possessing skewness-invariant measures of kurtosis, developed by [11], and the quantile-based skew GLO studied by [2]. The properties of the generalization of the logistic distribution can be easily obtained from the probability density curves.
The $L-$moment ratio diagram(a plot for the τ_4 against τ_3) for each generalization of the logistic is obtained. Distributions with two or more shape parameters cover a two-dimensional area on the $L-$moment ratio diagram. Clearly from Figure 8, the GLO_{QB} and the GLO_{QB} show more flexibility than the $GSLO_{QB}$ with regard to distributional shape since large area is covered..

References

[1] N Balakrishnan. *Handbook of the Logistic Distribution*. CRC Press, 1992.

[2] N Balakrishnan and H Y So. A generalization of quantile-based skew logistic distribution of van Staden and King. *Statistics and Probability Letters*, 107(2015):44–51, 2015.

[3] W G Gilchrist. *Statistical Modelling with Quantile Functions*. CRC Press, 2000.

[4] R K S Hankin and A Lee. A new family of non-negative distributions. *Australian & New Zealand Journal of Statistics*, 48(1):67–78, 2006.

[5] J R M Hosking. *The Theory of Probability Weighted Moments*. IBM Research Division, TJ Watson Research Center, 1986.

[6] 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.

[7] B V Omachar and P J van Staden. The quantile-based generalized logistic distribution. *The 60th World Statistics Congress of the International Statistical Institute (ISI2015)*, 2015.

[8] J S Ramberg, E J Dudewicz, P R Tadikamalla, and E F Mykytka. A probability distribution and its uses in fitting data. *Technometrics*, 21(2):201–214, 1979.

[9] J W Tukey. The practical relationship between the common transformations of percentages of counts and of amounts. *Statistical Techniques Research Group Technical Report*, 36, 1960.

[10] P J van Staden. *Modeling of generalized families of probability distribution in the quantile statistical universe*. PhD thesis, Department of Statistics, University of Pretoria, 2013.

[11] P J van Staden and King R A R. A quantile-based generalized logistic distribution possessing skewness-invariant measures of kurtosis. *The 60th World Statistics Congress of the International Statistical Institute (ISI2015)*, 2015.

[12] Wolfram. Mathematica 8.0. *Wolfram Research, Inc.*, 2010.

Appendix

All the figures in this paper was drawn in [12].

Appendix 1

$$g_1[x\_] := \frac{1}{2}\,\text{Erf}\left[\frac{x}{\sqrt{2}}\right]$$

$$g_2[x\_] := \left(1 + E^{\left(\frac{-\pi * x}{\sqrt{3}}\right)}\right)^{-1}$$

```
Plot[{g1[x], g2[x]}, {x, -7, 7}, PlotPoints → 50, MaxRecursion → 15,
 LabelStyle → Directive[Black, Medium, Bold], Frame → True,
 FrameLabel → {"x", "f(x)"}, FrameStyle → Thickness[Medium],
 FrameTicks → {Automatic, Automatic, None, None}, Axes → False,
 PlotStyle → {Black, {Black, Dashed}}, AspectRatio → 0.75, Epilog →
  {{Black, Line[{{-6, 0.8}, {-4, 0.8}}]},
   Text[Style["g1(x)", Black, Medium, Bold], {-3.5, 0.8}, {Left, Center}],
   {Black, Dashed, Line[{{-6, 0.6}, {-4, 0.6}}]},
   Text[Style["g2(x)", Black, Medium, Bold], {-3.5, 0.6}, {Left, Center}]}]
```

Appendix 2

```
Remove["Global`*"];
α = 0; β = 1; λ = 1; δ = {0, 0.2, 0.5, 0.9};
ParametricPlot[{{{α + β ((1 - δ[[1]])/λ ((p/(1 - p))^λ - 1) + δ[[1]]/λ (1 - ((1 - p)/p)^λ)),
    (-1 + p) p / ((p/(1-p))^λ β (-1 + δ[[1]]) - β (-1 + 1/p)^λ δ[[1]])},
   {α + β ((1 - δ[[2]])/λ ((p/(1 - p))^λ - 1)
      + δ[[2]]/λ (1 - ((1 - p)/p)^λ)), (-1 + p) p / ((p/(1-p))^λ β (-1 + δ[[2]]) - β (-1 + 1/p)^λ δ[[2]])},
  {α + β ((1 - δ[[3]])/λ ((p/(1 - p))^λ - 1) + δ[[3]]/λ (1 - ((1 - p)/p)^λ)),
    (-1 + p) p / ((p/(1-p))^λ β (-1 + δ[[3]]) - β (-1 + 1/p)^λ δ[[3]])},
   {α + β ((1 - δ[[4]])/λ ((p/(1 - p))^λ - 1) +
      δ[[4]]/λ (1 - ((1 - p)/p)^λ)), (-1 + p) p / ((p/(1-p))^λ β (-1 + δ[[4]]) - β (-1 + 1/p)^λ δ[[4]])}},
 {p, 0.0001, 0.9999}, PlotPoints → 50, MaxRecursion → 15, PlotRange →
  {{-7, 7}, {0, 0.35}}, PlotLabel → Style[Framed[" λ = 1 "], 16, Black],
 LabelStyle → Directive[Black, Medium, Bold], Frame → True, FrameLabel →
  {"x", "f(x)"}, FrameStyle → Thickness[Medium], FrameTicks → {Automatic,
    Automatic, None, None}, Axes → False, PlotStyle → {Black, {Black,
     Dashed}, {Black, DotDashed}, {Black, Dashing[{0.005, 0.005}]}},
 AspectRatio → 0.75, Epilog → {{Black, Line[{{2.5, 0.30}, {4, 0.3}}]},
   Text[Style["δ = 0", Black, Medium, Bold], {4.5, 0.30}, {Left, Center}], {Black,
    Dashed, Line[{{2.5, 0.27}, {4, 0.27}}]}, Text[Style["δ = 0.2", Black, Medium, Bold],
    {4.5, 0.27}, {Left, Center}], {Black, DotDashed, Line[{{2.5, 0.24}, {4, 0.24}}]},
   Text[Style["δ = 0.5", Black, Medium, Bold], {4.5, 0.24}, {Left, Center}],
   {Black, Dashing[{0.005, 0.005}], Line[{{2.5, 0.21}, {4, 0.21}}]},
   Text[Style["δ = 0.9", Black, Medium, Bold], {4.5, 0.21}, {Left, Center}]}]
```

Appendix 3

```
Remove["Global`*"];
α = 0; β = 1;
δ = 0.75;
κ = {0.5, 1, 4};
ParametricPlot[{{α + β ((1 - δ) Log[p^κ[[1]]] - Log[1 - p^κ[[1]]] δ),
    p (-1 + p^κ[[1]]) / (β (-1 + p^κ[[1]] (1 - 2 δ) + δ) κ[[1]])},
  {α + β ((1 - δ) Log[p^κ[[2]]] - Log[1 - p^κ[[2]]] δ), p (-1 + p^κ[[2]]) / (β (-1 + p^κ[[2]] (1 - 2 δ) + δ) κ[[2]])},
  {α + β ((1 - δ) Log[p^κ[[3]]] - Log[1 - p^κ[[3]]] δ), p (-1 + p^κ[[3]]) / (β (-1 + p^κ[[3]] (1 - 2 δ) + δ) κ[[3]])}},
 {p, 0.0001, 0.9999}, PlotPoints → 50, MaxRecursion → 15, PlotRange → {
   {-7, 7}, {0, 0.6}}, PlotLabel → Style[Framed["(c) δ = 0.75"], 16, Black],
 LabelStyle → Directive[Black, Medium, Bold], Frame → True, FrameLabel →
  {"x", "f(x)"}, FrameStyle → Thickness[Medium], FrameTicks → {Automatic,
   Automatic, None, None}, Axes → False, PlotStyle → {Black, {Black, Dashed},
   {Black, DotDashed}}, AspectRatio → 0.75, Epilog → {{Black, Line[{{2, 0.30},
      {3.5, 0.30}}]},
   Text[Style["κ = 0.5", Black, Medium, Bold], {3.65, 0.30}, {Left, Center}],
   {Black, Dashed, Line[{{2, 0.25}, {3.5, 0.25}}]}, Text[Style["κ = 1", Black,
     Medium, Bold], {3.65, 0.25}, {Left, Center}], {Black, DotDashed,
    Line[{{2, 0.20}, {3.5, 0.20}}]}, Text[Style["κ = 4", Black, Medium, Bold]
    , {3.65, 0.20}, {Left, Center}]}]
```

Appendix 4

```
Clear["Global`*"];
λ3 = 1;
λ4 = {-1, -0.5, 0.5, 1};
ParametricPlot[{{{1/λ3 ((p/(1 - p))^λ3 - 1) - 1/λ4[[1]] ((p/(1 - p))^λ4[[1]] - 1), p (1 - p)/((p/(1-p))^λ3 + ((1-p)/p)^λ4[[1]])},
  {1/λ3 ((p/(1 - p))^λ3 - 1) - 1/λ4[[2]] ((p/(1 - p))^λ4[[2]] - 1), p (1 - p)/((p/(1-p))^λ3 + ((1-p)/p)^λ4[[2]])},
  {1/λ3 ((p/(1 - p))^λ3 - 1) - 1/λ4[[3]] ((p/(1 - p))^λ4[[3]] - 1), p (1 - p)/((p/(1-p))^λ3 + ((1-p)/p)^λ4[[3]])},
  {1/λ3 ((p/(1 - p))^λ3 - 1) - 1/λ4[[4]] ((p/(1 - p))^λ4[[4]] - 1), p (1 - p)/((p/(1-p))^λ3 + ((1-p)/p)^λ4[[4]])}},
 {p, 0.0001, 0.9999}, PlotPoints → 50, MaxRecursion → 15, PlotRange →
  {{-7, 7}, {0, 0.25}}, PlotLabel → Style[Framed[" λ3 = 1 "], 16, Black],
 LabelStyle → Directive[Black, Medium, Bold], Frame → True,
 FrameLabel → {"x", "f(x)"}, FrameStyle → Thickness[Medium], FrameTicks →
  {Automatic, Automatic, None, None}, Axes → False, PlotStyle →
  {Black, {Black, Dashed}, {Black, DotDashed}, {Black, Dashing[{0.005, 0.005}]}},
 AspectRatio → 0.75, Epilog → {{Black, Line[{{1.5, 0.23}, {3.5, 0.23}}]},
   Text[Style["λ4 = -1", Black, Medium, Bold], {3.75, 0.23}, {Left, Center}],
   {Black, Dashed, Line[{{1.5, 0.21}, {3.5, 0.21}}]}, Text[Style["λ4 = -0.5",
     Black, Medium, Bold], {3.75, 0.21}, {Left, Center}], {Black, DotDashed,
    Line[{{1.5, 0.19}, {3.5, 0.19}}]}, Text[Style["λ4 = 0.5", Black, Medium, Bold],
    {3.75, 0.19}, {Left, Center}], {Black, Dashing[{0.005, 0.005}],
    Line[{{1.5, 0.17}, {3.5, 0.17}}]}, Text[Style["λ4 = 1", Black, Medium, Bold]
    , {3.75, 0.17}, {Left, Center}]}]
```

Appendix 5

```
Clear["Global`*"];

Plot[0.25 (5 τ3^2 - 1), {τ3, -1, 1}, PlotRange → {{-1, 1}, {-0.25, 0.99}},
 PlotStyle → {Black, Dashed}, LabelStyle → Directive[Black, Medium, Bold]
 , Frame → True, FrameStyle → Thickness[Small], FrameTicks → {{Automatic, None}
   , {Automatic, None}}, FrameLabel → {"τ3 : L-Skewness", "τ4 : L-Kurtosis"},
 Axes → False, AspectRatio → 0.75, Epilog → {First@ParametricPlot[{-λ, 1/6 (1 + 5 λ^2)},
     { λ, -1, 1}, PlotPoints → 50,
     MaxRecursion → 5, PlotRange → {{-1, 1}, {-0.25, 0.985}}, PlotStyle → Black],
   First@ParametricPlot[If[-1 < λ < 1 && 0 ≤ δ ≤ 1, {λ (1 - 2 δ), 1/6 (1 + 5 λ^2)},
      {Indeterminate, Indeterminate}], {λ, -0.999, 0.999}, {δ, 0, 1}, PlotStyle → Green,
     PlotRange → {{-1, 1}, {-0.25, 0.985}}, Mesh → False, BoundaryStyle → None],
   First@ParametricPlot[{0, λ}, {λ, 1/6, 1}, PlotPoints → 50, MaxRecursion → 5,
     PlotRange → {{-1, 1}, {-0.25, 0.985}}, PlotStyle → {Black, Dashing[{0.005, 0.005}]}],
   PointSize[0.02], Point[{0, 1/6}], Text[Style["Logistic", Black, Medium, Bold],
    {0, 1/6 - 0.06}], Text[Style["δ>0.5 & λ>0", Black, Medium, Bold], {-0.3, 0.6}],
   Text[Style["δ<0.5 & λ>0", Black, Medium, Bold], {0.3, 0.6}]}]
```

Appendix 6

```
Clear["Global`*"]; τre3 = -1/3; τre4 = 1/6;

τ322 = ((-1/6) (1 - 2 δ) κ + δ (PolyGamma[1/κ] - 3 PolyGamma[2/κ] + 2 PolyGamma[3/κ])) /
       ((1/2) (1 - 2 δ) κ + δ (- PolyGamma[1/κ] + PolyGamma[2/κ]));

τ422 = ((1/12) (1 - 2 δ) κ + δ (- PolyGamma[1/κ] + 6 PolyGamma[2/κ] - 10 PolyGamma[3/κ] + 5 PolyGamma[4/κ])) /
       ((1/2) (1 - 2 δ) κ + δ (- PolyGamma[1/κ] + PolyGamma[2/κ]));

Plot[0.25 (5 τ3^2 - 1), {τ3, -1, 1}, PlotRange → {{-1, 1}, {-0.25, 0.99}}, PlotStyle →
  {Black, Dashed}, LabelStyle → Directive[Black, Medium, Bold], Frame → True,
 FrameStyle → Thickness[Small], FrameTicks → {{Automatic, None}, {Automatic, None}},
 FrameLabel → {"τ3 : L-Skewness", "τ4 : L-Kurtosis"}, Axes → False, AspectRatio →
  0.75, Epilog → {{Black, Line[{{-1/3, 1/6}, {1/3, 1/6}}]},
   First@ParametricPlot[If[0 ≤ δ ≤ 1 && κ > 0, {τ322, τ422}, {Indeterminate,
       Indeterminate}], {δ, 0, 1}, {κ, 0.001, 1}, PlotStyle → Blue, PlotRange →
     {{-1, 1}, {-0.25, 0.985}}, Mesh → False, BoundaryStyle → None],
   First@ParametricPlot[If[0 ≤ δ ≤ 1 && κ > 0, {τ322, τ422}, {Indeterminate, Indeterminate}],
     {δ, 0, 1}, {κ, 1, 10}, PlotStyle → Blue, PlotRange → {{-1, 1}, {-0.25, 0.985}},
     Mesh → False, BoundaryStyle → None], First@ParametricPlot[If[0 ≤ δ ≤ 1 && κ > 0,
      {τ322, τ422}, {Indeterminate, Indeterminate}],
     {δ, 0, 1}, {κ, 10, 100}, PlotStyle → Blue, PlotRange → {{-1, 1}, {-0.25, 0.985}},
     Mesh → False, BoundaryStyle → None],
   First@ParametricPlot[If[0 ≤ δ ≤ 1 && κ > 0, {τ322, τ422}, {Indeterminate, Indeterminate}],
     {δ, 0, 1}, {κ, 100, 150}, PlotStyle → Blue, PlotRange → {{-1, 1}, {-0.25, 0.985}},
     Mesh → False, BoundaryStyle → None],
   First@ParametricPlot[If[0 ≤ δ ≤ 1 && κ > 0, {τ322, τ422}, {Indeterminate, Indeterminate}]
     , {δ, 0, 1}, {κ, 150, 200}, PlotStyle → Blue, PlotRange → {{-1, 1}, {-0.25, 0.985}},
     Mesh → False, BoundaryStyle → None],
   First@ParametricPlot[If[0 ≤ δ ≤ 1 && κ > 0, {τ322, τ422}, {Indeterminate, Indeterminate}]
     , {δ, 0, 1}, {κ, 200, 300}, PlotStyle → Blue, PlotRange → {{-1, 1}, {-0.25, 0.985}},
     Mesh → False, BoundaryStyle → None], PointSize[0.02], Point[{{1/3, 1/6}}],
```

Appendix 7

```
Clear["Global`*"];
Plot[0.25 (5 τ3^2 - 1), {τ3, -1, 1}, PlotRange → {{-1, 1}, {-0.25, 0.99}},
 PlotStyle → {Black, Dashed}, LabelStyle → Directive[Black, Medium, Bold],
 Frame → True, FrameStyle → Thickness[Small], FrameTicks → {{Automatic, None},
   {Automatic, None}}, FrameLabel → {"τ3 : L-Skewness", "τ4 : L-Kurtosis"},
 Axes → False, AspectRatio → 0.75, Epilog → {
   First@ParametricPlot[If[-1 < λ3 < 1 && -1 < λ4 < 1, {(Csc[π λ3] λ3^2 - Csc[π λ4] λ4^2)/(Csc[π λ3] λ3 + Csc[π λ4] λ4),
       (Csc[π λ3] λ3 (1 + 5 λ3^2) + Csc[π λ4] λ4 (1 + 5 λ4^2))/(6 (Csc[π λ3] λ3 + Csc[π λ4] λ4))}, {Indeterminate, Indeterminate}],
     {λ3, -0.999, 0.999}, {λ4, -0.999, 0.999}, PlotStyle → Orange, PlotRange → {{-1, 1},
       {-0.25, 0.985}}, Mesh → False, BoundaryStyle → None]}]
```

A comparison between the Bacherlier and Black-Scholes option pricing models

Thabo Victor Malope 10130889

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

02 November 2016

Abstract

There are many different models that can be used to obtain prices for options. In this report, we compare the option prices obtained using the Bacherlier and Black-Scholes models. Both of these models are based on Brownian motion. The assumptions that lead to the formulation of both models are also discussed.

We also study financial markets, together with the various instruments that are traded in these markets. Our main focus in this study is on options, particularly European call options. These are the most basic type of options that are available. An important concept that we also consider while calculating the option prices, is arbitrage. This concept is discussed in some detail in the report.

The Bacherlier and Black-Scholes models are fitted to a real world data set using two techniques. The first technique involves estimating the model parameters while the second technique involves a process called calibration which we describe in detail in this report. Calibration is necessary in order for us to make an objective comparison of the numerical results obtained. Our conclusions are based on the results obtained.

Declaration

I, *Thabo Victor Malope,* 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.

_____
Thabo Victor Malope

_____
Dr. Jaco Visagie

_____
Date: 02 November 2016

Acknowledgements

1) I would firstly like to thank God, the Almighty. It is indeed through Him that all things are possible, no matter how impossible they may seem at times. Philippians 4:13 "For I can do everything through Christ, who gives me strength."

2) Secondly, I would like to thank my family for always being supportive. It is truly a blessing to have such a loving family. Their unconditional love is very much appreciated.

3) Finally, I would like to thank my supervisor Dr. Jaco Visagie, for always being patient with me throughout this study.

Contents

1 Introduction **6**

2 Literature review **6**
2.1 Financial markets . . . 6
2.2 European options . . . 7
2.3 Arbitrage pricing . . . 8
2.4 The Bacherlier model . . . 9
2.5 The Black-Scholes model . . . 10

3 Practical implementation **10**
3.1 Confirmatory analysis . . . 11
3.1.1 Application of the Bacherlier model . . . 11
3.1.2 Application of the Black-Scholes model . . . 12
3.2 Observed financial data . . . 13
3.3 Distance measures . . . 14
3.4 Results obtained by model fitting . . . 14
3.4.1 Fitting the Bacherlier model . . . 15
3.4.2 Fitting the Black-Scholes model . . . 16
3.4.3 Comparison of the results . . . 17
3.5 Calibration results . . . 18
3.5.1 Calibration of the Bacherlier model . . . 18
3.5.2 Calibration of the Black-Scholes model . . . 19
3.5.3 Comparison of the results . . . 19

4 Conclusion **20**

Appendix **22**

List of Figures

1 Simulated stock price under the Bacherlier model . . . 11
2 Monte Carlo simulation of a call option under the Bacherlier model . . . 12
3 Simulated stock price under the Black-Scholes model . . . 12
4 Monte Carlo simulation of a call option under the Black-Scholes model . . . 13
5 S&P 500 index . . . 14
6 Observed (circles) and calculated (stars) option prices under the Bacherlier model . . . 16
7 Observed (circles) and calculated (stars) option prices under the Black-Scholes model . . . 17
8 *AAE* as a function of σ under the Bacherlier model . . . 18
9 *AAE* as a function of σ under the Black-Scholes model . . . 19

List of Tables

1 Estimation results . . . 18
2 Calibration results: Bacherlier model . . . 19
3 Calibration results: Black-Scholes model . . . 19
4 Calibration results: comparison . . . 20

1 Introduction

The New York, London, Tokyo and Johannesburg stock exchanges are very well known. Reports of the trading activities in these financial markets frequently make the front pages of newspapers. These reports are also often featured on television newscasts. In the sections to follow, we study financial markets, as well as some of the instruments that are traded in these markets.

Various financial instruments are traded in a financial market. Some of these instruments (called derivatives) derive their value from a more basic underlying asset, often a stock. Large financial markets contain numerous different types of derivatives, such as options. In this study, our primary focus will be on European options, particularly call options. These are the simplest types of options that are available.

There are many different models that can be used to obtain prices for options. In this report, we compare two option pricing models based on Brownian motion; the Bacherlier and Black-Scholes models. We also discuss the assumptions that lead to the formulation of each of these two models.

The Bacherlier model was the first attempt to model the workings of a financial market mathematically. It, however, had some problems. These include the model allowing for stock prices to be negative, which is in contradiction with economic theory. The Black-Scholes model followed the Bacherlier model 73 years later. The newer model is not afflicted by the same problems as its predecessor.

Although more advanced models are available, the Black-Scholes model remains the industry standard. Comparing the Bacherlier and Black-Scholes models is interesting because the models have a similar form but different motivations.

Both models will be fitted to observed stock price data and the corresponding option prices will be calculated using the parameter estimates obtained. Observed prices of options and those calculated under each model, are compared in terms of a distance measure. Furthermore, given the observed option prices, we will choose the parameters of the models considered in such a way that we minimize a distance measure between these and the option prices calculated under each model.

The rest of the report is structured as follows: Section 2 discusses the theory that formed a central part of this study. Section 3 discusses the fitting and the calibration of the models considered, to an observed data set. A comparison of the results is also given in Section 3. Our conclusions are given in Section 4.

2 Literature review

This section discusses financial markets as well as the assets found in these markets. Section 2.1 defines a financial market while Section 2.2 introduces European options. Section 2.3 introduces the concept of arbitrage pricing. We end this section by introducing, in Sections 2.4 and 2.5, the two models considered.

2.1 Financial markets

A financial market is a market in which people and institutions trade financial instruments. This market is sub-divided into different markets where all of these trading activities take place; [5]. The various types of financial instruments available in these markets are explained briefly below.

We firstly consider a stock. A stock is an equity investment that represents partial ownership in a listed company and entitles the holder to a part of that company's earnings and assets. Investors, be it individuals or institutions, can buy stocks of a particular company at the current market price. The price of the stock is a function of the performance of the issuing company. This price fluctuates over time. As a result, investors in stocks may realize financial gains, but they are also exposed to the risk of financial losses.

The financial market considered also include more secure investments for risk-adverse investors. Such investors can invest in bonds. Generally, the bond issuer is obliged to pay the bond holder regular interest payments during the term of the contract, as well as a specified amount at the end of the term of the contact (the date of maturity); [6]. The terms that are specified in the contract of the bond determines how and when interest is going to be paid. Investing in bonds generally carries no risk; a bond has a payoff that is known and certain. The only risk that can be incurred is that of the bankruptcy of the issuer. In this report, we assume that bonds are risk-free assets.

We denote by B_t, the value of a bond at time t. In this case, $B_t = B_0 e^{rt}$, where B_0 denotes the value of a bond at the beginning of a given period and r denotes the interest rate. We assume that the interest earned on bonds is constant and exponentially compounded throughout the report.

Different investors have different appetites for risk. Certain investors would prefer investing in assets that are riskier than stocks and bonds. This brings us to derivative instruments, which are used for many purposes. There are various types of derivative instruments. These include, amongst others, forwards, futures, swaps and options; [6]. Investors can use these instruments to reduce their exposure to risk, or to increase their exposure in the hope of increasing profits. European call options play a central role in this report. This derivative is discussed in more detail below.

We make some simplifying assumptions in the market considered. We allow the short-selling of assets (negative quantities of assets can be held). In this market, we assume that money can be borrowed and lent in arbitrary amounts at the same constant and exponentially compounded interest rate. We also allow fractional holdings of assets; meaning that numbers of stocks and bonds held are not restricted to whole numbers.

2.2 European options

There are various types of derivative instruments that investors have at their disposal. An option is defined as a contract which gives the holder the right, but not the obligation, to buy or sell an asset at a specified price within a specified period or at a certain point in time; [6]. This report mainly focuses on options, particularly European call options. These options are the simplest types of options available in the derivatives market; [4].

A European call option is defined as a contract which gives the holder the right, but not the obligation, to buy an asset at a specified price at a certain point in time. Let S_t denote the stock price at time t, and let T denote the date of the expiry of the option contract. Let K denote the price at which the stock can be bought; K is called the strike price of the option. Suppose that at time T, the stock price, S_T, is greater than the strike price, K. The holder of the call option can then buy the stock at the strike price K and immediately sell it at the current stock price S_T. This holder's profit will then be $S_T - K$. If the stock price is less than the strike price, the holder of the call option will not exercise the option. The payoff function of a call option is thus given by

$$P_{call} = \begin{cases} S_T - K & \text{if} \quad S_T > K, \\ 0 & \text{if} \quad S_T \leq K. \end{cases}$$

As was mentioned before, options can be used to reduce an investor's exposure to risk. Investors can however, deliberately expose themselves to risk in the hopes of increasing profits. This is precisely what speculators do. Suppose that a speculator is of the opinion that the price of a particular stock will increase. The speculator may then buy a call option. If indeed the stock price S_T is greater than the strike price K, the investor will exercise the option. The realized profit will be $S_T - K$ in this case. Speculating is, however, more risky than buying the stock. This is since an investor can lose the full amount that they have invested, if there is a large decrease in the stock price. On the other hand, if they buy the stock, a large decrease in the stock price will not reduce their capital to zero. However, since the price of the option is typically much less than that of the stock, the potential gains associated with buying options are much more than those associated with buying the stock.

2.3 Arbitrage pricing

An important concept in arbitrage pricing is that of a portfolio, which is a collection of financial assets. Portfolios are designed according to different investors' risk appetites and their investment objectives. The values of some derivative instruments can be replicated by constructing a portfolio consisting of stocks and bonds; [2]. This means that, at any given point in time, a self-financing portfolio's return characteristics exactly matches those of the derivative instrument; [6]. A portfolio is said to be self-financing if and only if the change in its value depends only on the change of the asset prices; [2].

The idea of a portfolio brings us to that of a portfolio strategy. A portfolio strategy details the amount of each asset held in our portfolio at any given instant; [2]. In a financial market consisting of a stock S_t and a bond B_t, a portfolio strategy can be represented by $\pi_t = (\phi_t, \psi_t)$, where ϕ_t and ψ_t represents the number of units of stock and bond held at time t. If the market contains additional assets (such as options), the definition of a portfolio can be extended accordingly.

In simple terms, arbitrage is a situation where a portfolio can be constructed at no initial cost, and this portfolio has a positive probability of yielding a positive payoff and a zero probability of yielding a negative payoff. Let π_t be a portfolio strategy at time t and let V_t be the value of this portfolio at time t; $V_t = \phi_t S_t + \psi_t B_t$. Then a portfolio strategy π_t constitutes an arbitrage opportunity on the time interval $[0, T]$ if:

a) π_t is self-financing.
b) The initial value of π_t is zero; $V_0^\pi = 0$.
c) $\mathbb{P}\left(V_T^\pi \geq 0\right) = 1$ and $\mathbb{P}\left(V_T^\pi > 0\right) > 0$; [3].

As an example, an arbitrage opportunity occurs when a portfolio that replicates the payoff of a given asset can be constructed for a price other than that of the asset. Investors will take advantage of this opportunity to realize a profit. They will simultaneously buy and sell this asset and its replicating portfolio, in arbitrary quantities in order to realize a profit from the difference in the prices. They will do so by buying the cheaper of the two and immediately thereafter selling it at the higher price. This will lead to a high demand for the asset with the lower price. Using the same reasoning, the supply of the asset with the higher price will be high. The laws of supply and demand will force the difference between the asset prices to shrink until the prices coincide exactly. These sets of trades carry no risk whatsoever. This means that investors can make arbitrary risk-free profits; [2]. It is thus important that the price of a given option must match that of its replicating portfolio at any given point in time if such a replicating portfolio exists; [6]. We assume that there are no arbitrage opportunities in our market. This assumption of no-arbitrage makes sense from an economic point of view, as argued above.

In order to calculate option prices, we need a suitable model for the price of a stock. Bacherlier proposed using Brownian motion as a model for the stock price. Black and Scholes, on the other hand, proposed using a different model. They proposed using a geometric Brownian motion. We assume that, under a given model, the evolution of the stock price is governed by some unknown probability measure that we denote by $\mathbb{P}$. This probability measure determines the behaviour of the stock price. $\mathbb{P}$ can be estimated from the observed historical stock prices.

The first fundamental theorem of asset pricing states that European call option prices can be calculated as expected values of the random variable $(S_T - K)^+$, taken with respect to $\mathbb{Q}$, where $\mathbb{Q}$ is a probability measure satisfying certain requirements; $\mathbb{Q}$ must be an equivalent martingale measure. We first define what we mean by equivalence. If A is an event in the sample space where $\mathbb{P}$ and $\mathbb{Q}$ are measures, then measures $\mathbb{P}$ and $\mathbb{Q}$ are said to be equivalent if and only if $\mathbb{P}(A) > 0 \iff \mathbb{Q}(A) > 0$; [2]. In other words, if A is possible under $\mathbb{P}$, then it is possible under $\mathbb{Q}$, and if A is impossible under $\mathbb{P}$, then it is impossible under $\mathbb{Q}$. In order to explain what is meant by a martingale measure, we define a martingale. A stochastic process $\{W_t\}_{t \geq 0}$ is said to be a martingale with respect to the measure $\mathbb{P}$ if and only if:

a) $\mathbb{E}_{\mathbb{P}}(|W_t|) < \infty$ for all t,
b) $\mathbb{E}_{\mathbb{P}}(W_t|\mathcal{F}_s) = W_s$ for all $s \leq t$,

where $\mathbb{E}_{\mathbb{P}}$ denotes the expected value taken with respect to the probability measure $\mathbb{P}$; [2]. A martingale measure is a probability measure which makes the expected future value of the discounted stock price, conditional on its present value and past history, equal to its present value; [2]. Under this newly introduced probability measure $\mathbb{Q}$, which is referred to as the risk-neutral measure, prices are obtained by calculating expected values of the random variable $(S_T - K)^+$ and discounting.

Brownian motions defined under the measures $\mathbb{P}$ and $\mathbb{Q}$ are equivalent if and only if their volatilities are equal; [2]. Below we consider the changes in probability measure from $\mathbb{P}$ to equivalent martingale measures under both of the models considered.

2.4 The Bacherlier model

Movements in the prices of stocks in financial markets are random and happen on a continual basis; [2]. This implies that we cannot model the price of a given stock in a deterministic way. The first attempt to model the behaviour of financial markets was made in Louis Bacherlier's thesis entitled "The Theory of Speculation", published in 1900; [1].

Bacherlier did not consider the effect of interest in his thesis, because at the time all payments happened on the same day. We introduce continuously compounded interest into his model. This is done to ensure that his model is realistic from a modern day "time-value of money" perspective, where the payments involved happen at two different points in time (a rand today will not have the same value a year from now). Bacherlier modelled the discounted value of a stock, $e^{-rt}S_t$, using Brownian motion. The model for the price of a stock under Bacherlier's model is given by the following:

$$S_t = e^{rt}(S_0 + \mu t + \sigma W_t),$$

where S_0 denotes the current price of a given stock, r denotes the risk-free interest rate, μ and σ denote the drift and volatility of the stock. W_t denotes a standard Brownian motion. Under this model, the returns of the stock price process follow a Brownian motion.

In order for the discounted stock to form a martingale under the Bacherlier model, the drift parameter μ must be set equal to zero. By setting the drift parameter to zero, we obtain a new probability measure $\mathbb{Q}$ under which arbitrage-free option prices are obtained. Under this new probability measure $\mathbb{Q}$, the model for the stock price is given by the following:

$$S_t = e^{rt}(S_0 + \sigma W_t).$$

This model for the price of a stock is used to derive a formula for the price of a European call option. The formula for the price of a European call option under the Bacherlier model is given by the following:

$$V(S_0, K, T) = \left(S_0 - e^{-rT}K\right)\Phi(-a) + \frac{\sigma\sqrt{T}}{\sqrt{2\pi}}e^{-\frac{1}{2}a^2}, \tag{1}$$

where $a = \frac{e^{-rT}K - S_0}{\sigma\sqrt{T}}$ and $\Phi(x) = P(Z \leq x)$ is the standard normal distribution function.

The main criticism against Bacherlier's model is that it allows the stock price to take negative values which would be inconsistent with economic theory; [2]. This is since Brownian motion W_t, used to model to the stock price, can assume any real value. Companies listed on a stock exchange cannot have negative stock prices. If the stock price of a particular company reaches zero, then that company will have to be liquidated.

A second criticism against the Bacherlier model is that the size of the price movements are not a function of the current stock price. By this we mean that the stock price is likely to move a fixed amount independent of the current stock price. As an example, consider a stock with a fixed volatility. Under the Bacherlier model, the probability that the stock price will increase by at least R10 in one day, is the same whether the current stock price is R10 or R1000.

Neither of the shortcomings mentioned above are present in the Black-Scholes model.

2.5 The Black-Scholes model

In their 1973 paper; [4], Black and Scholes propose using a geometric Brownian motion as a model for stock prices. This ensures that the modelled stock prices remain positive. Under this model, the distribution of stock price differences in any finite interval is log-normal. The log-returns are modelled using a Brownian motion. The volatility of the log-return on the stock is constant; [4]. This means that the variance of the log-returns in a given interval is proportional to the square root of the length of the interval.

Using the same notation as before, the formula for a stock price modelled by a geometric Brownian motion is given by the following:

$$S_t = S_0 exp\left(\sigma W_t + \mu t\right).$$

In order to ensure that the price process of the stock forms a martingale, the drift parameter μ is set equal to $\left(r - \frac{1}{2}\sigma^2\right)$. As was the case under the Bacherlier model, this change in probability measure ensures that we can calculate option prices that are arbitrage-free. The stock price model under the new probability measure is given by:

$$S_t = S_0 exp\left(\sigma W_t + \left(r - \frac{1}{2}\sigma^2\right)t\right).$$

This model leads to a formula for the price of a European call option; [5]. Using the same notation as before, the price V, of a European call option is given by the following:

$$V\left(S_0, K, T\right) = S_0 \Phi\left(\frac{log\frac{S_0}{K} + \left(r + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}\right) - Ke^{-rT}\Phi\left(\frac{log\frac{S_0}{K} + \left(r - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}\right), \tag{2}$$

where $\Phi\left(x\right) = P\left(Z \leq x\right)$ is the standard normal distribution function; [2].

In the next section, we show the numerical results pertaining to the models discussed above.

3 Practical implementation

In this section, we calculate the prices of European call options under both the Bacherlier and Black-Scholes models. We first calculate the price of a single option under each of the models, using the formulas provided in the previous sections. Thereafter we estimate these prices using Monte Carlo simulation.

We also fit both the Bacherlier and Black-Scholes option pricing models to a real world data set. The parameters of the two models are estimated based on the observed stock prices. The estimated parameters are then used to calculate option prices under the given model. Given the observed and calculated option prices, we can then calculate distance measures between these two sets of prices and compare the results.

The calibration of the two models to observed option prices, is also considered. The calibration of a given option pricing model to a set of observed option prices, entails choosing the parameters of the given model in such a way that the observed and calculated option prices correspond as closely as possible. We will illustrate this in detail in the calibration sections of both models.

The price of an option is calculated using two independent methods, in Section 3.1. In Section 3.2, we discuss the observed financial data that we use for the practical application. Section 3.3 discusses the different measures used to test the fit of the models. The results obtained by model fitting and those obtained by calibration, are given in Sections 3.4 and 3.5.

Various numerical and visual comparisons of the results will be featured in this section. All of the analysis and the simulations in this report was done using the statistical programming language R; [7].

3.1 Confirmatory analysis

In this section, the price of a single European call option under both the Bacherlier and Black-Scholes models, is calculated. This price is calculated using the formula given under each model and thereafter estimated using Monte Carlo simulation. Calculating the price using two independent methods, is used as a confirmatory analysis; i.e. this procedure is used to confirm that the algorithms used for the calculation of option prices contain no coding errors.

3.1.1 Application of the Bacherlier model

In this section, we calculate the price of a specific European call option using the formula given in (1) . The price of this call option is calculated by taking the following values: $S_0 = 100$, $T = 126$, $K = 100$, $\sigma = 1$, $r = \frac{0.10}{252}$ and $\mu = 0$. Figure 1 gives a possible path of the simulated stock price under this model. The code used in order to obtain this figure is given in Appendix A.

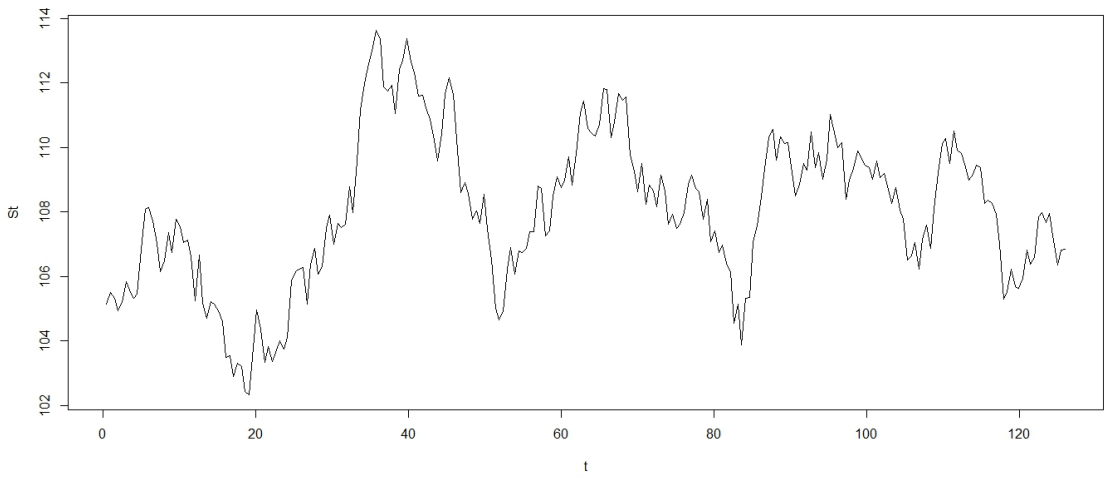


Figure 1: Simulated stock price under the Bacherlier model

The formula given in (1) gives us a value of $V = 7.3328$ for this option. This value is the exact price of this option. We now estimate this option price using Monte Carlo simulation. The price, V, is estimated by the following:

$$V = e^{-rT}\mathbb{E}_n\left[(S_T - K)^+\right],$$

where $\mathbb{E}_n$ is the empirical estimate of the expected value. A total of a 100000 Monte Carlo replications were used to estimate the option price. The value of the option was first estimated using 1000 simulations, then 2000, thereafter 3000 up until all 100000 simulations were used. The code for the simulation is included in Appendix B. Figure 2 below shows a graph of the estimated option price as a function of the total number of simulations used.

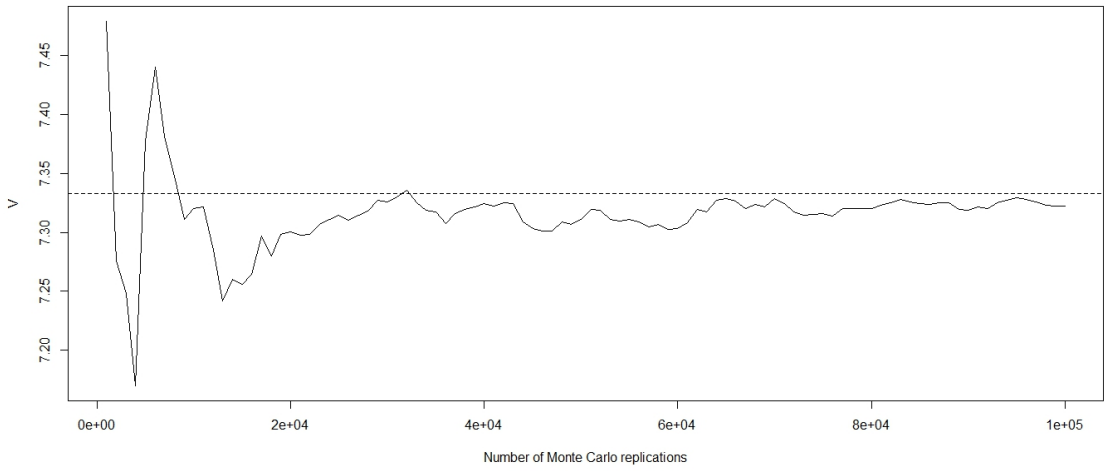


Figure 2: Monte Carlo simulation of a call option under the Bacherlier model

From Figure 2 above, we conclude that the estimated option price converges, as the number of simulations increase, to the same value given by the formula. We also conclude that the code used to calculate the option prices is correct.

3.1.2 Application of the Black-Scholes model

In this section, a specific European call option's price is calculated. This price is calculated using the Black-Scholes formula given in (2) and thereafter using Monte Carlo simulation.

We use the same values that we used when we calculated the price of this option under the Bacherlier model. The only difference here is that under the Black-Scholes model, we take $\sigma = 0.01$ and $u = \left(r - \frac{1}{2}\sigma^2\right)$. A simulated stock price path under the Black-Scholes model is given by Figure 3 below.

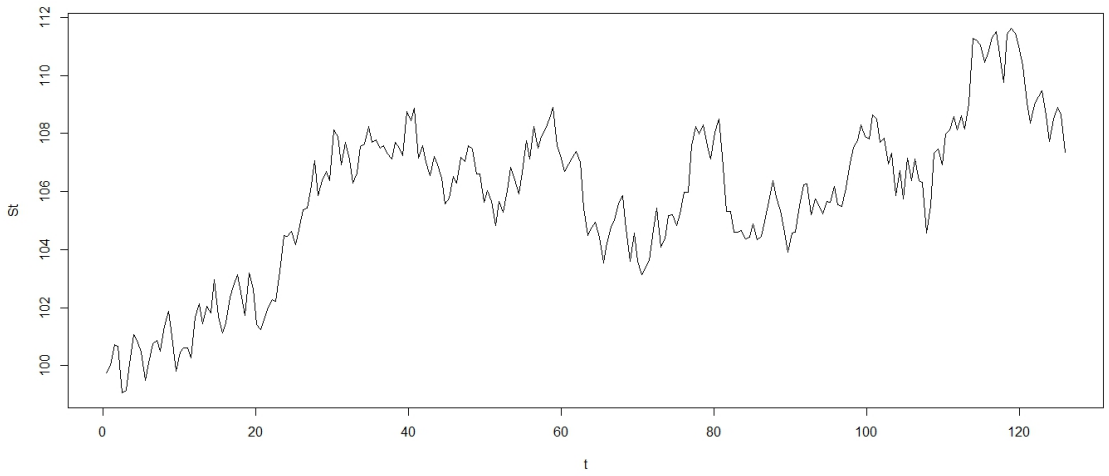


Figure 3: Simulated stock price under the Black-Scholes model

The code used for this simulation is given in Appendix C. The exact value of $V = 7.2308$ for this option, is calculated using (2); the Black-Scholes option pricing formula. Monte Carlo simulation is then used to estimate this price. The option price, V, is then estimated by the following:

$$V = \mathbb{E}_n \left[(S_T - K)^+ \right],$$

where $\mathbb{E}_n$ again denotes the empirical estimate of the expected value. This option price is first estimated using 1000 Monte Carlo replications, then 2000, thereafter 3000 up until a 100000 replications are used. Appendix D gives the code for this simulation. A graph of the estimated option price as a function of the number of simulations used, is given by Figure 4 below.

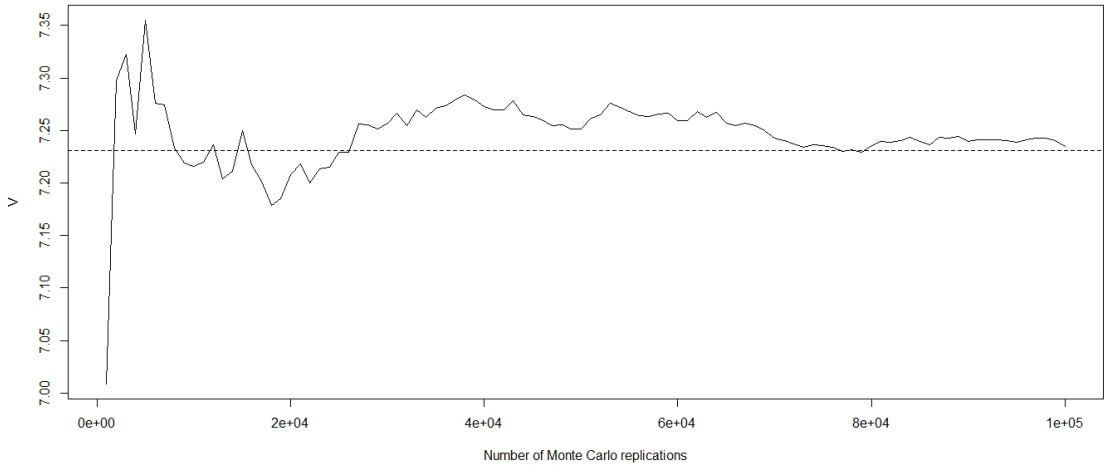


Figure 4: Monte Carlo simulation of a call option under the Black-Scholes model

From Figure 4 above, we see that the results obtained indicate a similar convergence to that found under the Bacherlier model. As a result, we are confident that the code that we used to calculate the option prices gives us prices that are accurate.

3.2 Observed financial data

For a practical application, we use the log-returns of the S&P 500 index from the 19th of April 2001 to the 18th of April 2002. On the last day of this period, the prices of 75 European call options were recorded on the S&P 500 index. The strike prices of these options vary from \$975 to \$1500 whereas the times to maturity vary from 21 days to 436 days. This data is given in Appendix E. The price path of the S&P 500 index over this period is given by Figure 5 below.

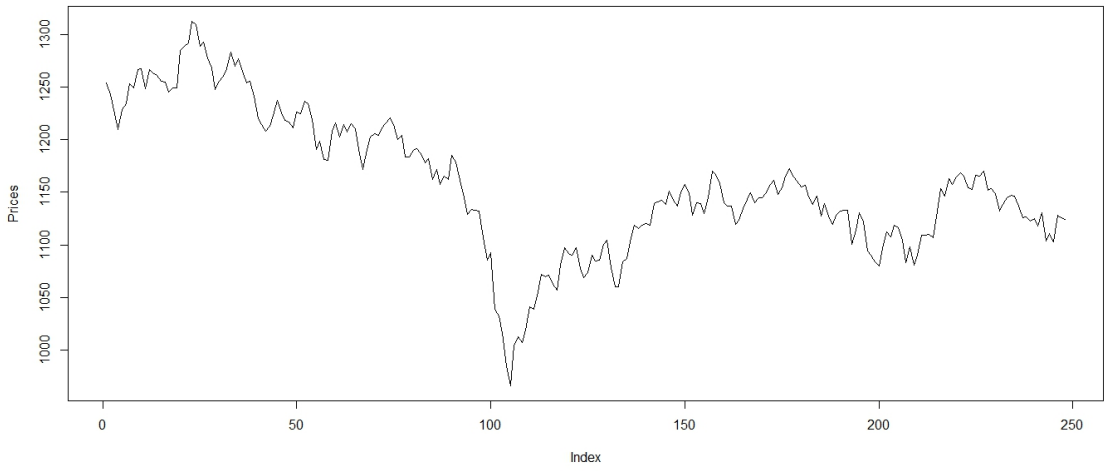


Figure 5: S&P 500 index

The code used to obtain the above figure is given in Appendix F.

3.3 Distance measures

There are various distance measures that can be used to measure the discrepancy between observed option prices and option prices calculated under some model. For the purposes of this application, we define three that are commonly used.

In a market containing n options, we denote the observed and the model prices of the i^{th} option, by π_i^O and π_i^E respectively. The model price is obtained under a given model. The average absolute error (AAE) is defined by

$$AAE = \frac{1}{n}\sum_{i=1}^{n} |\pi_i^O - \pi_i^E|.$$

The root mean square error ($RMSE$) is defined by

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(\pi_i^O - \pi_i^E\right)^2}.$$

The average relative error (ARE) is defined by

$$ARE = \frac{1}{n}\sum_{i=1}^{n} \frac{|\pi_i^O - \pi_i^E|}{\pi_i^O}.$$

In order to calibrate the models considered to observed option prices, we need to use one of these distance measures. For this purpose, we use the AAE. This measure is chosen because of its simple interpretation. The AAE is simply interpreted as the average amount that a given option pricing model misprices the options considered.

3.4 Results obtained by model fitting

In this section, we fit the Bacherlier and Black-Scholes models to observed stock prices. The results obtained when estimating the parameters of these two models are presented in this section. We use the estimated parameters to obtained prices for options. A comparison of the results will also be given in this section.

3.4.1 Fitting the Bacherlier model

We will now fit the Bacherlier model to the observed data set. When fitting the Bacherlier model to observed stock prices, we want to estimate the parameters of the model; μ and σ, under the probability measure $\mathbb{P}$. In order to estimate μ and σ, consider the following, under Bacherlier's model:

$$S_t = e^{rt}\left(S_0 + \mu t + \sigma W_t\right),$$

where W_t is a normal random variable with mean 0 and variance t. The data that we have available give the stock prices at the end of each day. Let S_t denote the stock price at the end of day t.

Then

$$\begin{aligned}
S_{t+1} &= e^{r(t+1)}\left(S_0 + \mu\left(t+1\right) + \sigma W_{t+1}\right) \\
&= e^{r(t+1)}\left(S_0 + \mu t + \sigma W_t + \mu + \sigma\left(W_{t+1} - W_t\right)\right) \\
&= e^{r(t+1)}\left(S_0 + \mu t + \sigma W_t\right) + e^{r(t+1)}\left(\mu + \sigma\left(W_{t+1} - W_t\right)\right) \\
&= S_t e^r + e^{r(t+1)}\left(\mu + \sigma\left(W_{t+1} - W_t\right)\right) \\
S_{t+1} &= S_t e^r + e^{r(t+1)}\left(\mu + \sigma Z\right) \text{ where } Z \sim N\left(0,1\right).
\end{aligned}$$

As a result

$$S_{t+1} - S_t e^r = e^{r(t+1)}\left(\mu + \sigma Z\right),$$

and

$$\frac{S_{t+1} - S_t e^r}{e^{r(t+1)}} = \mu + \sigma Z.$$

As a result, under the Bacherlier model,

$$e^{-rt}\left(e^{-r}S_{t+1} - S_t\right) \sim N\left(\mu, \sigma^2\right).$$

The expression, $e^{-rt}\left(e^{-r}S_{t+1} - S_t\right)$, represents the first differences of the given time series data, discounted accordingly under the Bacherlier model. These differences are the daily returns for the S&P 500 index and are independent normal random variables under the model used. We now estimate the parameters of the model based on these values. This entails estimating the mean and variance, which are denoted by:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} U_i \text{ and } \hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(U_i - \hat{\mu}\right)^2,$$

where $U_i = e^{-ri}\left(e^{-r}S_{i+1} - S_i\right)$ for $i = 1$ to n. $\hat{\mu}$ and $\hat{\sigma}^2$ are unbiased estimators for the theoretical population mean μ and variance σ^2.

Under the martingale measure $\mathbb{Q}$, we only estimate the volatility, $\hat{\sigma}$. $\hat{\mu}$ is set equal to zero in order for the price process to form a martingale under this model. The estimated volatility $\hat{\sigma}$ of the daily returns of the S&P 500 index over the time period considered is $\hat{\sigma} = 13.1591$. The code used in this estimation procedure is in Appendix G. We use this estimated volatility to calculate option prices, using (1). A graph showing the observed option prices and those calculated is given in Figure 6 below. The observed option prices are given by the circles whereas the prices under the model are given by the stars.

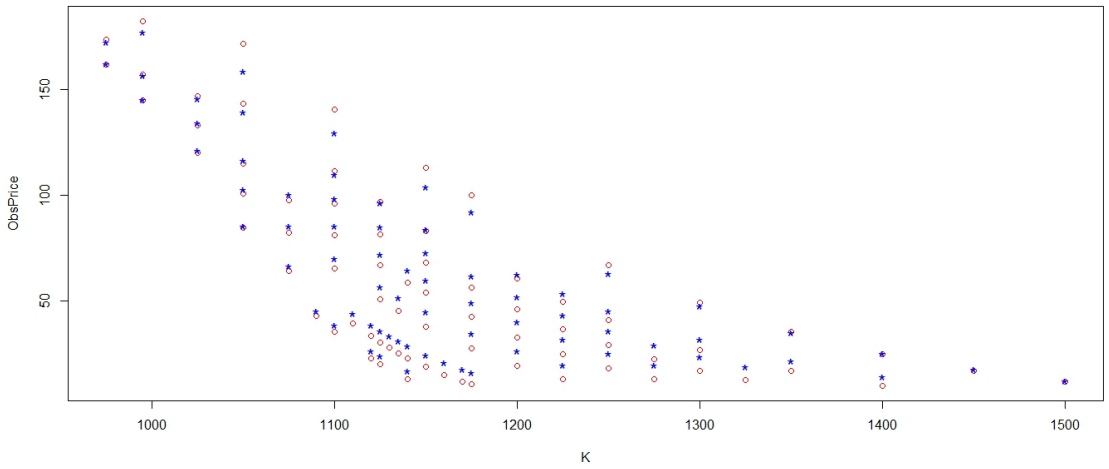


Figure 6: Observed (circles) and calculated (stars) option prices under the Bacherlier model

The prices obtained by the model correspond quite closely to the market prices. However, the model seems to overestimate the market prices in the majority of the cases.

3.4.2 Fitting the Black-Scholes model

We now fit the Black-Scholes model to a real world data set, as we did with the Bacherlier model. We use the same S&P 500 index data discussed in Section 3.2. The parameters of the model; μ and σ, are estimated. These parameters are those of the distribution of the stock prices observed under the probability measure $\mathbb{P}$.

In order to estimate the parameters μ and σ under the Black-Scholes model, we consider the following:

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

where W_t is a normal random variable with mean 0 and variance t. We are given stock prices at the end of each day, so S_t denotes the stock price at the end of the day t.

Then

$$\begin{aligned} S_{t+1} &= S_0 exp\left(\sigma W_{t+1} + \mu\left(t+1\right)\right) \\ &= S_0 exp\left(\sigma W_t + \mu t + \mu + \sigma\left(W_{t+1} - W_t\right)\right) \\ &= S_0 exp\left(\sigma W_t + \mu t\right) exp\left(\mu + \sigma\left(W_{t+1} - W_t\right)\right) \\ &= S_t exp\left(\mu + \sigma\left(W_{t+1} - W_t\right)\right) \\ S_{t+1} &= S_t exp\left(\mu + \sigma Z\right) \text{where } Z \sim N\left(0,1\right). \end{aligned}$$

As a result

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

and

$$ln\left(\frac{S_{t+1}}{S_t}\right) = \mu + \sigma Z.$$

Therefore

$$ln\left(\frac{S_{t+1}}{S_t}\right) \sim N\left(\mu, \sigma^2\right),$$

since if a random variable z is log-normally distributed, then the random variable $ln(z)$ is normally distributed; [6]. The log-returns $ln(S_{t+1}) - ln(S_t)$, of the given stock price data are obtained. These values are independent normally distributed random variables and they represent the daily returns on the S&P 500 index under this model. The mean and variance of these daily returns are denoted by:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} U_i \text{ and } \hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(U_i - \hat{\mu})^2,$$

where $U_i = ln(S_{i+1}) - ln(S_i)$ for $i = 1$ to n. These estimates are the unbiased estimates for the population parameters, μ and σ^2.

Under the Black-Scholes model, we obtain a new probability measure $\mathbb{Q}$ by setting $\mu = \left(r - \frac{1}{2}\sigma^2\right)$. μ and σ are then estimated and the estimates are found to be $\hat{\mu} = -0.0004$ and $\hat{\sigma} = 0.0116$. These estimates are used to calculate the prices of options under this model. This is done using (2). Figure 7 below shows the observed option prices and prices obtained under this model. The code used to calculate the option prices under this model, is given in Appendix H.

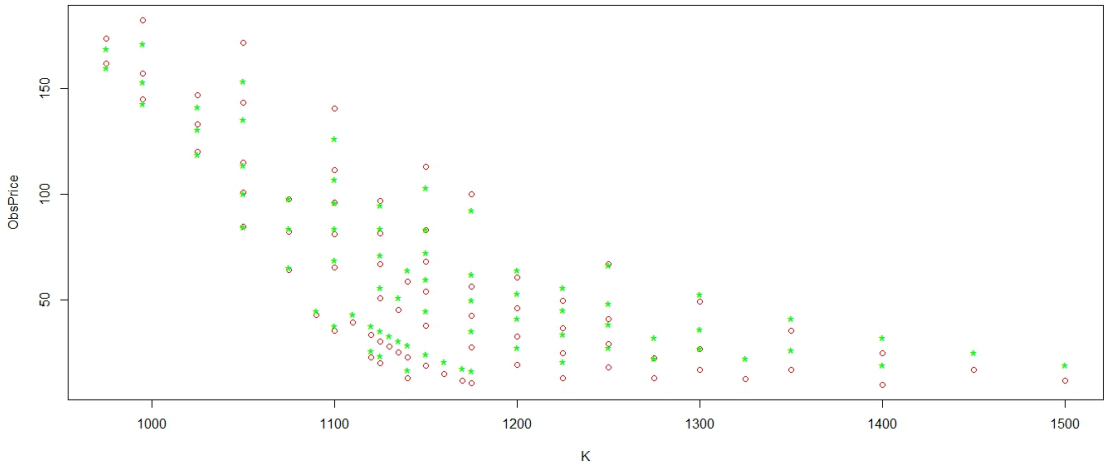


Figure 7: Observed (circles) and calculated (stars) option prices under the Black-Scholes model

We briefly comment on the above figure. The model and the market prices correspond very closely, as was the case with the Bacherlier model. However, in most of the cases, the model tends to overestimate the market prices.

3.4.3 Comparison of the results

We compare, in this section, the fit of both models to the observed prices of options. We use the distance measures defined in Section 3 for our comparison. The parameter estimates obtained under each model, together with the distance measures, are given in Table 1 below. The code used to obtain these estimates is included in Appendices G and H, right at the end of the programs.

	Bacherlier model	Black-Scholes model
μ	0	−0.00044
σ	13.1591	0.0116
AAE	4.4209	5.6359
$RMSE$	5.1345	6.5970
ARE	0.1467	0.1979

Table 1: Estimation results

Looking at the distance measures obtained, the Bacherlier model performs better than the Black-Scholes model in estimating the market prices. This means that the calculated option prices correspond more closely to the market prices, under the Bacherlier model than is the case under the Black-Scholes model.

3.5 Calibration results

In calibrating a given option pricing model to observed option prices in the market, we aim to minimize some distance measure between the observed prices and the prices calculated using the model. In order to compare the fit of various models to observed option prices, three distance measures were defined above; AAE, $RMSE$ and ARE. In this section, we consider the process of minimizing one of these three distance measures, namely the AAE.

Our aim is to minimize the AAE by adjusting the volatility of the stock price under each model. This is since the drift of the stock price process does not affect the option prices calculated. We adjust the volatility under each model by constructing on the real line, a grid of possible values for σ. That is for $k \in \mathbb{N}$, we have that $\hat{\sigma}_k \in [i, j]$ for some $i, j \in \mathbb{R}$, where $i < j$. We then calculate the option prices as well as the corresponding AAE, for each value of σ. The calibration process used entails choosing the value of σ that minimizes the AAE.

3.5.1 Calibration of the Bacherlier model

We consider the minimization of the AAE with respect to the volatility σ under this model. Figure 8 shows a graph of the AAE as a function of the σ values. Appendix I gives the code used to obtain the below figure.

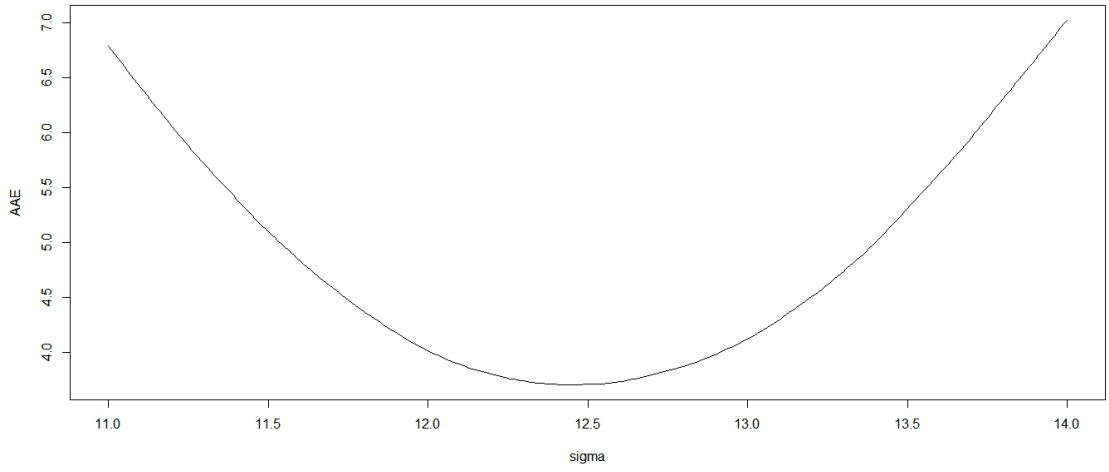


Figure 8: AAE as a function of σ under the Bacherlier model

The minimum value of the AAE is 3.7049 and is obtained when $\sigma = 12.44$. For this value of σ, the corresponding values of the $RMSE$ and the ARE are given in Table 2 below.

	$\hat{\sigma}_{initial} = 13.1591$	$\hat{\sigma}_{min-AAE} = 12.44$
AAE	4.4209	3.7049
$RMSE$	5.1345	5.0818
ARE	0.1467	0.0993

Table 2: Calibration results: Bacherlier model

3.5.2 Calibration of the Black-Scholes model

We now obtain the value of σ that minimizes the AAE under the Black-Scholes model. The AAE, as a function of the σ values, is given in Figure 9 below. The code used to obtain this graph is provided in Appendix J.

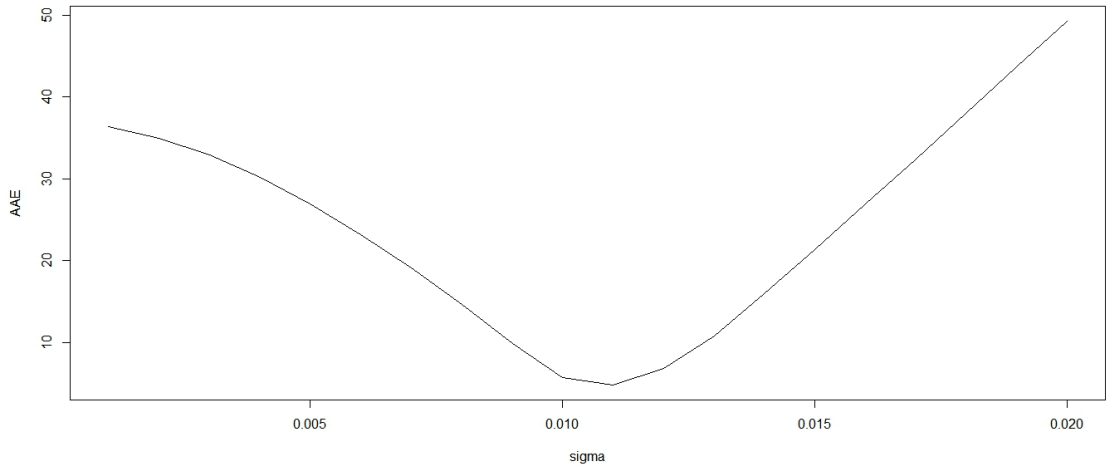


Figure 9: AAE as a function of σ under the Black-Scholes model

The minimum value of the AAE, obtained when $\sigma = 0.0108$, is 4.6947. Table 3 below gives the values of the $RMSE$ and the ARE for this value of σ.

	$\hat{\sigma}_{initial} = 0.0116$	$\hat{\sigma}_{min-AAE} = 0.0108$
AAE	5.6359	4.6947
$RMSE$	6.5970	6.5118
ARE	0.1979	0.1199

Table 3: Calibration results: Black-Scholes model

3.5.3 Comparison of the results

We consider, in this section, a comparison of the results of the calibrations of the two models considered. These results are given in Table 4 below.

	Bacherlier model	Black-Scholes model
$\hat{\sigma}_{initial}$	13.1591	0.0116
$\hat{\sigma}_{minimum}$	12.44	0.0108
AAE	3.7049	4.6947
$RMSE$	5.0818	6.5118
ARE	0.0993	0.1199

Table 4: Calibration results: comparison

Looking at the results, we see that with both models a smaller volatility value than the one initially estimated, is required in order for the observed and calculated option prices to correspond much more closely. Again, considering the calibration results, we see that the Bacherlier model still performs better than the Black-Scholes model in predicting the market prices. This is an interesting finding, as one would expect the Black-Scholes model to perform better.

4 Conclusion

The purpose of this research was to compare two option pricing models; namely the Bacherlier and Black-Scholes models. The report discussed the assumptions that led to the formulation of these two models. This was done keeping in mind that the development of the two models happened at different points in time. The stock price, under both models, is driven by Brownian motion. The Black-Scholes model is considered an improvement over the Bacherlier model, since the possibility of negative stock prices is removed under the Black-Scholes model.

In order to get a broader understanding of the comparison, we provided an overview of financial markets as well as the various financial instruments that are traded in these markets. Some of the financial instruments that were studied include stocks, bonds and options; specifically European call options. These options were given special attention as they formed the basis of our comparison. Arbitrage-free option pricing was also discussed in detail in this report.

The highlight of our research was the fitting and the calibration of both option pricing models to a real world data set. The fitting of the models entailed estimating the model parameters based on the observed stock prices over a given period of time. Calibration, on the other hand, entailed minimizing a distance measure (the average absolute error) between the option prices observed and those calculated using a model. It was rather interesting to see that in both cases of our comparison, the Bacherlier model performed better than the Black-Scholes model. This is despite the Black-Scholes model being an improvement over the Bacherlier model and the most used option pricing model in practice.

The two option pricing models considered in this report are both quite simple. The option prices under both models are determined by a single parameter. Further research might generalize this study by including more complex models.

References

[1] L. Bacherlier. The Theory of Speculation. *Annales Scientifiques de l'Ecole Normale Superieure*, 3:21–86, 1900.

[2] M. Baxter and A. Rennie. *Financial Calculus : An Introduction to Derivative Pricing.* Cambridge University Press, 1996.

[3] T. Bjork. An Introduction to Point Processes from a Martingale Point of View. Available at http://www.math.kth.se/matstat/fofu/PointProc3.pdf, 2011.

[4] F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy*, 3:637–654, 1973.

[5] V. Goodman and J. Stampfli. *The Mathematics of Finance : Modeling and Hegding.* American Mathematical Society, 2001.

[6] D.G. Luenberger. *Investment Science.* Oxford University Press, 2009.

[7] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2016.

Appendix

A) Simulation of the stock price path under the Bacherlier model:

```
S0 = 100
T = 126 # denotes the time to maturity. There are 252 business days in a year.
n = 250 # denotes the number of sub-intervals up to the time to maturity.
K = 100
sigma = 1
r = (0.1/252)

dt = (T/n)
t = seq(dt,T,by=dt)
x = c(sigma*sqrt(dt)*rnorm(n))
B = cumsum(x)
St = exp(r*T)*(S0+B)
plot(t,St,type="l")
```

B) Calculation of the price of a European call option using the Bacherlier option pricing model. This price is then estimated using Monte Carlo simulation:

```
S0 = 100
T = 126
n = 100
K = 100
sigma = 1
r = (0.1/252)

dt = (T/n)
t = seq(dt,T,by=dt)
MC = 100000
SP = matrix(0,MC,n)

# Bacherlier option pricing formula

EuropeanOption.dim <- function(S0,T,K,sigma,r) {

a1 = (exp(-r*T)*K-S0)/(sigma*sqrt(T))
a2 = exp((a1^2)*(-1/2))
a3 = 1/sqrt(2*pi)
V = (S0-exp(-r*T)*K)*pnorm(-a1)+(sigma*sqrt(T)*a2*a3)

return(V)
}

V = EuropeanOption.dim(S0,T,K,sigma,r)
V

# Monte Carlo simulation

for (j in 1:MC) {

x = c(sigma*sqrt(dt)*rnorm(n))
```

```
B = cumsum(x)
St = exp(r*T)*(S0+B)
SP[j,] = St

}

V = matrix(0,100,1)

for (j in 1:100) {

ST = SP[,n]
S = ST[1:(j*1000)]
V[j]= exp(-r*T)*mean((S-K)*(S>K))

}

MC =seq(1000, 100000, by=1000)
plot(MC, V, xlab="Number of simulation trials" , type="l")
abline(h = 7.332798, untf = FALSE, lty = 2)
```

C) Simulation of the stock price path under the Black-Scholes model:

```
S0 = 100
T = 126
n = 250
K = 100
sigma = 0.01
r = (0.1/252)
mu = (r-sigma*sigma/2)

dt = (T/n)
t = seq(dt,T,by=dt)
x = c(mu*dt+sigma*sqrt(dt)*rnorm(n))
B = cumsum(x)
St = S0*exp(B)
plot(t,St,type="l")
```

D) Calculation of the price of a European call option using the Black-Scholes option pricing model. This price is then estimated using Monte Carlo simulation:

```
S0 = 100
T = 126
n = 100
K = 100
sigma = 0.01
r = (0.1/252)
mu = (r-sigma*sigma/2)

dt = (T/n)
t = seq(dt,T,by=dt)
MC = 100000
SP = matrix(0,MC,n)
```

```
# Black-Scholes option pricing formula

EuropeanOption <- function(S0,T,K,sigma,r) {

b1 = log(S0/K)+(r+0.5*(sigma^2))*T
b2 = log(S0/K)+(r-0.5*(sigma^2))*T
c = exp(-r*T)
V = S0*pnorm(b1/(sigma*sqrt(T)))-K*c*pnorm(b2/(sigma*sqrt(T)))

return(V)
}

V = EuropeanOption(S0,T,K,sigma,r)
V

# Monte Carlo simulation

for (i in 1:MC) {

x = c(mu*dt+sigma*sqrt(dt)*rnorm(n))
B = cumsum(x)
St = S0*exp(B) SP[i,] = St

}

V = matrix(0,100,1)

for (j in 1:100) {

ST = SP[,n]
S = ST[1:(j*1000)]
V[j]= exp(-r*T)*mean((S-K)*(S>K))

}

MC =seq(1000, 100000, by=1000)
plot(MC, V, xlab="Number of simulation trials" , type="l")
abline(h = 7.230768, untf = FALSE, lty = 2)
```

E) The table below consists of the prices of the 75 European call options on the S&P 500 index:

Strike price K	$T = 21$	$T = 46$	$T = 111$	$T = 176$	$T = 241$	$T = 306$	$T = 436$
975			161.6	173.3			
995			144.8	157		182.1	
1025			120.1	133.1	146.5		
1050		84.5	100.7	114.8		143	171.4
1075		64.3	82.5	97.6			
1090	43.1						
1100	35.6		65.5	81.2	96.2	111.3	140.4
1110		39.5					
1120	22.9	33.5					
1125	20.2	30.7	51	66.9	81.7	97	
1130		28					
1135		25.6	45.5				
1140	13.3	23.2		58.9			
1150		19.1	38.1	53.9	68.3	83.3	112.8
1160		15.3					
1170		12.1					
1175		10.9	27.7	42.5	56.6		99.8
1200			19.6	33	46.1	60.9	
1225			13.2	24.9	36.9	49.8	
1250				18.3	29.3	41.2	66.9
1275				13.2	22.5		
1300					17.2	27.1	49.5
1325					12.8		
1350						17.1	35.7
1400						10.1	25.2
1450							17
1500							12.2

F) Code for the path of the S&P 500 index from the 19th of April 2001 to the 18th of April 2002.

```
Data <- read.table(file="SP500_logrets_19Apr2001_18Apr2002_yahoo2.csv")
Prices <- as.matrix(Data)
Prices <- Prices[N:1]
plot(Prices, type="l")
```

G) Observed and estimated prices of options under Bacherlier model:

```
Data <- read.table(file="SP500_logrets_19Apr2001_18Apr2002_yahoo2.csv")
Prices <- as.matrix(Data)
Prices <- Prices[N:1]

N = length(Prices)
Louis = matrix(0,N-1,1)
# r = 1.9% per year (interest)
# q = 1.2% per year (divident)
# nett interest rate = r - q
r = (0.007/252)

for (i in 1:N-1) {

t = -(N-i)
```

```
u = exp(-r*t)*(exp(-r)*Prices[i+1]-Prices[i])
Louis[i] = u

}

volatility = sd(Louis, na.rm=FALSE)
volatility

Data <- read.table(file="Calls.csv", sep=",")
Options <- as.matrix(Data)

S0 = Prices[N]
sigma = volatility

EuropeanOption <- function(S0,T,K,sigma,r) {

a1 = (exp(-r*T)*K-S0)/(sigma*sqrt(T))
a2 = exp((a1^2)*(-1/2))
a3 = 1/sqrt(2*pi)
V = (S0-exp(-r*T)*K)*pnorm(-a1)+(sigma*sqrt(T)*a2*a3)

return(V)
}

V = matrix(0,75,1)
K = Options[,1]
ObsPrice = Options[,2]
T = Options[,3]

for (j in 1:length(ObsPrice)) {

V[j] = EuropeanOption(S0,T[j],K[j],sigma,r)

}

V
# Graphing the observed and estimated prices of options

plot(K,ObsPrice,col = "red")
points(K,V,col = "blue",pch = "*",cex = 1.5)

AAE = mean(abs(ObsPrice-V))
AAE
RMSE = sqrt(mean((ObsPrice-V)^2))
RMSE
ARE = mean(abs(ObsPrice-V)/(ObsPrice))
ARE
```

H) Observed and estimated prices of options under the Black-Scholes model:

```
Data <- read.table(file="SP500_logrets_19Apr2001_18Apr2002_yahoo2.csv")
Prices <- as.matrix(Data)
```

```
Prices <- Prices[N:1]

N = length(Prices)
LogRtns = matrix(0,N-1,1)

for (i in 1:N-1) {

u = log(Prices[i+1]/Prices[i])
LogRtns[i] = u

}

drift = mean(LogRtns)
drift
volatility = sd(LogRtns, na.rm=FALSE)
volatility

Data <- read.table(file="Calls.csv", sep=",")
Options <- as.matrix(Data)

S0 = Prices[N]
sigma = volatility
# r = 1.9% per year (interest)
# q = 1.2% per year (divident)
# nett interest rate = r - q
r = (0.007/252)

EuropeanOption <- function(S0,T,K,sigma,r) {

b1 = log(S0/K)+(r+0.5*(sigma^2))*T
b2 = log(S0/K)+(r-0.5*(sigma^2))*T
c = exp(-r*T)
V = S0*pnorm(b1/(sigma*sqrt(T)))-K*c*pnorm(b2/(sigma*sqrt(T)))

return(V)
}

V = matrix(0,75,1)
K = Options[,1]
ObsPrice = Options[,2]
T = Options[,3]

for (j in 1:length(ObsPrice)) {

V[j] = EuropeanOption(S0,T[j],K[j],sigma,r)

}

V
# Graphing the observed and estimated prices of options

plot(K,ObsPrice,col = "red")
```

```
points(K,V,col = "green",pch = "*",cex = 1.5)

AAE = mean(abs(ObsPrice-V))
AAE
RMSE = sqrt(mean((ObsPrice-V)^2))
RMSE
ARE = mean(abs(ObsPrice-V)/(ObsPrice))
ARE
```

I) Minimizing the AAE with respect to sigma under the Bacherlier model:

```
Data <- read.table(file="SP500_logrets_19Apr2001_18Apr2002_yahoo2.csv")
Prices <- as.matrix(Data)

N = length(Prices)
Prices <- Prices[N:1]
Louis = matrix(0,N-1,1)
r = (0.007/252)

for (i in 1:N-1) {

t = -(N-i)
u = exp(-r*t)*(exp(-r)*Prices[i+1]-Prices[i])
Louis[i] = u

}

volatility = sd(Louis, na.rm=FALSE)

Data <- read.table(file="Calls.csv", sep=",")
Options <- as.matrix(Data)

S0 = Prices[N]
sigma = volatility

EuropeanOption <- function(S0,T,K,sigma,r) {

a1 = (exp(-r*T)*K-S0)/(sigma*sqrt(T))
a2 = exp((a1^2)*(-1/2))
a3 = 1/sqrt(2*pi)
V = (S0-exp(-r*T)*K)*pnorm(-a1)+(sigma*sqrt(T)*a2*a3)

return(V)
}

V = matrix(0,75,1)
K = Options[,1]
ObsPrice = Options[,2]
T = Options[,3]

sigma = seq(11,14,by=0.01)
xxx = matrix(0,length(ObsPrice),length(sigma))
```

```
AAE = matrix(0,length(sigma),1)

for (i in 1:length(sigma)) {

for (j in 1:length(ObsPrice)) {
V[j] = EuropeanOption(S0,T[j],K[j],sigma[i],r)
xxx[j,i] = V[j]

}}

for (j in 1:length(sigma)) {

AAE[j] = mean(abs(xxx[,j]-ObsPrice))

}

plot(sigma,AAE,type="l")
min(AAE)
sigma[which.min(AAE)]
```

J) Minimizing the AAE with respect to sigma under the Black-Scholes model:

```
Data <- read.table(file="SP500_logrets_19Apr2001_18Apr2002_yahoo2.csv")
Prices <- as.matrix(Data)

N = length(Prices)
LogRtns = matrix(0,N-1,1)
Prices <- Prices[N:1]

for (i in 1:N-1) {

u = log(Prices[i+1]/Prices[i])
LogRtns[i] = u

}

drift = mean(LogRtns)
volatility = sd(LogRtns, na.rm=FALSE)

Data <- read.table(file="Calls.csv", sep=",")
Options <- as.matrix(Data)

S0 = Prices[N]
r = (0.007/252)
sigma = volatility

EuropeanOption <- function(S0,T,K,sigma,r) {

b1 = log(S0/K)+(r+0.5*(sigma^2))*T
b2 = log(S0/K)+(r-0.5*(sigma^2))*T
c = exp(-r*T)
V = S0*pnorm(b1/(sigma*sqrt(T)))-K*c*pnorm(b2/(sigma*sqrt(T)))
```

```
return(V)
}

V = matrix(0,75,1)
K = Options[,1]
ObsPrice = Options[,2]
T = Options[,3]

sigma = seq(0.001,0.02,by=0.0001)
xxx = matrix(0,length(ObsPrice),length(sigma))
AAE = matrix(0,length(sigma),1)

for (i in 1:length(sigma)) {

for (j in 1:length(ObsPrice)) {
V[j] = EuropeanOption(S0,T[j],K[j],sigma[i],r)
xxx[j,i] = V[j]

}}

for (j in 1:length(sigma)) {

AAE[j] = mean(abs(xxx[,j]-ObsPrice))

}

plot(sigma,AAE,type="l")
min(AAE)
sigma[which.min(AAE)]
```

Evaluating the assessment practice of an extended programme module at the University of Pretoria

Sizwe Mbele 11230763

STK795 Research Report

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

Supervisor: Mrs AD Corbett

Department of Statistics, University of Pretoria

2 November 2016 (final)

Abstract

To many people, the sole purpose of studying is to pass without fully understanding the content of the work. For deep learners however, this is not the case. This paper will explore the role of three approaches associated with Higher Education students' learning, namely: surface, strategic and deep approaches to learning in the context of a first year extended progamme in foundation level statistics at the University of Pretoria. An evaluation of the extent to which it succeeds in shaping deep learning is conducted. The ICE model of assessment used in the programme is untangled in terms of its application in both formative and summative assessment activities. Results include descriptive statistics to identify relationships in how students perceive their own level of learning, to how they actually perform on higher level tasks. McNemar's test of symmetry in SAS (9.4) is used to reflect on the statistical significance of the results.

Declaration

I, *Sizwe Senzokuhle Mbele,* 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.

_____
Sizwe Senzokuhle Mbele

_____
Mrs. AD Corbett

_____
Date

Acknowledgements

The author would like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a postgraduate bursary as well as supervisor Mrs AD Corbett, Dr. Inger Fabris-Rotelli and the University of Pretoria for providing the resources to carry out the research.

Contents

1 Introduction **6**

2 Background Theory **7**
2.1 Literature Review . . . 7
2.2 Theoretical Framework . . . 8
2.3 The ICE Model . . . 10

3 Application **11**
3.1 Description . . . 11
3.2 Methodology . . . 14
3.3 Results . . . 15

4 Conclusion **21**

Appendix **23**

List of Figures

1 2013 ASSIST and Project work classification . . . 16
2 2014 ASSIST and Project work classification . . . 16
3 2015 ASSIST and Project work classification . . . 17
4 2013 Agreement plot between project work and 'After ASSIST' . . . 24
5 2014 Agreement plot between project work and 'After ASSIST' . . . 25
6 2015 Agreement plot between project work and 'After ASSIST' . . . 26

List of Tables

1 Approaches to learning . . . 8
2 Five Levels of increasing complexity described by SOLO . . . 9
3 WST 133/143 Assessment Framework with ICE . . . 12
4 Contingency table of 2013 . . . 15
5 Contingency table of 2014 . . . 15
6 Contingency table of 2015 . . . 15
7 Tests of symmetry - Project work . . . 18
8 Simple kappa coefficient statistics - Project work . . . 18
9 2013 Final marks contingency table . . . 19
10 2014 Final marks contingency table . . . 19
11 2015 Final marks contingency table . . . 19
12 Simple kappa coefficient statistics - Final module marks . . . 20
13 2013 ASSIST scores contingency table . . . 20
14 2014 ASSIST scores contingency . . . 20
15 2015 ASSIST scores contingency table . . . 20

1 Introduction

Educators all around the world are faced with challenges that include teaching students appropriate content and shaping them into people who can not only regurtitate the work they are taught but also to understand it. This is often more pertinent in the case of foundation level teaching, where teachers are tasked with introducing learners to more rigorous and demanding routines that present difficult concepts with the aim of understanding it and being able to apply it. Statistics is defined by the Oxford dictionary as the practice or science of collecting and analysing numerical and categorical data in large quantities, especially for the purpose of inferring proportions in a whole from those in a representative sample. It is clear from the intricacy of the definition alone that understanding statistical concepts is an important component that cannot be neglected.

The study of scientific methods such as statistics requires students to grasp and understand the theoretical content well before using it in application. The challenge then arises for educators to strike a balance on how to achieve this while learners continue to do well in school. Shaping students' approach to learning to a more critical way of thinking is known as deep learning. On the opposite end of that scale is what is known as surface learning and is characterised broadly by the tendency to stick closely to the minimum course requirements only. In between the two is what is known as strategic learning which is characterised by the motivation to perform well in an examination. Many current university students have been taught by teachers at school level only to achieve the necessary marks they need to move onto the next year of schooling. This has built a fear of failure in students and since the method works for them, they see no need to change it. They have almost been coached to be surface learners and bring that "skill" with them to university, which results in students not being interested in engaging with real understanding of concepts as well as they should. This further impacts their capacity to be employable graduates once they do graduate. In statistics, where understanding concepts builds from the bottom up; a deep approach of learning is vital for students to truly excel at the subject and become quality graduates that contribute credibly to the field. Simply studying to pass without understanding the connection between concepts and the extention of ideas to the real world of practising statistics will neither contribute towards becoming an informed user of statistics nor a successful statistician of profession.

Thorough research has been conducted concerning how then to reverse the type of coaching students tend to be receiving in school and develop a deep approach to learning necessary for university studies. A deep approach of learning can be encouraged by giving students the opportunity to discuss and debate their own understanding of the work they have been presented but also by teachers that design assessment which rewards making connections at a higher level [6]. As [8] indicates, teachers in higher education have considerable responsibility for the organisation of their courses in order to achieve this goal.

The aim of this paper is to evaluate the assessment practice for Mathematical Statistics (WST) 133/143 in the four year extended programme at the Mamelodi Campus of the University of Pretoria for the extend to which it succeeds in encouraging deep learning. It is envisioned to produce statistical results, using data collected over the past three years on both a reflective and an action level to determine whether it achieves its goal of shaping students to a deep level approach of learning. Results will consist of descriptive statistics processed in SAS (9.4) and Microsoft EXCEL as well as categorical analysis techniques to help answer this telling question.

In addition to the background theory following in section 2, the methodology for this research study is presented in section 3 along with the results, and concluding remarks follow in section 4.

2 Background Theory

2.1 Literature Review

According to [12], research has shown that there are considerable learning benefits when teachers introduce regular formative assessment into the classroom practice. It is now widely recognized that a key component in the learning process is assessment and feedback. An ICE Model of assessment introduced by Robin James Wilson and Sue Fostaty-Young [15] and further developed by more researchers will be used in this study to represent three hierarchical levels of learning growth [15]. The acronym for the ICE Model breaks down to I(deas), C(onnections) and E(xtensions) and is a framework that describes the progression of student learning from novice all the way through to expert. It is a useful tool to ensure that the intended outcomes are appropriate for the level of learning.

Students need to be actively involved in the learning process as discussed in Troskie-de Bruin and Otto's [10] research study surrounding the influence of assessment practices on students' learning approach. This notion alongside with Abedin et. al [3] is recognition that students' learning approach is important and the focus should be on the alignment of teaching, learning and assessment. The paper on using ICE to improve student learning by Sue Fostaty Young [14] describes the extension element of learning as the "AHA!" phase where students who reach the extensions phase are able to ask the "so what"? question, using previous knowledge and its application to internalise learning beyond the original learning context. A surface learner may attempt answering an extensive level question using a literal and not well thought out route.

Mantz Yorke [13] states that too often students' success is measured by indicators not pertaining to their conceptual and methodological strengths, like retention and completion rates amongst others. He goes on to say: "In the context of the first year experience, success is probably best viewed in terms of the extent to which the student (from a school or other background) is able to adjust to the demands posed by study in higher education" [13].

The ICE model helps curb this issue, providing a skeletal device to work with to quantify the level of student approach towards learning in a first year statistics module. This model is divided into 3 sub-sections: Ideas, Connections and Extensions, with the latter serving as a true measure of a deep level approach learner as it encourages them to create new learning from the old. Learning and understanding is tested extensively using this model of assessment as examination papers are divided into questions that purposely test each sub-section. Assessment in higher education shapes the experience of students and influences their behaviour more than the teaching itself [5]. The theoretical framework component in section 2.2 of this paper will elaborate more on taxonomies of assessment in higher education to aid in drawing possible conclusions about the extent to which the programme under evaluation succeeds in shaping students' approach to learning, considering both item construction and response level as indicators.

Research compiled by Troskie-de Bruin and Otto [10] is similar to that proposed in this paper. The authors' research question addresses the extent to which the assessment practices at a higher education institution influenced the quality of student learning. Quality of student learning is being evaluated using the same measuring instrument used in this research yet competing techniques are being used in terms of analysis. In reporting their main findings, the authors of [10] did a correlation study between examination results and learning approach and concluded that no significant correlation could be found between learning approach and academic performance on examination papers. In this paper, the assumption is that deep level learning approaches are encouraged by the mere implementation of the ICE assessment model. We are interested in the shift more specifically in terms of growing from surface level approach to learning at a deeper level, making tests of symmetry more appropriate.

2.2 Theoretical Framework

Learning approaches

Substantial attention has been given to the study of students' learning approaches at all levels and with varying disciplines. Entwistle describes the deep approach of learning as the intention to infer meaning to generate industrious learning processes that involve describing ideas, pattern and principle recognition on the one hand and making use of the proof and scrutinizing the logic of the argument on the other[6]. In contrast, the surface approach students only intend to just cope with the task without carrying out any deep processes of the material. The third approach is one associated with students who organise their learning with the objective of achieving high grades and is known as the strategic approach [6]. The following table of descriptors details the approaches further:

Surface	Strategic	Deep
• Memorizing • Skim Reading • Piecing bits of information together • Selecting/ Picking what is needed from the material • The intention is to reproduce/regurgitate	• Identify what is needed for the full mark and focus on that • May or may not involve understanding. • The intention is to excel and motivated by good marks	• Looking at whole texts to understand the author's intention • Selecting within the material • The intention is to understand

Table 1: Approaches to learning

Taxonomies of assessment

The idea of an 'ideal' assessment framework has received a significant amount of attention by researchers in the past. Bloom's taxonomy of educational objectives, which is recognised as the archetypal framework, is "a framework for classifying statements of what we expect or intend for students to learn as a result of instruction" [7]. The structure of the taxonomy (focusing on item construction) is divided into six main categories, namely: Remember, Understand, Apply, Analyse, Evaluate and Create; which depict a step by step level of growth in the way a student learns. The categories according to [7] explained are:

1) Recall facts and basic concepts.
2) Explain ideas and concepts.
3) Use information in new concepts.
4) Draw connections among ideas.
5) Justify a decision.
6) Produce original work

The Structural Observation of Learning Outcomes (SOLO) taxonomy developed by Biggs and Collis "provides a systematic way of describing how a learner's performance grows in complexity when mastering many tasks" [4]. The structure of this taxonomy focuses on response quality and describes a student's understanding of a subject or topic in five levels of increasing complexity:

1) Prestructural; describes a response of a student that has not understood the point and uses a generic and simple way to answer a question.
2) Unistructural; where basic connecions are made but the significance of the connections have not been fully grasped by the student.
3) Multistructural; multiple connections are made at this level. Aspects of the task are picked up and used but the whole significance is also missed.
4) Relational; students are now able to recognise the value of each of the parts in relational to the whole .
5) Extended Abstract Level; the student makes connections within and beyond the subject.

The following table showcases the type of verbs associated with each level of complexity:

Prestructural	Unistructural	Multistructural	Relational	Extended Abstract
• Name • Spot	• Define • Identify • Perform simple procedure	• Define • Describe • List • Combine	• Compare • Explain causes • Classify • Analyse • Apply • Formulate questions	• Evaluate • Theorise • Generalise • Predict • Create • Hypothesise • Reflect on

Table 2: Five Levels of increasing complexity described by SOLO

The quality of extrapolation and capacity of making connections desrcibed by the extended abstract level of the SOLO taxonomy is exactly what is required of statistics students. They are challenged daily by the complexity of the work they encounter and need to make connections from a basic level all the way through to a well developed one. The student that achieves this level of complexity in responding to higher level questions has understood the subject matter and has been able to make deductions beyond what they have been taught. This student possess a deep level of learning.

The ICE model, described in more detail in the next section, combines the focus on item construction and response quality into a compact and more portable assessment framework, developed by Robin James Wilson and Sue Fostaty-Young [14].

2.3 The ICE Model

The ICE (Ideas, Connections and Extensions) model of assessment is an intuitive and easy to understand framework used by lecturers and students alike to track the development of the student from surface learning to deep learning. It has been viewed as a useful framework in ensuring that the intended outcomes are appropriate for the level of learner [15]. This model has been implemented by the programme under evaluation for assessment design and classification of outcomes. The weight allocated to the various learning outcomes plays an important role (as indicated in Table 3). It is important to have a balanced mix of the categories 'ideas', 'connections' and 'extensions'. In attempting to achieve this goal, the ICE model is seen to combine both the item construction and response quality in one rather portable model, easy to understand for students.

Ideas are the building blocks of learning, the factual recall of information and grasping of basic concepts. It is only the information that students possess and is made of the facts that are contained in the assigned text. These make up the basis of new learning and form the first step of the model. An example of an idea level task in statistics is:

"Calculate the mean and the median from the given sample".

Knowing that the mean constitutes an average and median is the midpoint of a frequency distribution is an example of a basic concept that transpires from instruction. The question does not test the student's ability to make connections from previous ideas nor does it encourage new learning, it is merely a calculation.

Connections are the ties we make from our previous knowledge. It can be considered as knowledge built from ideas that have previously been stored. According to [14], it is recognizing general ideas across different contexts and being able to demonstrate relationships and connections among concepts. It can also be viewed as the ability to articulate relationships or articulating new learning to what is already there. An example of a connection level task is:

"Sketch a histogram in which the mean is greater than the median"

This requires students to build from several ideas that are assigned in the text and make the connections to answer the question effectively.

The third hierarchical level known as extensions occurs when students no longer have to refer to the rules for operations and make connections to several concepts even beyond the scope of the immediate topic being presented. It is associated with predicting future outcomes, justifying a position, proposing solutions and evaluating outcomes. New learning and ways of thinking is the ultimate outcome of this level and this skill is associated with learners who have a deep approach to learning. An example of an extensions level task is:

"Describe an example in which the median is a better measure than the mean"

The answer to a this question is not one that a student can simply learn from a textbook and will require them to know properties of both measures and beyond that, encourage creative thinking to provide a unique example that answers the question.

3 Application

Measuring the success of the aforementioned assessment practice is a task that involves collecting data related to students' performances over time, analysing the possible changes in the performance and how the intervention of the assessment framework lends a hand in explaining possible variation. The data used in this study was readily available and had been collected over time.

3.1 Description

Data and Instrumentation

Data about students who were registered for and enrolled in first year Mathematical Statistics (WST 133/143) from 2013 up to and including 2015 at the Mamelodi Campus of the University of Pretoria was made available for analysis. The performance data included grades and percentages of individual students across various assessment activities during the semester together with how that attributes to the final mark received at the end of the year. Measurements of how students performed over time, as well as the scores of a self-report questionnaire on Approaches on Study Skills Inventory for Students (ASSIST), developed by Marton and Saljo [3], is used for analysis purposes in this study.

The purpose of this research is to determine whether there are indications that the assessment model is shifting students' approach to learning from a surface learning approach to deeper learning approach, based on the design of activities to specifically encourage a deep approach to learning. To quantify the effect of the assessment being investigated, first year students for the years 2013, 2014 and 2015 were questioning regarding their attitude towards learning by completing the ASSIST questionnaire before exposure to the assessment model and its intention. A self report reflective score is calculated subsequently referred to as the "Before ASSIST score". The students were then introduced to the ICE model of assessment and after completing several activities throughout the first semester, were again asked to complete the ASSIST questionnaire reported as "After ASSIST score".

The ASSIST instrument has been used globally by credible institutions to assess students' learning in higher level education making it a useful tool for the purposes of this research. The three learning approaches of deep, strategic and surface are identified by the instrument which allows students to rank statements about their own level of learning using a 5-point scale from 1 ('definitely disagree') to 5 ('definitely agree'). The scores are then tallied and the approach is identified from that. Validation of the instrument is important for it to be taken as reliable and to improve the quality of the results it yields, hence extensive research surrounding the validity of the ASSIST instrument has raised questions specifically about whether or not it is logically and factually sound in concluding the information it does [9]. For the cohort of students that make up the bulk of the data for this investigation, the validity of the instrument has previously been researched by a former University of Pretoria student, Sharon Kgowedi, through a CFA factor analysis [1]. It was concluded that "ASSIST as an instrument yielded reliable and valid results for assessing the learning approaches of Statistics students on this particular programme".

Assessment Tasks

The ICE classification of tasks is useful for the purpose of gauging student learning approaches. A student who is capable of successfully completing higher level tasks is more likely to have used the deep learning approach. The teacher would thus be able to determine the learning approach of a student by looking at their relative score in each of the three ICE levels. The following table summarises the WST assessment framework and how the ICE classification is incorporated and merged with the assessment tasks.

Method	I(deas)	C(onnections)	E(xtensions)	Total weight
Semester tests	15	10	5	30
Class tests	5	5		10
Practical exam	5	5		10
Project work		5	15	20
Continuous assessment	20	5	5	30
SEMESTER MARK	50	25	25	100

Table 3: WST 133/143 Assessment Framework with ICE

The table above outlines the 2015 assessment weighting framework for the semester mark, with the only difference in preceeding years being the weight distribution of the Project Work (15 in 2013 and 2014). The exam mark is composed similarly with 50% (Idea level), 25% (Connection level) and 25% (Extension level). A final module mark is calculated with the semester mark carrying 60% of weight and exam mark carrying 40%. Three items are being analysed:

- Project work for WST 133
- Final module mark of WST 143
- Before ASSIST against After ASSIST

This is done assuming that an assessment practice which encourages a deep approach to learning would impact positively on the academic performance of students who are naturally inclined to follow this approach, but could also potentially change the learning approach to deeper levels of learning.

Project work

Project work as a specific component of the WST 133/143 assessment framework addresses the much desired need to encourage the adoption of a deep approach to learning. A deep learning approach is understood as learning identified by a motivation to seek meaning beyond assigned text, understanding of principles and assumptions thereof and being able to identify relationships between various ideas as previously mentioned. Project work demands these performance outcomes of students as it goes beyond just regurgitating the content in the assigned text. As an activity it invites almost only higher level responses like connections and extensions, thus making the performances of students in project work an appropriate and reliable measure to identify their learning approaches on an action level.

Classifiying Project work

The distinction in the three learning approaches can be identified from the way students performed in project work. The scales may differ for various tasks. For the purposes of this research, the classification of each approach was divided in the following manner for WST 133:

- Deep - 66% and higher
- Strategic - Between 55 % and 65%
- Surface - Below 55%

The mark a student obtained in the project work was then merged with their reflective 'After ASSIST' classification acquired after the ICE framework had been introduced to them and the project had been completed. This allowed students an opportunity to accurately score themselves after getting to understand what the model meant and required. The purpose of merging the information was to perform a cross tabulation analysis as it is a useful analytical tool and is a main-stay of this research study.

The project work activity was completed in groups of 4-5 which brought with it a limitation in terms of

allocating accurate marks for the work each individual did. Despite the strength of project work as the most useful activity to measure the shaping of deep learning, given the extent to which it tests at an extension level, a need to test the impact of the treatment in shaping individual students' approach, is perhaps important. The final module marks obtained in the final semester was therefore chosen as another response variable.

Final module mark

The final module marks incorporate a balanced mix of assessment levels across all activities and also account for individual performance of students. This lends a more precise measure of the impact of the assessment model as an intervention that encourages deep learning for all students. Taken here as an 'activity' on its own, it is worth mentioning that a student's total mark is comprised of several activities across each level of the ICE model throughout the semester as seen in table 3.

Classifying Final Module Mark

As depicted in table 3, in theory, a student can obtain a distinction at the end of the semester without having performed on an extension level. This however would imply that a student received full marks allocated to all the idea and connection level questions in each activity and none for extension level, which is highly unlikely in practice. Therefore, using the fact that the final mark comprises a balanced mix of levels, the classification will differ from a strict extension level task such as project work and is categorised in the following manner:

- Deep - 60% and higher
- Strategic - Between 50% and 59%
- Surface - Below 50%

Final module marks for WST 143 were merged with "After ASSIST" scores to perform a cross-tabulation analysis that will aid in concluding the impact of the assessment model as an intervention for individual students.

ASSIST Before and ASSIST After

Approaches to learning reflect the individual differences in strategies used to achieve a particular learning outcomes and are gauged by the ASSIST instrument on a reflective level for this cohort of students. The purpose of capturing students' own perceptions of their learning approaches before and after the intervention of the ICE model is to explore the way they have been shaped over and above improvements in their grades. This measure of their reflective approaches will give a clear and concise overview of the way students have received the model which plays a key a role in assessing its longevity and impact.

The trimmed 18 item (on a five point likert scale) ASSIST questionnaire was completed by students in the years 2013, 2014 and 2015. The items are specifically designed to measure the three learning approaches with questions arranged randomly from surface to deep level learning characteristics. There are six items of each approach and the highest of the tallied scores constitutes that student's learning approach.

3.2 Methodology

McNemar's Test and Bowker's Test of Symmetry

McNemar's standard test typically compares the proportions for two correlated variables that have been divided into sharply distinguished parts to check if the two classifications give similar results where the observed frequencies occuring outside the main diagonal of the matrix reports the non-compliance of the two measurements. For this study, the test had to be extended to a 3x3 square table based on the three different learning approaches. A test of agreement called the Bowker's test of symmentry is consequently required. To analyse the significance of the implementation of this assessment model as a treatment on shaping students' learning approaches, a SAS frequency procedure with an agree option is used to derive a contingency table along with statistics that aid with verifying the hypothesis of treatment significance.

The null hypothesis states that the cell proportions are symmentric. That is,

$$H_0 : O_{ij} = O_{ji}$$

for all pairs of table cells where O_{ij} and O_{ji} are the frequencies of the symmetrical pairs in our 3x3 design.

Bowker's Test of Symmetry

The basic assumptions of Bowker's test are that measurements are on a nominal scale and dependenct. Cochran's Q (test statistic) is computed for multiway tables and Bowker's test of symmetry is computed as,

$$Q_B = \sum\sum(n_{ij} - n_{ij})^2/(n_{ij} + n_{ij}) \text{ for i} < \text{j}.$$

The hypotheses for the purposes of this research are as follows:

H_0 = The number of students who classified themselves reflectively as deep learners (or surface learners) is exactly the same for each possible symmetric action level classification ,
H_a = The number of students who classified themselves reflectively as deep learners (or surface learners) is different for at least one of the possible symmetric action level classification.

Using the second round of ASSIST classifications against the project work students completed during the semester as well as the final module mark, a Bowker's test is used to provide critical statistics that will aid in exploring the effect of the assessment model. The most important statistic associated with the Bowker-McNemar test is Cohen's kappa coefficient which measures the inter-rater agreement for nominal variables.

Kappa Coefficient

Cohen's kappa coefficient is a statistic introduced by Jacob Cohen in 1960 which measures inter-rater agreement for categorical variables. It can be used to measure the level of agreement between two independent ratings, beyond chance. The simple kappa coeffiecient is calculated as,

$$\kappa = (P_0 - P_e)/(1 - P_e),$$

where $P_0 = \sum_i p_{ij}$ and $P_e = \sum_j p_{i\cdot}p_j$.
The kappa coefficient is 1 if there is absolute agreement of the raters. The strength of the agreement is measured by the magnitude of the statistic. Kappa is greater than zero when the observed agreement is larger than the chance agreement.

3.3 Results

Cross-tabulation analysis - Project work

To represent the relationship between how a student performed on action level and their own reflection of learning approach, a pivot table was compiled using Microsoft EXCEL. The following results display the cross-tables for 'After ASSIST' classification, represented as rows, and project work classification represented as columns for the years 2013, 2014 and 2015 for WST 133 (first semester project work).

COUNT	Project Work Classification			
ASSIST Results	Deep	Strategic	Surface	Grand Total
Deep	79	2	10	91
Strategic	78	9	10	97
Surface	32	4	4	40
Grand Total	189	15	24	228

Table 4: Contingency table of 2013

COUNT	Project Work Classification			
ASSIST Results	Deep	Strategic	Surface	Grand Total
Deep	89	20	15	124
Strategic	40	14	11	65
Surface	28	11	8	47
Grand Total	157	45	34	236

Table 5: Contingency table of 2014

COUNT	Project Work Classification			
ASSIST Results	Deep	Strategic	Surface	Grand Total
Deep	64	21	18	103
Strategic	45	23	14	82
Surface	23	9	10	42
Grand Total	132	53	42	227

Table 6: Contingency table of 2015

Tables 4, 5 and 6 show frequencies by cross-classifying the observations from 2013, 2014 and 2015, respectively. Over the course of the three years, the number of students performing on a surface level in the assessment task has been lower than that of the other two approaches depicted, despite an increasing percentage of surface level performances in each subsequent year (10.5% in 2013, 14.4% in 2014 and 18.5% in 2015). These slight increases, might however be due to the change in the assessment standards of project work since 2013. The ICE model was introduced to students in 2013 but only extensively implemented in 2015; over this three year period, refinement of assessment to align with the model is a possible reason why surface learning shows a slight increase as the years went on. Improvements are expected from 2015 onwards due to the fact that students and lecturers alike are now familiar with the model and the standards of assessment required.

Similar to research done by [10], this study is interested in the development of deeper thinking learners. The results in tables 4, 5 and 6 disinctly show a movement of students being shaped into deeper thinking learners. In 2013, 80% (32) of students who classified themselves as surface learners actually performed on a deep level in the project work activity, with only 10.98% (10) of those that classified themselves as deep

learners, actually performing on a surface level. Similar improvements are seen for the years 2014 and 2015, with 59.57% (28) and 54.76% (23) of learners who classified themselves as surface learners actually perfomed on a deep level, respectively.

The following bar graphs depict the desired movement from the reflective "After ASSIST" classification by students themselves and the performance classification in the formative group work project activity for all three years.

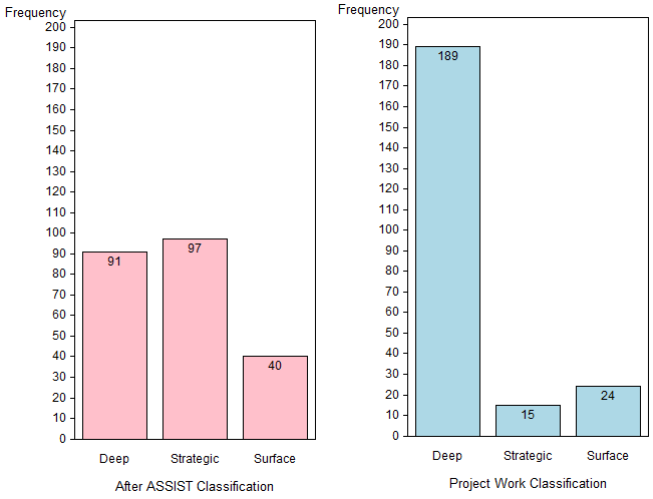


Figure 1: 2013 ASSIST and Project work classification

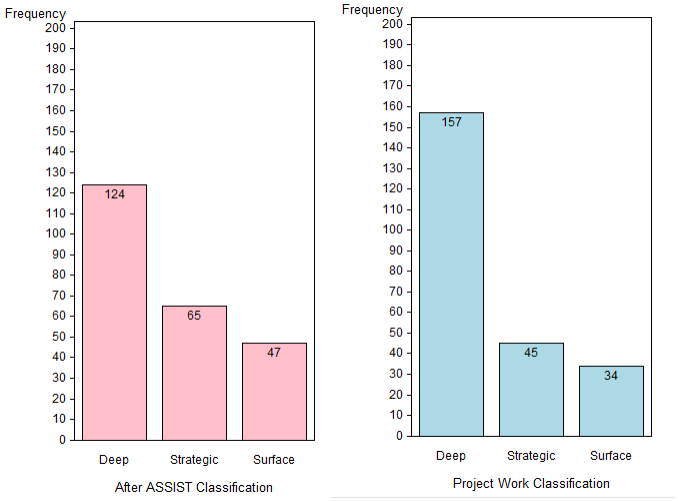


Figure 2: 2014 ASSIST and Project work classification

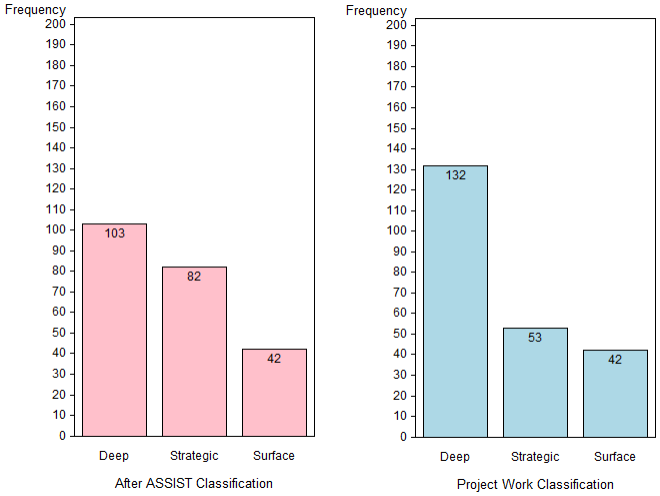


Figure 3: 2015 ASSIST and Project work classification

Statistical interpretation - Project work

The data from the students' project work and "After ASSIST" classifications were analysed in SAS using the 'Agree' option in the frequency procedure. The SAS (9.4) output in Table 7 documents the results of the Bowker- McNemar's test for the years 2013, 2014 and 2015.

Bowker-McNemar test	2013	2014	2015
Statistics	86.2952	10.5969	10.4240
DF	3	3	3
Pr > S	<.0001	0.0141	0.0153

Table 7: Tests of symmetry - Project work

Considering the p-values associated with all three years, the null hypothesis is rejected. There is sufficient evidence to suggest that the number of students who classified themselves under the three approaches is different for at least one of the possible symmetric level approaches. This supports the statistical significance of the descriptive findings in tables 4,5 and 6, indicating a shift to a deeper level of learning or that students changed their approach to learning. The implementation of the ICE model of assessment seems to be effective in shaping students' approach towards a deeper level of learning.

Table 8 shows the estimate of Cohen's kappa coefficient across all three years:

Cohen's kappa	2013	2014	2015
KAPPA	0.0421	0.0695	0.0727
ASE	0.0319	0.0471	0.0489
95% Lower Confidence Interval	-0.0205	-0.0228	-0.0231
95% Upper Confidence Interval	0.1046	0.1619	0.1684

Table 8: Simple kappa coefficient statistics - Project work

The magnitude of the kappa statistics across all three years indicate that the strength of agreement in the reflective and action level ratings are weak whilst the observed disagreement remains larger than the chance agreement. This result supports the significance of the treatment in its intention to encourage deeper levels of learning. There is is a definite indication that more learners classifying themselves as surface level, perform on a deep level in project work.

Cross-tabulation analysis - Final module mark

Unlike project work, the final module mark is a performance score that includes a substantial element of summative assessment activities on an individual level. As such, it could be considered gauging the possible impact of the intervention for individual students. The results presented in tables 9, 10 and 11 represent the relationship between individual students performance on an action level and the "After ASSIST" reflective classification.

COUNT	Final module mark classification			
After ASSIST	Deep	Strategic	Surface	Grand Total
Deep	43	25	23	91
Strategic	53	26	18	97
Surface	10	10	20	40
Grand Total	106	61	61	228

Table 9: 2013 Final marks contingency table

COUNT	Final module mark classification			
After ASSIST	Deep	Strategic	Surface	Grand Total
Deep	62	24	29	115
Strategic	44	6	13	63
Surface	16	7	19	42
Grand Total	122	37	61	220

Table 10: 2014 Final marks contingency table

COUNT	Final module mark classification			
After ASSIST	Deep	Strategic	Surface	Grand Total
Deep	67	24	12	103
Strategic	65	11	6	82
Surface	28	9	5	42
Grand Total	160	44	23	227

Table 11: 2015 Final marks contingency table

Final module mark results, much like the project work activity, show an increase in the number of deep level learners across the three years from those that classified themselves as deep using the ASSIST instrument. The number of students who classified themselves as surface learners and actually performed on a surface level also increased in the years 2013 (52.5%) and 2014 (45.24%) when the ICE model had not been entirely implemented in the extended programme module. When the intervention was formally implemented in 2015, the number of surface level learners decreased by 45% from their reflective classification in favour of deep learning. Table 11 also shows the highest increase in the number of deep level learners; the 55% increase from 103 to 160 serves as a strong indication that the assessment model in this module is indeed shaping students' learning approaches, especially since the 2015 results are the latest that have been collected and analysed.

Statistical interpretation - Final module marks

Table 12, computed similarly to table 8, detail the kappa statistics for the final module marks of WST 143.

Cohen's kappa	2013	2014	2015
KAPPA	0.0674	0.0047	-0.0726
ASE	0.0493	0.0487	0.0445
95% Lower confidence interval	-0.0292	-0.0907	-0.1599
95% Upper confidence interval	0.1639	0.1001	0.0146

Table 12: Simple kappa coefficient statistics - Final module marks

The slight agreement given by the kappa statistics in the years 2013 and 2014, similar to the analysis of project work, support the significance of the ICE model as a treatment in its intention to encourage deep levels of learning. There is a clear indication that the treatment has achieved its goal in shifting of surface level learners performing on a deep level. The negative kappa statistic (-0.0726) indicates a less than chance agreement and is not interpreted differently from any other kappa statistics less than 0.20. The magnitude of the statistic supports the significance of the treatment in shaping students' approach to learning.

Cross-tabulation analysis - ASSIST scores

As a means of a truer reflection on how students perceive a change in their own approach to learning, a cross-tabulation of the before and after ASSIST scores was compiled. From a research perspective, the importance of these results attempts to indicate how students' attitude toward learning on a reflection basis has shifted, if at all. The results for the years 2013, 2014 and 2015 are presented in the following tables:

COUNT	After ASSIST			
Before ASSIST	Deep	Strategic	Surface	Grand Total
Deep	25	10	6	41
Strategic	19	40	8	67
Surface	2	2	7	11
Grand Total	46	52	21	119

Table 13: 2013 ASSIST scores contingency table

COUNT	After ASSIST			
Before ASSIST	Deep	Strategic	Surface	Grand Total
Deep	52	14	16	82
Strategic	50	43	15	108
Surface	9	5	12	26
Grand Total	111	62	43	216

Table 14: 2014 ASSIST scores contingency

COUNT	After ASSIST			
Before ASSIST	Deep	Strategic	Surface	Grand Total
Deep	59	13	15	87
Strategic	32	61	13	106
Surface	5	3	9	17
Grand Total	96	77	37	210

Table 15: 2015 ASSIST scores contingency table

Across the three years, theres is an indication that the ICE model of assessment does indeed succeed in encouraging a deep approach to learning. The consistent rise in the number of 'deep' approach classifications from before to after ASSIST measurement tables in 13, 14 and 15 confirm this observation.

A result seen across all three years is the drop in the number of learners who classified themselves as strategic and spreading themselves between deep and surface learning. This result can be alluded to the fact that at the beginning of every year, most students are motivated by getting the best marks and do not focus entirely on understanding. Towards the end of the year students will categorise themselves more realistically after having exposed to the assessment model. Taking the case of 2015 (Table 15), 9 of the 29 students who initially felt they were strategic learners classified themselves as deep learners in the 'After ASSIST' questionnaire. This 9.4% increase strengthens the assumption that the assessment model as an intervention shapes students and encourages deep learning.

4 Conclusion

The purpose of this paper was to evaluate the assessment practice for Mathematical Statistics (WST) 133/143 in the four year extended programme at the Mamelodi Campus of the University of Pretoria for the extend to which it succeeds in encouraging a deep learning approach in students. Cross tabulations were compiled from the results that students obtained in the reflection level of the ASSIST questionnaire, project work and final module marks to provide a thorough descriptive analysis about the relationship between how students performed on an action level task (project work and final module mark) and their own reflective learning approach ('After ASSIST').

The three learning approaches were cross tabulated against one another to analyse marginal homogeneity. The ASSIST results of before and after the intevention had been exposed to students serves a strong indication of the impact of the ICE assessment model on a reflective level with a reported increase in all three years investigated in the number of deep level learners. This consistent rise is depicted in tables 13, 14 and 15 (5 in 2013, 29 in 2014 and 9 in 2015).

The project work activity was completed in groups of 4-5 which brought with it limitations in terms of allocating accurate marks for the work each individual did. Results however supported the strength of the argument that the implementation of the ICE model shapes students' approach to learning in the extended programme module where the p-values obtained in table 7 support the statistical significant of the descriptive findings in tables 4, 5 and 6.

It was further found that the final module marks as a response for the individual student's movement from surface to deep learning also depict an improvement across all three years among the students. The kappa statistics for all three years in question show a slight agreement which invariably support the significance of the ICE model of assessment as a treatment in its intention to encourage deep levels of learning approach.

Overall, results showed that the implementation of the ICE model was effective in shaping students' approach to learning towards deeper levels of learning in all three facets that were investigated as the statistics pertaining to agreement and significance align with the descriptive findings.

References

[1] Molatelo S. Kgowedi, Validation of an ASSIST (Approaches and Study Skills Inventory) questionnaire. STK 795 Essay, Department of Statistics, University of Pretoria, 2013.

[2] The data analysis for this essay was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA.

[3] Nur Fadhlina Zainal Abedin, Zuraida Jaafar, Sakinah Husain, and Rosnani Abdullah. The validity of assist as a measurement of learning approach among mdab students. *Procedia-Social and Behavioral Sciences*, 90:549–557, 2013.

[4] John B Biggs and Kevin F Collis. *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome)*. Academic Press, 2014.

[5] Sue Bloxham and Pete Boyd. *Developing Effective Assessment In Higher Education: A Practical Guide: A Practical Guide*. McGraw-Hill Education (UK), 2007.

[6] Noel Entwistle. Promoting deep learning through teaching and assessment: conceptual frameworks and educational contexts. In *TLRP conference, Leicester*. Citeseer, 2000.

[7] David R Krathwohl. A revision of bloom's taxonomy: An overview. *Theory into practice*, 41(4):212–218, 2002.

[8] Paul Ramsden et al. The context of learning in academic departments. *The experience of learning*, 2:198–216, 1997.

[9] John TE Richardson. Methodological issues in questionnaire-based research on student learning in higher education. *Educational Psychology Review*, 16(4):347–358, 2004.

[10] C Troskie-de Bruin and D Otto. The influence of assessment practices on students' learning approach. *South African Journal of Higher Education*, 18(2):322–335, 2004.

[11] Anthony J Viera and Joanne M Garrett. Understanding interobserver agreement: the kappa statistic. *Fam Med*, 37(5):360–363, 2005.

[12] Dylan Wiliam, Clare Lee, Christine Harrison, and Paul Black. Teachers developing assessments for learning: Impact on student achievement. *Assessment in Education*, 2010.

[13] Mantz Yorke. Employability in higher education: what it is–what it is not. *Learning and Employability Series*, 1, 2006.

[14] S Fostaty Young. Teaching, learning, and assessment in higher education: Using ice to improve student learning. In *Proceedings of the Improving Student Learning Symposium, London, UK*, volume 13, pages 105–115, 2005.

[15] Sue Fostaty Young, C Susan Fostaty Young, and Robert J Wilson. *Assessment and learning: The ICE approach*. Portage & Main Press, 2000.

Appendix

```
data research; set sasuser.tableone;
    proc import OUT = sasuser.Tableone DATAFILE= "C:\Users\Sizwe Mbele\Desktop\STK795 Analysis\SAS\2013Table.xlsx" DBMS = xlsx REPLACE; GETNAMES = YES; DATAROW = 2;
    RUN;
    */ WST 133 2014 Data; proc import OUT = sasuser.TableTwo DATAFILE= "C:\Users\Sizwe Mbele\Desktop\STK795 Analysis\SAS\2014Table.xlsx" DBMS = xlsx REPLACE; GETNAMES = YES; DATAROW = 2;
    RUN;
    *2015 data for 133; proc import OUT = sasuser.TableThree DATAFILE= "C:\Users\Sizwe Mbele\Desktop\STK795 Analysis\SAS\2015Table.xlsx" DBMS = xlsx REPLACE; GETNAMES = YES; DATAROW = 2;
    RUN; **Final module marks**; **2013**; proc import OUT = sasuser.Tablefour DATAFILE= "C:\Users\Sizwe Mbele\Desktop\2013FM.xlsx" DBMS = xlsx REPLACE; GETNAMES = YES; DATAROW = 2;
    RUN;
    **2014**; proc import OUT = sasuser.Tablefive DATAFILE= "C:\Users\Sizwe Mbele\Desktop\2014FM.xlsx" DBMS = xlsx REPLACE; GETNAMES = YES; DATAROW = 2;
    RUN;
    **2015**; proc import OUT = sasuser.Tablesix DATAFILE= "C:\Users\Sizwe Mbele\Desktop\2015FM.xlsx" DBMS = xlsx REPLACE; GETNAMES = YES; DATAROW = 2;
    RUN;
    proc freq data = research; Title 'Bowkwer - McNemar Test for samples - 2013'; tables after*project_classification_133/agre
test kappa; run;
    *2014 Bowker-McNemar; proc freq data = sasuser.tabletwo; Title 'Bowkwer - McNemar Test for samples - 2014'; tables after*project_classification/ agree; test kappa; run;
    *2015 Bowker-McNemar; proc freq data = sasuser.tablethree; Title 'Bowkwer - McNemar Test for samples - 2015'; tables after*project_classification/ chisq agree; test kappa; run;
    **Statistics for final module marks**;
    **2013**;
    proc freq data = sasuser.tablefour; Title 'Kappa Statistics - Final module marks - 2013'; tables final_mark_classification*after; test kappa; run;
    **2014**; proc freq data = sasuser.tablefive; Title 'Kappa Statistics - Final module marks - 2014'; tables final2014*after; test kappa; run;
    **2015**; proc freq data = sasuser.tablesix; Title 'Kappa Statistics - Final module marks - 2015'; tables final2015*after; test kappa; run;
    */2013 Graphs; pattern1 color = pink; axis1 minor = none label = ("After ASSIST Classification"); axis2 minor=none label=("Frequency") order=(0 to 200 by 10) ; PROC gchart DATA = research; VBAR after/ discrete width = 8 inside = freq raxis=axis2 maxis = axis1; RUN;
    pattern1 color = lightblue; axis1 minor = none label = ("Project Work Classification"); axis2 minor=none label=("Frequency") order=(0 to 200 by 10) ; PROC gchart DATA = research; VBAR project_classification_133/discrete width = 8 inside = freq raxis=axis2 maxis = axis1; RUN;
    ods listing gpath="C:\Users\Sizwe Mbele\Desktop\STK795 Analysis";
    */2014 Graphs; pattern1 color = pink; axis1 minor = none label = ("After ASSIST Classification"); PROC gchart DATA = sasuser.tabletwo; VBAR after/discrete width = 8 inside = freq raxis=axis2 maxis = axis1; RUN;
    pattern1 color = lightblue; axis1 minor = none label = ("Project Work Classification"); PROC gchart DATA = sasuser.tabletwo; VBAR project_classification/discrete width = 8 inside = freq raxis=axis2 maxis = axis1; RUN;
    *2015 Graphs; pattern1 color = pink; axis1 minor = none label = ("After ASSIST Classification");
    PROC gchart DATA = sasuser.tablethree; VBAR after/discrete width = 8 inside = freq raxis=axis2 maxis = axis1; RUN;
    pattern1 color = lightblue; axis1 minor = none label = ("Project Work Classification");
    PROC gchart DATA = sasuser.tablethree; VBAR project_classification/discrete width = 8 inside = freq raxis=axis2 maxis = axis1; RUN;
```

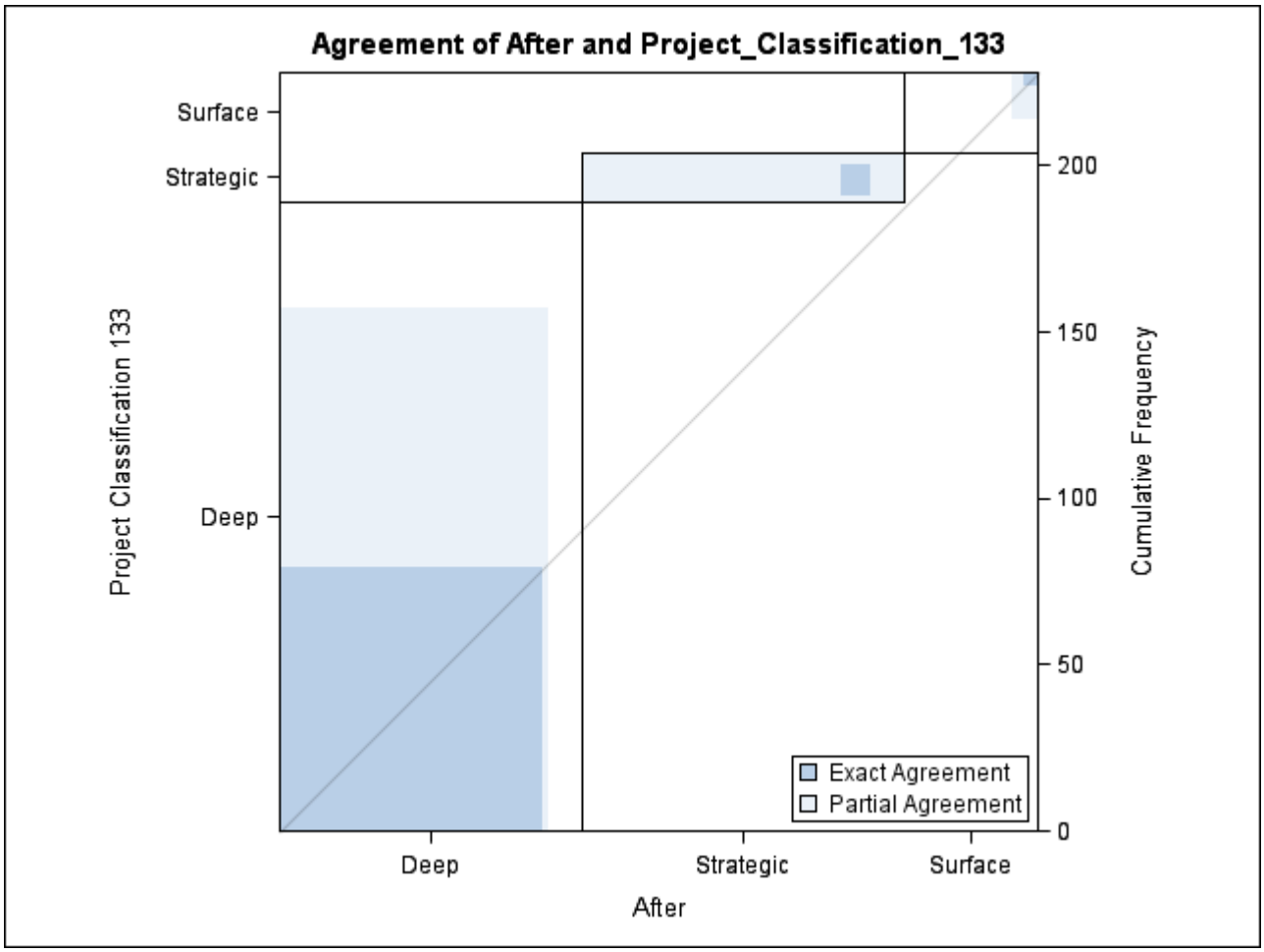


Figure 4: 2013 Agreement plot between project work and 'After ASSIST'

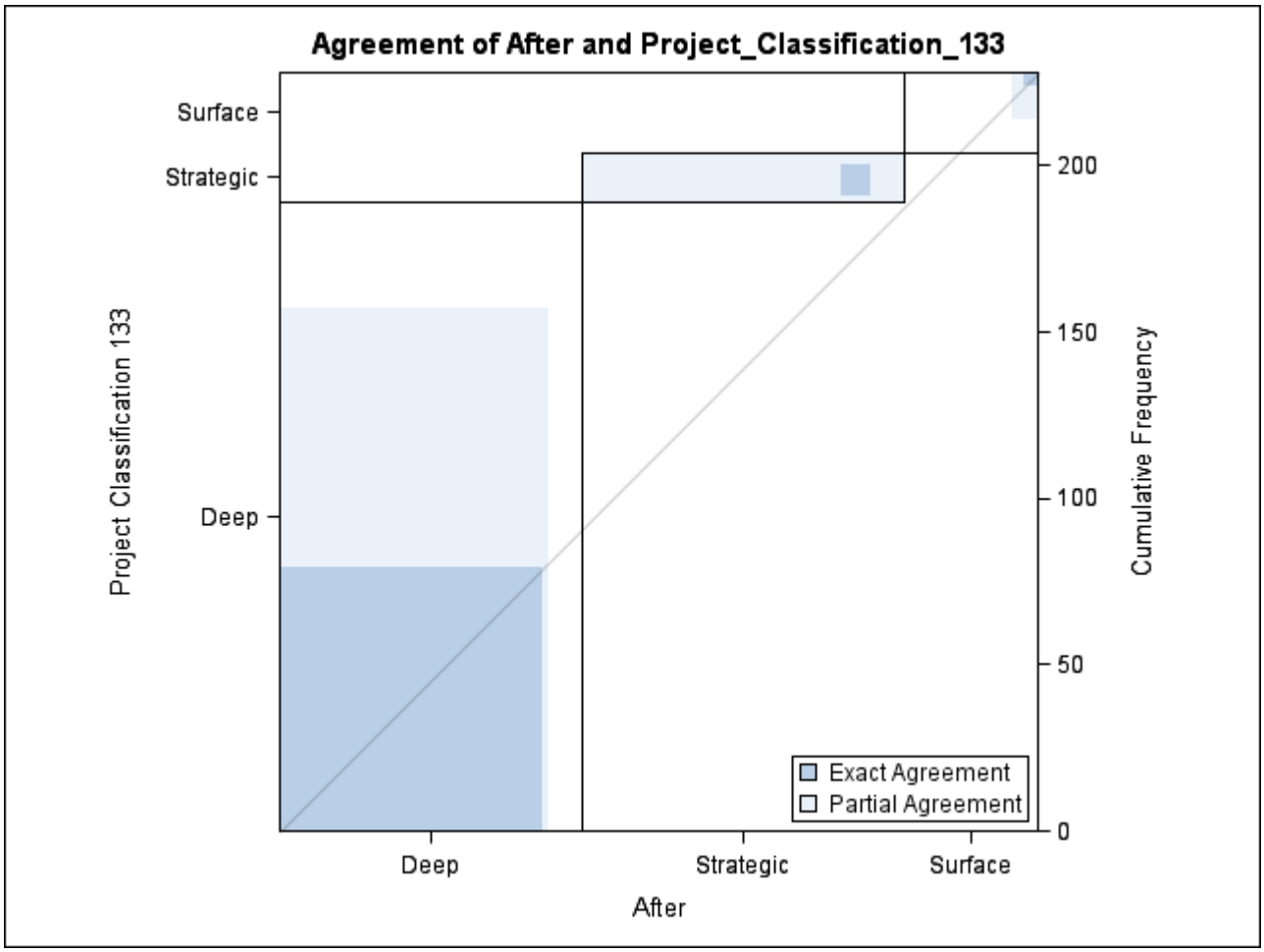


Figure 5: 2014 Agreement plot between project work and 'After ASSIST'

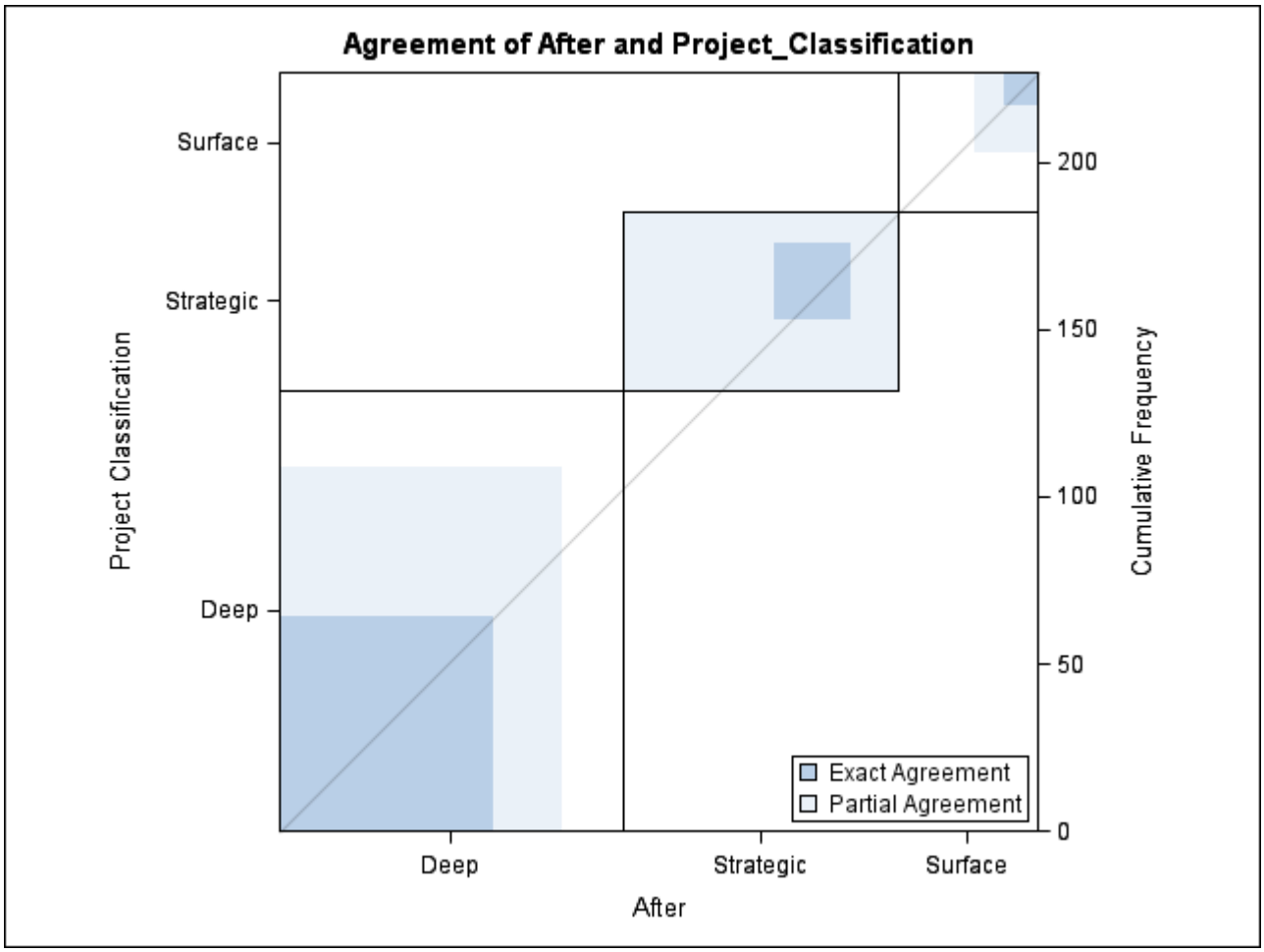


Figure 6: 2015 Agreement plot between project work and 'After ASSIST'

Evaluation of the run-length distribution of the Shewhart-type Q-chart for the gamma distribution

Carola Meyer u12280276

STK795 Research Report

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

Supervisor: Dr. S.W. Human

Department of Statistics, University of Pretoria

November 2, 2016

Abstract

Statistical Process Control (SPC) is concerned with monitoring key quality attributes of a process. Traditionally an experimental phase i.e. Phase I, is used to estimate the charting parameters for on-line Phase II monitoring purposes. In practice this in an expensive and sometimes impractical route to follow. Additionally, for the case of charting a statistic with a skewed distribution, using control limits with three standard deviations on the upper as well as the lower control will prove ineffective. In this research a Shewhart-type Q-chart based on the gamma distribution is investigated with the help of a SAS program; it is a self-starting chart with standardised control limits.

Declaration

I, *Carola Ilse Meyer,* 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.

_____
Carola Ilse Meyer

_____
Dr. S.W. Human

_____
Date

Contents

1 Introduction **5**

2 Background Theory **6**
2.1 The relationship between the exponential, gamma, chi-square and F-distribution 6
2.2 Derivation of the two-sample statistic 7
2.3 Derivation of the charting statistic 9
2.4 Control limits 10
2.5 Classical probability integral transformation 10
2.6 Run-length and other measures 11

3 Application **11**
3.1 Description of SAS program 11
3.2 Simulation results 12

4 Conclusion **15**

Appendix **18**

List of Figures

1 Run-length histogram for $n = 5$, $\lambda = 0.5$ 13
2 Average run length 14
3 Average run length and the median run length for $n = 5$ 15
4 Standard deviation of the run length 16

List of Tables

1 Evaluation of Data 14

1 Introduction

Statistical Process Control (SPC) has been in existence since the 1930's, when W.A. Shewhart and W.E. Deming were influential in the establishment of control charts [11]. They developed the so-called Shewhart charts, which chart different statistics from normal distributions on separate charts, e.g. the mean (location), variance (scale) and the range. These statistics are then compared to an upper control limit (UCL) and a lower control limit (LCL) to monitor whether a shift in the process has taken place, which needs to be investigated and when need be corrected. In order to do this, one needs to know the correct parameters against which to compare these statistics. The traditional approach is a system of two phases, namely Phase I (also called the retrospective phase) and Phase II (prospective phase). Phase I consists of taking samples in order to estimate the correct parameters against which the charting statistics can be compared. This only happens when the correct parameters of the process are unknown, which is most of the time [5]. When this has been established to a satisfactory manner, the process of quality control can begin.

The process that follows will either produce statistics that are plotted in-between the allocated control limits and will therefore be in control (IC); or when the statistics are plotted outside the control limits, the chart will give a signal, which will then have to be investigated. There is a distinction between common and assignable causes of signals. In the case of a common cause, the process will still be in control, the signal will just be due to uncontrollable variations in the process. This signal will be called a false alarm. An assignable cause will be due to an error in the process, which needs to be investigated and corrected. In this case, the process is declared out-of-control (OOC).

The traditional approach poses several problems. First of all, not all processes follow a normal distribution. Processes following a different distribution will result in distorted charting and false evaluations of the process. Second, separate charts are needed to monitor the mean and the variance, as the measures are plotted on different scales. This could pose a challenge to the person monitoring the different charts. Third, preparing samples for Phase I testing can be very costly or impractical in the industry, especially when typical recommendations such as 30 samples of size more than 5 need to be taken in order to determine the right parameters [3]. This creates a problem especially in the case where fewer parts need to be produced i.e. the process is only short-run. In their literature review, Jensen et al also point out the problems of using a Phase I to determine the estimators of the parameters, especially concerning the bigger variability in estimated parameters [5].

The issue concerning non-normal processes has mainly been addressed by constructing non-parametric charts i.e. charts that can monitor processes from different distributions, e.g. Chakraborti et al [2]. Another option is to construct a chart for a specific distribution [1].

Gosh et al discovered that a much higher number of measurements were needed in Phase I for the estimation of parameters than the usual 30 [3]. This was the case for charts measuring the location. Quesenberry emphasized this point, by recommending more than a hundred measurements [10]. Jensen et al came to the conclusion that against usual practice, the control limits in Phase II should therefore be adjusted when the process had been documenting an IC mode for too long [5].

Quesenberry devised the Q-chart, which monitors whether a process is stable or not. This means it does not measure the location or scale against a pre-established parameter, but compares the sample means or variances in order to establish any significant changes in these statistic. Monitoring can therefore start at the beginning of the process. These statistics are then transformed in order to chart them on a standardized normal control chart. This is especially helpful for charting statistics with a skewed distribution, like a range or the standard deviation. The advantage of transformed statistics, is that both the location and the scale parameters can be plotted on one chart, which can simplify the monitoring process, as the scale will be the same. In order to establish whether a process is still in control or has started to venture out of control, control limits are established around a centre line. Quesenberry suggests limits of $\pm$ 3 standard deviations. If a process requires more or less sensitive detection abilities, then probability limits of $LCL = q(1-\alpha)$ and $UCL = q\alpha$ can be utilised. "$q\alpha$ is the $(1-\alpha)th$ fractile of a standard normal distribution" [9].

These charts belong to the field of self-starting charts. Hawkins suggested a similar approach, where his self-starting CUSUM charts continuously update the estimates, but charts them as well [4]. Sullivan and Jones also measure the deviation of former sample averages [12], Li et al use "the likelihood ratio test and the exponentially weighted moving average procedure" [7] and Jones broadens the control limits to mirror

the variability of the parameters [6].

One of the disadvantages of Quesenberry's Q-chart, is that it is specified for a normal distribution. Adamski, Human and Bekker succesfully adapted it for the exponential distribution [1]. They use the starting statistic developed by Quesenberry [9] and adapt it for a process with an underlying exponentional distribution. The components of the charting statistic follow a chi-squared distribution, the charting statistic itself follows a new generalized multivariate beta distribution [1].

This report contains the result of simulations of a Q-chart, where the distribution of the separate parts of the charting statistic follow the gamma distribution. The underlying process however is the univariate exponential distribution. A SAS program was written in order to simulate the run-length and calculate various characteristics in order to evaluate the run-length.

2 Background Theory

The Q-chart addresses two features in statistical process control, which can be costly and impractical. It belongs to the start-up charts, which can immediately start to monitor the relevant process, without implementing a Phase I sampling to determine the parameters to be used. The second feature is that it uses standardized control limits or charting statistics. This is especially useful when the charting statistic follows a skewed distribution. The Q-chart does not determine the value of the process parameters, but will merely state whether the process is still in control compared to the previous measured samples. Therefore we will have to make the assumption that the first measurement made in the specific process will be in-control.

2.1 The relationship between the exponential, gamma, chi-square and F-distribution

For this investigation, a charting statistic with three different components is used. The underlying process follows a univariate exponential distribution with the location parameter θ. Each component represent either one sample, or a group of samples. The grouped samples can be combined to form one statistic, as the degrees of freedom (df) for the statistic will depend on the number of samples in the statistic. Each statistic is drawn from the gamma distribution for the purposes of this assignment.

The following results will be needed for the derivation of the charting statistic.

The gamma distribution usually has two parameters: a location parameter (θ) as well as a shape parameter (α). The gamma distribution is used to measure the waiting time for the α^{th} event. The notation of any statistic with a gamma distribution will be e.g. $V \sim GAM(\theta, \alpha)$.

The exponential distribution is a special form of the gamma distribution, in that it is a gamma distribution with a shape parameter of $\alpha = 1$. If this were the case for V, the notation would be: $V \sim GAM(\theta, 1) \sim EXP(\theta)$.

The chi-square distribution also belongs to this family. If the random variable V has a gamma distribution with $\theta = 2$ and $\alpha = \frac{\upsilon}{2}$, i.e. $V \sim GAM(2, \frac{\nu}{2})$, this is identical to a chi-square distribution with $V \sim \chi^2_{\upsilon}$.

For a $EXP(\theta)$ process with identically and independently distributed elements $X_i = (X_1, X_2, ..., X_n)$,

$$\textstyle\sum_{i=1}^{n} X_i \sim GAM(\theta, n) \quad \Rightarrow \overline{X} \sim GAM(\frac{\theta}{n}, n) \quad \Rightarrow \frac{2n\overline{X}}{\theta} \sim \chi^2_{2n} \tag{1}$$

This means that if a random variable P is a function of $\overline{X}$ and takes on the form $\frac{2n\overline{X}}{\theta}$, it will have a chi square distribution with $2n$ degrees of freedom, i.e. we can say that:

$$P = \frac{2n\overline{X}}{\theta} \sim \chi^2_{2n} \tag{2}$$

The chi-square distribution forms a link to the family of the normal distribution. The formula to standardise any normal distribution is: $z = \frac{(X-\overline{X})}{SD}$, or $z = \frac{(X-\mu)}{\sigma}$ with $\overline{X}$ as the sample estimate for μ (the population mean) and SD as an estimate for σ (the population standard deviation). $z^2 = \chi^2_1$, therefore the chi-square distribution can be used to express the sum of squares of a deviation of the mean. The degrees of freedom will denote the number of squared deviations added together, i.e. it could represent the number of samples drawn. The expression for the estimate of an unbiased population variance is $s^2 = \frac{\sum(y-\overline{y})^2}{N-1}$. N

signifies the sample size. This can be expressed in terms of the chi-square distribution: $\chi^2_{N-1} = \frac{(N-1)s^2}{\sigma^2}$. The F-distribution is the ratio of two sample variances or estimates of the sample variances, therefore $F = \frac{s_1^2}{s_2^2}$ but also the ratio of two chi-square statistics, divided by its degrees of freedom: $F = \frac{\chi^2_{\upsilon_1/\upsilon_1}}{\chi^2_{\upsilon_2/\upsilon 2_1}}$. This means that $\frac{\chi^2_{N-1}}{(N-1)} = \frac{(N-1)s^2}{(N-1)\sigma^2} = \frac{s^2}{\sigma^2}$. σ^2 represents the population variance, and therefore in a ratio of two chi-square variable from the same population, this will cancel out:

$$F_{\upsilon_1,\upsilon_2} = \frac{\chi^2_{\upsilon_1/\upsilon_1}}{\chi^2_{\upsilon_2/\upsilon 2_1}} = \frac{s_1^2}{s_2^2} \tag{3}$$

.

2.2 Derivation of the two-sample statistic

The workings and the notations in this report will heavily correspond to those of Adamski et al. [1], including all of the formulae, except for the transformation to the gamma distribution.

In order for charting statistic of the Q-chart to be calculated one will have to draw r samples of size n. It is assumed that all the "observations of the samples are independent and identically distributed [1]." As stated above, the underlying process is assumed to be a univariate exponential distribution. The location parameter θ is assumed to be unknown. The first sample is drawn and the mean of this sample is used as an estimate of θ. The mean of the second sample is then compared to the mean of the first, in order to detect, whether there was a change in the process. Both the sample means are then used to calculate an overall mean. This overall mean will retain information of the both sample one and sample number two. The third sample mean is then compared to this overall mean, consisting of the sample means of sample one and two. When there is no change, the sample mean of sample number three is used to calculate a new overall sample mean. This process will carry on until a shift is detected in the process. From the second sample onwards an overall mean can be calculated with the following formula :

$$\overline{\overline{X}}_r = \tfrac{1}{r}\left[\overline{X}_r + (r-1)\,\overline{\overline{X}}_{r-1}\right] \quad for \quad r = 1, 2, 3, ... \tag{4}$$

Here $\overline{\overline{X}}_r$ denotes the overall mean and consists of the mean of sample r and the overall mean of samples $r = 1, 2, 3, ..., r-1$. $\overline{\overline{X}}_{r-1}$ is multiplied by the number of sample means included in $\overline{\overline{X}}_{r-1}$. The sum of the overall mean $\overline{\overline{X}}_{r-1}$ and the current sample mean is then divided by the total amount of sample means considered. When the process is in control $\overline{\overline{X}}_r$ will be the MLE of θ. Essentially, the overall mean is calculated from two samples; the current sample r and a pooling of all the samples from $1 : r-1$. The hypothesis is that both $\overline{X}_r$ and $\overline{\overline{X}}_r$ are independent samples drawn from univariate exponential distributions with the same unknown parameter θ. In order to test this hypothesis the two-sample statistic is used:

$$U_r^* = \frac{\overline{X}_r}{\overline{\overline{X}}_{r-1}} \tag{5}$$

It is assumed that the individual samples in $\overline{\overline{X}}_{r-1}$ are drawn from a univariate exponential distribution with parameter θ and those in $\overline{X}_r$ from an univariate exponential distribution with parameter θ_1, i.e. both samples are independently distributed. $\theta_1 = \lambda\theta$ and $\lambda > 0$. λ signifies the shift in the process. When $\theta = \theta_1$, $\lambda = 1$ that means that no shift has occured. When it is found that the $\overline{X}_r = \overline{\overline{X}}_{r-1}$, the conclusion is reached that both samples are drawn from an univariate exponential distribution with the same unknown parameter θ.

In order to use U_r^* , which has a F distribution, we need to find two random variables, which are a function of $\overline{X}_r$ and $\overline{\overline{X}}_{r-1}$ respectively and have a chi-square distribution, in order for U_r^* to test whether $\theta = \theta_1$.

From (2) we can deduce the form of a random variable as a function from the overall sample mean $\overline{\overline{X}}_{r-1}$ with a chi-square distribution:

$$Y = \frac{2n(r-1)\overline{\overline{X}}_{r-1}}{\theta} \sim \chi^2_{2n(r-1)} \tag{6}$$

where $n$signifies the number of measurements in each sample and $r-1$ represents the number of samples represented by $\overline{\overline{X}}_{r-1}$, as

$$\overline{\overline{X}}_{r-1} = \quad \frac{1}{r-1}\sum_{i=1}^{r-1}\overline{X}_i = \frac{1}{(r-1)n}\sum_{i=1}^{r-1}\sum_{j=1}^{n}X_{ij} \qquad for \quad r = 1,2,3,...$$

$\overline{X}_i$ represents the sample means of sample i and X_{ij} the j^{th}measurement in sample i.
Equation (6) can also be written in the following ways:

$$\overline{\overline{X}}_{r-1} = \theta\frac{Y}{2n(r-1)}$$

$$\frac{Y}{2n(r-1)} = \frac{\overline{\overline{X}}_{r-1}}{\theta} \tag{7}$$

Here Y represents the chi-square statistic divided by its degrees of freedom from (3), thus preparing it for the F-distribution.

For sample r the parameter is set as θ_1 before it has been established whether $\theta_1 = \theta$, as every sample drawn is assumed to be drawn from an independent distribution. $\overline{X}_r$ is the mean for only one sample and therefore a random variable that is a function of $\overline{X}_r$ will have the following form, where the degrees of freedom indicate that only one sample was considered:

$$X = \frac{2n\overline{X}_r}{\theta_1} \sim \chi^2_{2n} \tag{8}$$

Where

$$\overline{X}_r = \theta_1\frac{X}{2n}$$

and

$$\frac{X}{2n} = \frac{\overline{X}_r}{\theta_1}$$

U_r^* can be rewritten as

$$U_r^* = \lambda Z$$

where

$$\lambda = \frac{\theta_1}{\theta}$$

and

$$Z = \frac{\frac{X}{2n}}{\frac{Y}{2n(r-1)}} = \frac{\frac{\overline{X}_r}{\theta_1}}{\frac{\overline{\overline{X}}_{r-1}}{\theta}} \sim F_{2n,2n(r-1)}$$

This means:

$$U_r^* = \frac{\theta_1}{\theta}\frac{\frac{\overline{X}_r}{\theta_1}}{\frac{\overline{\overline{X}}_{r-1}}{\theta}} = \frac{\overline{X}_r}{\overline{\overline{X}}_{r-1}} \sim F_{2n,2n(r-1)} \tag{9}$$

In essence (5) was multiplied by $\frac{\theta_1}{\theta}/\frac{\theta_1}{\theta} = 1$.
When no shift has occured $U_r^* = Z$. After a shift, i.e. $\lambda \neq 1$ and $U_r^* = \lambda Z$.
In order to transform Y and X to the gamma distribution, we consider (1):

$$\sum_{i=1}^{n}X_i \sim GAM(\theta, n) \quad \Rightarrow \overline{X} \sim GAM(\tfrac{\theta}{n}, n) \quad \Rightarrow \tfrac{2n\overline{X}}{\theta} \sim \chi^2_{2n}$$

The degrees of freedom of the chi-square distribution are not dependent on θ, however the degrees of freedom of the gamma distribution are. This poses problems for the simulation of the process, as in order to estimate the MLE for θ, we need an estimate for θ in the degrees of freedom. We can however transform $\overline{X}$ in such a

way that the gamma distribution will have a location parameter of one. i.e. we multiply $\overline{X}$ and the location parameter by $\frac{n}{\theta}$:

$$\overline{X} \sim GAM(\frac{\theta}{n}, n) \Rightarrow \frac{n\overline{X}}{\theta} \sim GAM(1, n)$$

Therefore:

$$Y = \frac{n(r-1)\overline{\overline{X}}_{r-1}}{\theta} \sim GAM(1, n(r-1)) \quad and \quad X = \frac{n\overline{X}_r}{\theta_1} \sim GAM(1, n) \tag{10}$$

2.3 Derivation of the charting statistic

For good quality control, it is essential to detect a shift in the process as soon as possible. We assume that a permanent shift will take place at sample k and this shift will be detected when the charting statistic is bigger than the UCL or smaller than the LCL. After a shift, the process parameter will no longer be θ but θ_1. Therefore all the means from sample 1 to sample $k-1$ are originating from samples drawn from a univariate exponential distribution with parameter θ and all the sample means from sample k to sample $k+t-1$ stem from samples from a univariate exponential distribution with θ_1 as a parameter. We are therefore concerned with the following statistic: U^*_{k+t}, $k = 2, 3, ...$ and $t = 0, 1, 2, ...$

$$U^*_{k+t} = \frac{\overline{X}_{k+t}}{\overline{\overline{X}}_{k+t-1}} \tag{11}$$

The overall mean for $k+t-1$ in the denominator is divided into two parts: the overall mean of samples $1:k-1$, i.e. all the samples before the shift occurred and the overall mean from samples $k:k+t-1$, i.e. all the samples after the shift occurred until one sample before the current sample. From the form of equation (4), we can derive the following form of equation (11):

$$U^*_{k+t} = \frac{\overline{X}_{k+t}}{\frac{1}{k+t-1}\left[(k-1)\overline{\overline{X}}_{[1:k-1]} + t\overline{\overline{X}}_{[k:k+t-1]}\right]} \tag{12}$$

When we multiply (12) with $\frac{\theta_1}{\theta\theta_1}/\frac{\theta_1}{\theta\theta_1}$, we will obtain the following form:

$$U^*_{k+t} = \frac{\frac{\theta_1}{\theta}\frac{\overline{X}_{k+t}}{\theta_1}}{\frac{1}{k+t-1}\left[(k-1)\frac{\theta_1}{\theta_1}\frac{\overline{\overline{X}}_{k-1}}{\theta} + t\frac{\theta_1}{\theta}\frac{\overline{\overline{X}}_{[k:k+t-1]}}{\theta_1}\right]} \tag{13}$$

This will prepare the charting statistic for the gamma distribution form, where we divide each sample mean by its relevant estimated parameter: $\frac{\overline{\overline{X}}_{k-1}}{\theta}$; $\frac{\overline{\overline{X}}_{[k:k+t-1]}}{\theta_1}$ and $\frac{\overline{X}_{k+t}}{\theta_1}$. By substituting $\lambda = \frac{\theta_1}{\theta}$ into (13), we get:

$$U^*_{k+t} = \frac{\lambda\frac{\overline{X}_{k+t}}{\theta_1}}{\frac{1}{k+t-1}\left[(k-1)\frac{\overline{\overline{X}}_{k-1}}{\theta} + t\lambda\frac{\overline{\overline{X}}_{[k:k+t-1]}}{\theta_1}\right]}$$

We can now substitute the three statistics in the form of (7) into the preceding formula:

$$U^*_{k+t} = (k+t-1)\frac{\lambda\left\{\frac{W_{k+t}}{n}\right\}}{\left[(k-1)\left\{\frac{W_{[1:k-1]}}{n(k-1)}\right\} + t\lambda\left\{\frac{W_{[k:k+t-1]}}{nt}\right\}\right]}$$

We now cancel $\frac{(k-1)}{(k-1)}$, $\frac{t}{t}$ and the $n's$. This will bring us to the ultimate form of the charting statistic. The F-distribution needs the degrees of freedom from the chi-square distribution, as it is only defined as the ratio of two chi-square distributed variables and not of two gamma distributed variables:

$$U^*_{k+t} = (k+t-1)\frac{\lambda W_{k+t}}{W_{[1:k-1]} + \lambda W_{[k:k+t-1]}} \sim F_{2n,(2n(k-1)+2nt)} \tag{14}$$

The statistic consists of the following random variables:

$$W_{k+t} = \frac{n\overline{X}_{k+t}}{\theta_1} \sim GAM(1, n)$$

$$W_{[1:k-1]} = \frac{n(k-1)\overline{\overline{X}}_{k-1}}{\theta} \sim GAM(1, n(k-1))$$

$$W_{[k:k+t-1]} = \frac{nt\overline{\overline{X}}_{[k:k+t-1]}}{\theta_1} \sim GAM(1, nt)$$

When t=0, the term $W_{[k:k+t-1]}$ will be undefined and the denominator will only consist of $W_{[1:k-1]}$

2.4 Control limits

The Q-chart is part of a Shewhart type chart, where the charting statistic is charted between or outside control limits:

$$LCL^*_{k+t} < U^*_{k+t} < UCL^*_{k+t}$$

For the purposes of this report, the control limits will be

$$UCL_{k+t} = \frac{UCL^*_{k+t}}{(k+t-1)}$$

LCL_{k+t} will have the same distribution.

Using control limits with 3-sigma limits will work for statistics which are normally distributed, or rather symmetrical around a certain point; however if the distribution of the statistic to be charted is skewed, then a 3-sigma limit will prove to be impractical for one direction or the other.

For a normal distribution Shewhart chart, one will choose the control limits after the following formula.

$$UCL = \mu + k\sigma$$

$$CL = \mu$$

$$LCL = \mu - k\sigma$$

According to the empirical rule for data with a bell-shaped distribution, close to 100% of the data will be within 3 standard deviations of the mean, therefore $k = 3$. For the standard normal distribution, the mean will have a value of zero and the standard deviation a value of one. If we substitute this information into the above formula we will obtain control limits of plus/minus three respectively.

2.5 Classical probability integral transformation

The Q-chart uses the classical probability integral transformation [9] to standardize the charting statistic in order to compare it to 3-sigma limits above and below the centre line (CL). This transformation has the advantage that no information about the original charting statistic is lost.

If one takes the CDF of U^*_{k+t} it will return a probability between [0,1], i.e. the

$$F(U^*_{k+t}) = P_{u(k+t)} \sim UNIF(0, 1)$$

where $P_{u(k+t)}$ signifies the given probability of the CDF of U^*_{k+t}.

If we use the inverse function of the standard normal distribution and substitute $P_{u(k+t)}$ into it, it will return the corresponding z-value of the standard normal function:

$$F^{-1}(P_{u(k+t)}) = Z_{P[u(k+t)]} \sim N(0,1)$$

If this value is less than -3 or larger than 3, the chart will signal.

In the same way we can transform the control limits:

In this instance the control limits (LCL = lower control limit, UCL = upper control limit) are the Z-value of the standard normal distribution $LCL = -3$ and $UCL = 3$.

We take the CDF of LCL and UCL to obtain a probability between [0,1]. For the the LCL:

$$F(LCL) = P_{LCL} \sim UNIF(0,1)$$

We insert the given value into the inverse function $F - distribution$ with the df of the current sample in order to obtain a value which can be compared to the current charting statistic:

$$F^{-1}(P_{LCL}) = X_{P[LCL]} \sim F_{2n,(2n(k-1)+2nt)}$$

The same procedure is followed for the UCL.

2.6 Run-length and other measures

The probability to obtain a signal when the process is still in-control is called the false alarm rate (FAR). It represents the probability associated with the z-value of the control limits derived from the standard normal distribution. In this case it would be -3/3. i.e. $P(Signal) = P(z \leq -3) + P(z \geq 3) = 2 * (1 - 0.99865) = 0.0027$. The reciprocal of this figure will be the in-control average run-length, for a FAR of 0.0027 this will be around 370. Control limits that are narrower will result in a higher FAR. Conversely, wider limits will generate a lower FAR. However a chart with narrower limits will also be able to detect a shift in the process earlier and vice versa. A balance between the cost of the time it takes to investigate a false alarm and the cost to produce more components, which will have to be discarded will have to be found for each individual process. Especially for short-run processes or high cost productions, it could be desirable to have a higher FAR versus higher wastage.

According to Jensen et al [5], if the parameters of a chart are known, the run-length (RL) of a control chart are the number of observations monitored until a signal occurs. For all independent and identically distributed (i.i.d.) samples, the run-length, as a random variable, as well as the control limits will be constants. For a normal Shewhart charts the parameter of the run-length is the probability that a signal will be observed. In the case of known parameters and i.i.d. statistics, the run-length distribution will be a geometric distribution, for any underlying distribution. When parameters are estimated, the RL is no longer geometrically distributed. The probability of a signal therefore does not have a meaningful interpretation. Therefore measures such as the average run length (ARL) and the standard deviation of the run length (SDRL), the median of the run length (MDRL) and different percentiles will form a more complete evaluation of a control chart.

The average run length of a chart is the average number of observations before an observation plots outside the control limits i.e. a signal is obtained. The in control ARL (IC ARL) is the number of observations before a false alarm is obtained and the out-of-control ARL will be the number of observations before a signal, when the process has indeed become OOC.

.

3 Application

3.1 Description of SAS program

The SAS program for this investigation into the run-length distribution of the gamma distribution is contained in the Appendix. It consists of two do-loops: an outer do-loop for the number simulations and an inner do-loop for the calculation of the run length of each simulation.

The following scenarios are simulated:

- the sample sizes $n = 5$ and $n = 10$
- shift ratio $\lambda = 0.5, 1, 1.5, 2, 5$.
- shift time $k = 31$ for every scenario.

For each scenario, 20 000 simulations were executed and different calculations were done on the resulting the data.

For every simulation the maximum considered run length calculation is capped at 2000. The control limits are set at 3 standard deviations of the standard normal distribution above and below a central line of 0, i.e. a z-value of -3 & 3. The CDF of the control limits returns the uniform distributed value between 0 and1, which represents the probability of a signal above the UCL and below the LCL. These transformation are done outside the inner do-loop. The last transformation, to compare the charting statistic to the control limits is done inside the inner do-loop for every iteration.

The first sample for every calculation represents all the samples before the shift $W_{[1:k-1]}$. The degrees of freedom for the chi-square distribution of this statistic would be $2n$ * the number of samples taken. In this instance it is $k-1$ samples, i.e. $df = 2n(k-1)$. For the gamma distribution the degrees of freedom are n * number of samples taken, i.e. $W_{[1:k-1]} \sim GAM(n(k-1))$. The location parameter for the gamma distribution is set as a default of one in SAS and is therefore omitted in the notation for this section.

The inner do-loop consists of two sections, one for time $t = 0$ and one for time $k + t$. At the time of the shift $t = 0$, i.e. no sample has been collected after the shift has taken place. The charting statistic will therefore only consist of the ratio of the current sample, $W_{k+t} \sim GAM(n)$ and $W_{[1:k-1]} \sim GAM(n(k-1)$, therefore

$$U_{k+t} = \frac{\lambda W_{k+t}}{W_{[1:k-1]}} \sim F_{2n,2n(k-1)} \tag{15}$$

Next the control limits need to be transformed to a F statistic in order to be compared to U_{k+t}. The Quantile function is used $LCLf = Quantile('F', LCLc, 2n, 2n(k-1))/(k-1)$ with the current df of U_{k+t}. This statistic can then be compared to the suitable control limits in order to check, whether the statistic plots inside or outside of the control limits. Should it plot outside the control limits, the do-loop will end and a run length of one will be recorded for this simulation. If the charting statistic plots inside the control limits, the program will continue to run.

$W_{[k:k+t-1]}$ from the current iteration is updated from W_{k+t} and $W_{[k:k+t-1]}$ of the previous iteration. The degrees of freedom are $W_{[k:k+t-1]} \sim GAM(nt)$. Even though the statistic only ranges until $t-1$, the number of samples found in $W_{[k:k+t-1]} = t$, as counting starts with $t = 0$.

In essence, only $W_{k+t} \sim GAM(nt)$ is simulated in each do-loop. $W_{[1:k-1]}$ does not change, as we only assume one shift at time $k = 31$ in our simulation and therefore the sample before the shift can be seen as a constant.

For $t > 0$ the charting statistic is the following:

$$U_{k+t} = \frac{\lambda W_{k+t}}{W_{[1:k-1]} + \lambda W_{[k:k+t-1]}} \sim F_{2n,(2n(k-1)+2nt)}$$

The control limits are transformed to the following Quantile function e.g. for LCL. The same is valid for UCL:

$$LCLf = Quantile('F', LCLc, 2n, (2n(k-1)2nt))/(k+t-1)$$

The do-loop repeats itself until a charting statistic falls outside the control limits. The do-loop is discontinued and the count is recorded as the run length of the current simulation.

3.2 Simulation results

When the parameters of a chart are estimated, as is the case with the Q-chart, the run length will have a different distribution for every different type of settings, i.e. the size of the shift and the sample size will play a role. With known parameters, the run length will have a geometric distribution. This is the reason that

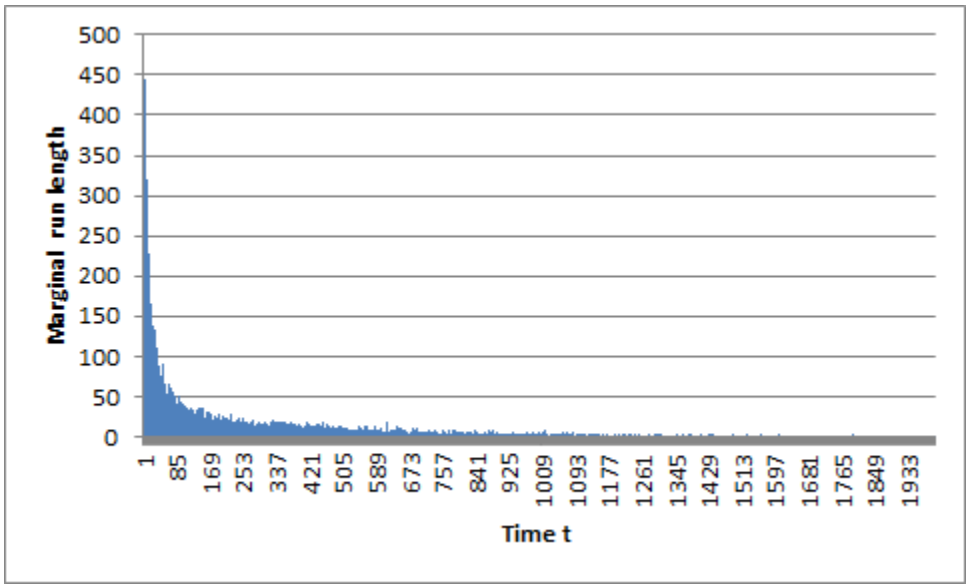


Figure 1: Run-length histogram for $n = 5$, $\lambda = 0.5$

different moments are used to evaluate the run length. The average run length (ARL), the standard deviation of the run-length (SDRL), the median run-length (MDRL) and different percentiles serve this purpose. Gosh et al. found that as the simulations tend to infinity that the run length distribution will converge to a geometric distribution [3]. Table 1 contains the values obtained for all combinations.

Figure 1 is an example of the run-length distribution of for $n = 5$ and $\lambda = 0.5$. The run-length distribution is highly skewed to the right. This attribute will affect some of the evaluation measures.

The ARL of $\lambda = 1$ represents the in-control ARL as no shift has occurred and both the current mean and the overall mean of the preceding samples are still equal.

From Table 1 it is clear that the chart performed in a satisfactory manner with an in-control ARL of around 370 for both $n = 5$ and $n = 10$.

$\lambda = 0.5$ represents a reduction in θ of 50% and $\lambda = 1.5$ an increase of 50% in θ.

An interesting feature can be seen in Figure 2 where the ARL of both $n = 5$ and $n = 10$ is plotted. For $n = 5$ it takes longer to detect a decrease in θ versus the increase, whereas for $n = 10$ it is the other way around.

The MDRL is consistently lower than the ARL; this is an indication of the right skewedness of the run-length distribution. This means that the density of observed run-lengths is higher at values lower than the ARL and more dispersed at values higher than the ARL.

For the IC ARL the run-length will approximate a geometric distribution. In this case, the SDRL will be nearly as large as the ARL [8].

The SDRL also shows signs of a right skewed distribution. It is consistently higher than the ARL. The ARL ranges between 1,0189 (for $n = 10$ and $\lambda = 5$) and 373,87625 (for $n = 10$ and $\lambda = 1$) for the data obtained; however the 95th percentile displays a maximum value of 1094,5 (for $n = 5$ and $\lambda = 1$). This value is nearly three times higher then the ARL. This explains a higher SDRL.

The SDRL for $n = 5$ is higher than for $n = 10$. This is due to a higher variability of run lengths for smaller sample sizes. The bigger the sample size, the more the SDRL will converge to the ARL. This characteristic can be observed from the IC ARL. Here the sample size is not higher, however the SDRL nearly equals the ARL, i.e. it displays the characteristic that if the variability of the ARL smaller is, the SDRL will be close to the value of the ARL.

In general the chart will detect larger shifts faster, i.e. a very low ARL. This can be seen for $\lambda = 5$, which represents an increase in theta of 500%, the ARL is around 1. Even the 95th percentile only displays a run-length of 2 for $n = 5$ and 1 for $n = 10$. The SDRL decreases significantly at $\lambda = 2$ as well, this means

N	K	λ	ARL	SDRL	MDRL	P05	P10	P25	P50	P75	P90	P95
5	31	0,5	252,7057	359,5427408	97	3	6	20	97	351	717	1002
10	31	0,5	71,25525	197,265758	10	1	2	4	10	34	171	408,5
5	31	1	367,042	365,1460763	254	19	39	106	254	512	844	1094,5
10	31	1	373,87625	374,4088665	260	20	39	107	260	515	856,5	1123
5	31	1,5	194,1631	299,7311371	63	2	4	13	63	254	572	811
10	31	1,5	121,3957	247,2846431	18	1	2	5	18	104	396	632,5
5	31	2	47,0885	149,1312381	6	1	1	2	6	19	91	257
10	31	2	7,0756	39,23860303	2	1	1	1	2	5	10	16
5	31	5	1,23745	0,573571222	1	1	1	1	1	1	2	2
10	31	5	1,0189	0,139440891	1	1	1	1	1	1	1	1

Table 1: Evaluation of Data

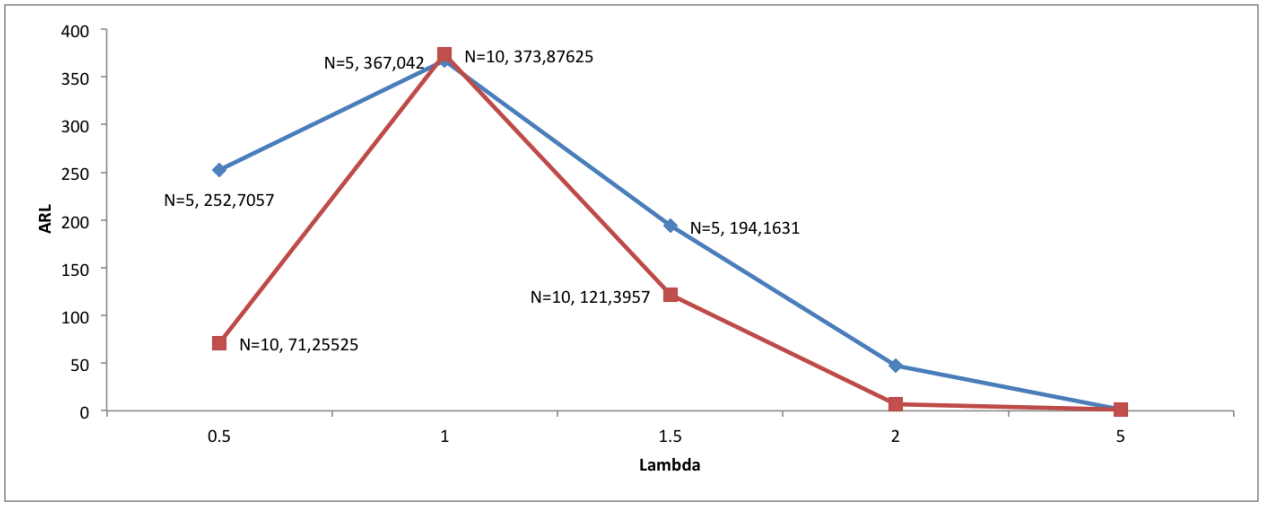


Figure 2: Average run length

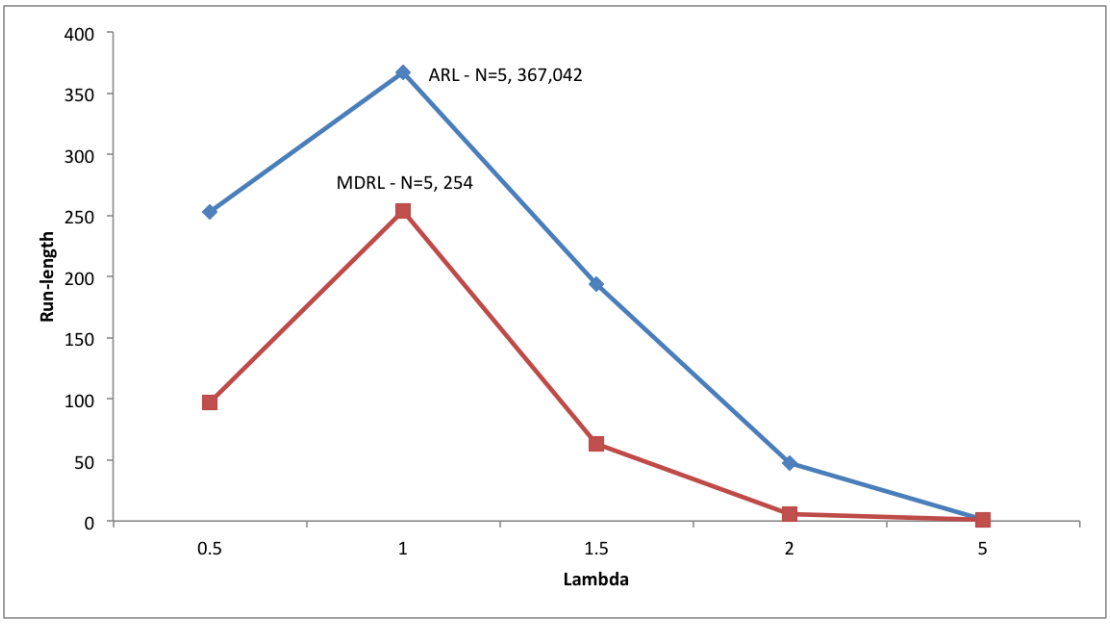


Figure 3: Average run length and the median run length for $n = 5$

that fewer runs will signal on a later stage for such a big shift. The percentiles for $n = 5$ for $\lambda = 0.5$ display a big range in the percentiles for the 5th percentile = 3 and the 95th percentile = 1002. One reason for this big range could be that with smaller sample sizes, one will have a higher variation in the data. The range of $n = 10$ is lower than $n = 5$ for all shifts. According to Luceño and Puig-Pey, a large IC ARL in connection with relatively small values in the lower quantiles is an indication of large amount of false alarms in the early stage of charting [8]. For $\lambda = 1$, that means that 5% of the false alarms will occur at a run length of 19 (for $n = 5$) or 20 (for $n = 10$).

4 Conclusion

Even though the parameters are not estimated in the Q-chart, the ARL, SDRL and MDRL show indications that the run-length distribution could be approximated by the geometric distribution, i.e. a highly right-skewed distribution. The IC ARL is on par with industrial standards (approximately 370) and the OOC ARL is more effective for $n = 10$, as it detects the changes faster with a bigger sample size.

The Q-chart displays characteristics that are well suited for its design, i.e. for short-run processes and situations, where a variety of components need to be processed. However higher sample sizes will still render better results than smaller sample sizes, as a result of the high variation smaller samples might return. One also needs to be careful to make sure that the production line works well from the beginning, as a process that is out of control from the onset, will not be detected immediately. Some knowledge of the nature of the process, i.e. is the process stable from the onset, or does the process need a "warm-up" period [9] is essential, in order to make efficient use of the chart. This is however a practical and workable chart for small-scale producers.

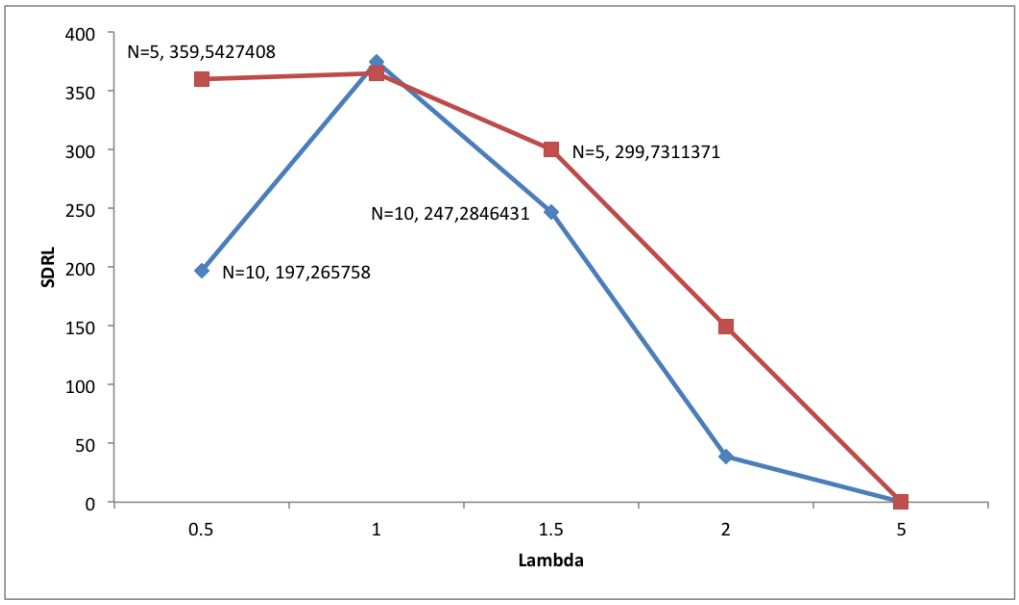


Figure 4: Standard deviation of the run length

References

[1] K. Adamski, S. W. Human, and A. Bekker. A generalized multivariate beta distribution: control charting when the measurements are from an exponential distribution. *Statistical Papers*, 53(4):1045–1064, November 2012.

[2] S. Chakraborti, P. Van der Laan, and S.T. Bakir. Nonparametric control charts: an overview and some results. *Journal of Quality Technology*, 33(3):304, 2001.

[3] B.K. Ghosh, M.R. Reynolds Jr, and V.H. Yer. Shewhart X-charts with estimated process variance. *Communications in Statistics-Theory and Methods*, 10(18):1797–1822, 1981.

[4] D.M. Hawkins. Self-starting CUSUM charts for location and scale. *The Statistician*, 36(4):299–316, 1987.

[5] W.A. Jensen, L.A. Jones-Farmer, C.W. Champ, and W.H. Woodall. Effects of parameter estimation on control chart properties: a literature review. *Journal of Quality Technology*, 38(4):349, 2006.

[6] L.A. Jones. The statistical design of EWMA control charts with estimated parameters. *Journal of Quality Technology*, 34(3):277, 2002.

[7] Z. Li, J. Zhang, and Z. Wang. Self-starting control chart for simultaneously monitoring process mean and variance. *International Journal of Production Research*, 48(15):4537–4553, 2010.

[8] A. Luceño and J. Puig-Pey. Evaluation of the run-length probability distribution for CUSUM charts: assessing chart performance. *Technometrics*, 42(4):411–416, 2000.

[9] C.P. Quesenberry. SPC Q charts for start-up processes and short or long runs. *Journal of Quality Technology*, 23(3):213–224, 1991.

[10] C.P. Quesenberry. The effect of sample size on estimated limits for sample mean and X control charts. *Journal of Quality Technology*, 25(4):237–247, 1993.

[11] W.A. Shewhart and W.E. Deming. *Statistical Method from the Viewpoint of Quality Control*. Courier Corporation, 1939.

[12] J.H. Sullivan and L.A. Jones. A self-starting control chart for multivariate individual observations. *Technometrics*, 44(1):24–33, 2002.

Appendix

```
proc iml;

Print 'SAS program for the evaluation of the run-length distribution of
the Shewhart-type Q-chart for the gamma distribution ' ;

*Variables;
*Number of measurements in each sample;
n=5;
*Time at which the shift takes place;
k=31;
*Shift ratio;
lambda=1.0;

*Number of simulations;
sim=20000;
*Matrix to store the runlength of each simulation;
runl=j(sim, 1,.);
*Maximum runlength to be investigated determined;
maxrunl=2000;

*Creating LCL and UCL;
*Distance of the control limits from the center line;
LCL=j(1,1,-3);
UCL=j(1,1,3);
*Classical probability integral transformation;
LCLc=j(1,1,.);
UCLc=j(1,1,.);
LCLc=CDF('Normal',LCL,0,1);
UCLc=CDF('Normal',UCL,0,1);
*Matrices for final transformation of the control limits;
LCLf=j(1,1,.);
UCLf=j(1,1,.);

*ss = simulation loop variable;
do ss=1 to sim;

        *Obtain first sample representing all samples before the shift;
        W_1_to_k_minus_1=j(1,1,.);
        *Degrees of freedom for a chi square variable;
        df1=2*n*(k-1);
        *Transformation of df to the gamma distribution;
        call randgen(W_1_to_k_minus_1,'gamma',0.5*df1);

        indicator=0;
        count=0;

        *At shift at time k t=0;
        t=0;

        *Represents the (k:t-2) samples;
```

```
*For t=0, this will also be 0;
W_k_to_k_plus_t_minus_1_a=0;

* i = sampling loop variable;
* Chart signals,when indicator =1;
do i = 0 to 1000000 until (indicator = 1);
        count=count+1;

        if i = 0 then do;
                *Obtain sample of the shift, for t=0;
                df3=2*n;
                W_k_plus_t=j(1,1,.);
                call randgen(W_k_plus_t,'gamma',0.5*df3);

                *Calculate charting statistic at k i.e. at t=0;
                U_k_plus_t= (lambda* W_k_plus_t)/( W_1_to_k_minus_1);

                *Transformation taking the df into account;
                LCLf=Quantile('F',LCLc,df3,df1)/(k-1);
                *Chisquare df's for the F distribution;
                UCLf=Quantile('F',UCLc,df3,df1)/(k-1);

                *Check for signal i.e. does the chart plot above UCL or below LCL;
                if U_k_plus_t>UCLf | U_k_plus_t<LCLf then indicator=1;
        end;
        else do;
                t=t+1;

                *Sample set between the shift and current sample (k+t-1);
                *Previous sample (t-1);
                W_k_to_k_plus_t_minus_1_b=W_k_plus_t;
                *Sum of (k+t-2) and (t-1) to generate the (k+t-1) sample;
                W_k_to_k_plus_t_minus_1
                = W_k_to_k_plus_t_minus_1_a + W_k_to_k_plus_t_minus_1_b;

                *Set as (t-2) sample for next run of loop;
                W_k_to_k_plus_t_minus_1_a = W_k_to_k_plus_t_minus_1;
                *Degrees of freedom for the sum of (k+t-1) samples;
                df2=2*n*t;

                *Obtain the (k+t) sample;
                df3=2*n;
                W_k_plus_t=j(1,1,.);
                call randgen(W_k_plus_t,'gamma',0.5*df3);

                *Calculate charting statistic for all samples when t>0;
                U_k_plus_t=((lambda* W_k_plus_t)/
                ( W_1_to_k_minus_1 + lambda*W_k_to_k_plus_t_minus_1));

                *Final transformation of control limits;
                LCLf=Quantile('F',LCLc,df3,(df1+df2))/(k+t-1);
                UCLf=Quantile('F',UCLc,df3,(df1+df2))/(k+t-1);
```

```
                        *Check for signal;
                        if U_k_plus_t>UCLf | U_k_plus_t< LCLf then indicator=1;
                end;
        end;
        *Count represents the run-length;
        runl[ss,1]=count;
        *Matrix to calculate the pmf for each sample after k;
        r=j(count,1,1);
        *Standardize the vector in order to concatenate later;
        kt=shape(r,maxrunl,1,0);
        *Concatenation of kt vectors;
        ksum=ksum||kt;
end;
*At time t obtain the sum for in control indicators;
kcount=ksum[,+];
*Vector of sample number index;
knr=1:maxrunl;
tknr=knr`;
*Probability of no signal at time k+i;
prob1=kcount/sim;

*Create labels for output;
np1=repeat(n,2000,1);
kp1=repeat(k,2000,1);
lambdap1=repeat(lambda,2000,1);

*Calculate pmf at k=1;
pmf1=j(1,1,1)-prob1[1,1];
*Calculate pmf for 1<k<=maxrunl;
pmf2=prob1[1:maxrunl-1,1]-prob1[2:maxrunl,1];
pmf=pmf1//pmf2;
*Calculate CDF for k+i;
CDF=cusum(pmf);
*Average run-length;
aver=j(1,1,runl[:,]);
*Standard deviation run-length;
sdrl=j(1,1,std(runl));
*Median run-length;
mdrl=j(1,1,median(runl));
*Define which percentiles needed;
p={0.05, 0.10, 0.25, 0.5, 0.75, 0.90, 0.95};
*Calculate percentiles;
call qntl(q, runl, p);
quant=q`;

*Create output matrices;
output1=np1||kp1||lambdap1||tknr||kcount||prob1||pmf||cdf;
output2=n||k||lambda||aver||sdrl||mdrl;
output3=n||k||lambda||quant;

create STK from output1[colname={n k lambda kt tcount prob pmf cdf }];;
append from output1;
```

```
create ARL from output2 [colname={n k lambda ARL sdrl mdrl}];
append from output2;

create percentiles from output3 [colname={n k lambda P05 P10 P25 P50 P75 P90 P95}];
append from output3;

quit;

proc export data = STK
                outfile = "C:\Users\Carola\Documents\STK\stkn5k31L1_0.csv"
                        dbms=dlm replace;
                        delimiter=',';

proc export data = ARL
                outfile = "C:\Users\Carola\Documents\STK\arln5k31L1_0.csv"
                        dbms=dlm replace;
                        delimiter=',';

proc export data = percentiles
                        outfile = "C:\Users\Carola\Documents\STK\pern5k31L1_0.csv"
                        dbms=dlm replace;
                        delimiter=',';

run;
```

Research Report Title

First Name and Surname Student Number

WST795/STK795 Research Report

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

Supervisor(s): Title Initals Surname, Co-supervisor(s): Title Initials Surname

Department of Statistics, University of Pretoria

26 July 2016 (draft 1) / 30 September 2016 (draft of final version) / 2 November 2016 (final)

Abstract

Short summary of the research proposed. This should be a two to three paragraphs long and should fully describe the content and contributions of the research report.

Declaration

I, *full students name,* declare that this essay, submitted in partial fulfillment of the degree *BSc(Hon) Mathematical Statistics / BCom(Hons) Mathematical 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.

_____
Insert Student's full name Here

_____
Insert Supervisor(s) name(s) here

_____
Date

Acknowledgements

Add acknowledgements here (not compulsory).

Contents

1 Introduction **6**

2 Background Theory **6**

3 Application **6**

4 Conclusion **6**
Appendix8

List of Figures

1 Caption 6

List of Tables

1 Example 6

Figure 1: Caption

	Column 1	Column 2
Row 1		
Row 2		

Table 1: Example

1 Introduction

The introduction should provide a detailed description of the topic and the aim of the research report. In addition the literature review should intertwine with this. It is important to always reference where needed. All work from somewhere else requires a reference [1]. Inline equation x. Display equation:

$$x = y \tag{1}$$

$$x = y \tag{2}$$
$$x = y + 1$$

.

Numbered equation:

$$x = y + 1 \tag{3}$$

Equation array:

$$\begin{aligned} x &= y + 2 + 3 \\ &= y + 5. \end{aligned}$$

2 Background Theory

The theory of the topic should be thoroughly discussed in this chapter. The student must should their proficiency on the topic as well as additional insight. This chapter may be separated into a few chapters as *necessary*.

3 Application

The application should be presented in this chapter. Code should be included in an appendix as well as additional output if needed.

4 Conclusion

The conclusion should summarise what was done in the research report. It should also provide shortfalls of the research and recommendations on what could be investigated in future. This section should be an honest summary of the research.

References

[1] R. Anguelov and I. Fabris-Rotelli. LULU operators and discrete pulse transform for multidimensional arrays. *Image Processing, IEEE Transactions on*, 19(3):3012–3023, 2010.

Appendix

Include any additional code, output or data here.

Estimation of large dimensional covariance matrices

Masego Modibane 13061632

WST795 Research Report

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

Supervisor: Dr N Strydom

Department of Statistics, University of Pretoria

2 November 2016

Abstract

The estimation of covariance matrices plays a vital role in industries such as the financial economics industry and more particularly in portfolio selection, risk management and asset pricing. The conventional estimator, namely the sample covariance matrix, becomes problematic in the large dimensional case. Numerous methodologies including shrinkage methods, factor model methods and Bayesian approaches were developed to overcome the problems that arise in this case. An overview of these estimation methodologies for large dimensional covariance matrices with a focus on the application in portfolio selection will be presented in this report.

Declaration

I, *Masego Modibane,* 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.

_____
Masego Modibane

_____
Dr Nina Strydom

_____
Date

Acknowledgements

To my supervisor, Dr Nina Strydom, your expertise, guidance and support made the process of compiling this report most insightful.
To my family, particularly my parents, thank you for your continuous support.
To PSG, thank you for the financial support provided in the form of the bursary programme throughout this process.

Contents

1 Introduction **6**

2 Background Theory **7**
2.1 Portfolio Selection and Mean-Variance Theory under the Markowitz Model 7
2.2 Sample Covariance Matrix 8
2.3 Overview of Estimation Methods 8
2.3.1 Shrinkage Estimation with Ledoit and Wolf's Shrinkage Constant 9
2.3.2 The Moore-Penrose Generalized Inverse 10
2.3.3 The Method of Principal Components 11

3 Application of Estimation Methods **12**
3.1 Shrinkage Estimation with Ledoit and Wolf's Shrinkage Constant 12
3.2 The Moore-Penrose Generalized Inverse 12
3.3 The Method of Principal Components 13

4 Conclusion **16**

Appendix **18**

List of Tables

1 Table of Top 40 Index market-capitalization weights vs weights produced by estimated covariance matrix using different methods 14

List of Figures

1 Line graph of Top 40 Index market-capitalization weights vs weights produced by estimated covariance matrix using different methods 15

1 Introduction

The use of large dimensional covariance matrix estimation in portfolio selection is essential since, generally, in financial markets the more recent data is desirable for inference on future data. This means a limited number of observation points are used for a large number of stocks. The focus of the application of large dimensional covariance matrix estimation is on portfolio selection under the Markowitz model. Under this model it is vital to have a covariance matrix of stock returns that is invertible. This invertible covariance matrix is needed to calculate the portfolio weights.

There are a number of ways to estimate covariance matrices. Use of the conventional estimator, namely the sample covariance matrix, becomes problematic in the large dimensional case. When the number of variables, p, is less than the number of observations, n, an invertible and unbiased sample covariance matrix can be found. However when $n < p$ the sample covariance matrix has unfavourable properties. It contains large amounts of estimation error and the inverse does not exist. Improvements on the sample covariance matrix as estimator for the population covariance matrix had to be made because of the unfavourable properties it poses for the case where $n < p$.

In earlier theory on the estimation of covariance matrices, the large dimensional case was not considered. Numerous methods broke down when the number of observations, n, were equal to or less than the number of variables, p. For example, methods that used either the Wishart or the inverted Wishart distribution for estimation of the covariance matrix failed since the restriction that the degrees of freedom should be strictly greater than the number of variables, p, is violated. This is because the degrees of freedom is $n - 1$ (when using sample data) and since $n < p$, the violation occurs. Thus considering these methods that break down in the large dimensional case, new theory on estimation of covariance matrices developed.

The aim of this report is to give an overview of some of these improvements to the sample covariance matrix as an estimator for the population covariance matrix. Shrinkage estimation represents one of these improvements. Numerous adjustments under shrinkage estimation are found in theory. However, methods under shrinkage estimation that used a loss function that involved an inverted covariance matrix would always fail. Ledoit and Wolf [5] improved on the shrinkage estimation by using a loss function that did not include an inverted covariance matrix. They then developed an optimal solution for the shrinkage estimator of the population covariance matrix.

In this report, under shrinkage estimation focus will be on the method developed by Ledoit and Wolf . Other methods of estimation such as the Moore-Penrose inverse method and the principal component analysis method will also be discussed. Under each of these methods, a detailed explanation on how to construct the covariance matrix is given. An application in portfolio selection is illustrated. With each method, the calculation of the covariance of stock returns from a dataset compiled of stocks listed on the JSE's[1] Top 40 All Share Index is done. After testing if an inverse of the covariance matrix calculated exists, the portfolio weights are calculated and then compared.

In Section 2 the terminology used in industry is explained followed by background theory on the sample covariance matrix and why it breaks down as an estimator. An overview of the different alternative methods of estimation is discussed next. In Section 3 an application illustrates three of these methods of estimation. The report finishes off with the conclusion in Section 4.

[1] Johannesburg Stock Exchange

2 Background Theory

2.1 Portfolio Selection and Mean-Variance Theory under the Markowitz Model

Portfolio selection is the process of selecting a combination of stocks to include in a portfolio while simultaneously considering the risk, returns and other features that can affect individual stocks and the portfolio as a unit. Risk can be affected by interest rates, equity prices (value of stock or portfolio), foreign exchange rates and price fluctuation of commodities. Return of a stock (per monetary unit invested) for time period t[2] is measured as:

$$TR_t = \frac{(P_t - P_{t-1}) + D}{P_{t-1}} \tag{1}$$

where t is the time period of interest, P_t is the closing price of the stock for the time period t, P_{t-1} is the closing price of the stock for time period $t-1$ and D is the dividends for the time period.

The return is the profit made through either the trading of stock or through the dividends received. It is variable in nature and subject to risk. These stock returns can be measured against the stock market return or specific market indices. These indices demonstrate how the market is faring generally. These market indices have several ways in which the stocks that fall under them are weighted. An equal-weighted index equally weights each constituent. Therefore, each stock equally contributes to the return as well as the risk of that index. A value-weighted index weights the stocks per share price of the stocks. A capitalization-weighted index is weighted by market capitalization[3]. The last two types of indices mentioned have the disadvantage of being heavily influenced, in risk and in return, by the large companies. The risk in such indices is not a true reflection of how large on average each constituent's risk is but rather how large on average the larger companies' risk is.

The aim of the mean-variance portfolio theory under the Markowitz model is to minimize the variance within a portfolio (risk associated with that portfolio) subject to certain constraints namely,

$$\sum_{i=1}^{p} w_i = 1 \tag{2}$$

$$\sum_{i=1}^{p} w_i \mu_i = \bar{r} \tag{3}$$

where w_i is the weight associated with stock i, μ_i is the average stock return of stock i and $\bar{r}$ is the average return of the portfolio. In constraint (2) the weights per stock in the portfolio should sum up to 1 (it is possible to get negative weights). Negative weights suggest short-selling the asset. Short-selling is the selling of a stock that is not owned by the seller. The seller borrows the stock and then sells the borrowed stock for which the seller gets credited. The seller, however, has to eventually pay for the borrowed stocks. The seller can make a profit from short-selling if the price at which the shares were sold is higher than the price the seller is expected to pay for them. In constraint (3) the weighted average of the portfolio should be equal to the specified average return wanted, $\bar{r}$. This, collectively, can be referred to as the Markowitz problem. The solution to this problem finds the portfolio weights that minimize portfolio variance for a given value of average return. In order to find this solution, an invertible covariance matrix is required.

[2] t could represent a day, week, month, year etc.

[3] Weight of company using capitalization-weighted index=$\frac{Capital\ Of\ Stock}{Total\ Market\ Capital}$

2.2 Sample Covariance Matrix

The sample covariance matrix, $\mathbf{S}$, is the conventional estimator of the population covariance matrix, $\boldsymbol{\Sigma}$. It can be calculated as follows:

$$\mathbf{S} = \frac{1}{n-1}\mathbf{Y}'(\mathbf{I} - \frac{1}{n}\mathbf{J})\mathbf{Y} \tag{4}$$

where $\mathbf{Y}$ is the data matrix ($n \times p$ matrix), n is the number of observations, p is the number of variables, $\mathbf{I}$ is $n \times n$ identity matrix and $\mathbf{J}$ is an $n \times n$ matrix of $1's$.

An advantage of the sample covariance matrix is that it can easily be constructed using the data matrix $\boldsymbol{Y}$, as seen in equation (4). Other advantages include that it is an unbiased estimator of the population covariance matrix, i.e. $E(\mathbf{S}) = \boldsymbol{\Sigma}$, and the inverse exists when $n > p$.

Under the assumption of normality, the sample covariance matrix, $\boldsymbol{S}$, is the maximum likelihood estimator of the true covariance matrix, $\boldsymbol{\Sigma}$. The maximum likelihood estimator is based solely on the data and does not perform well when the data is small, i.e $n \leq p$. Generally, problems with the sample covariance matrix, $\boldsymbol{S}$, being an estimator arises when $n \leq p$. The rank of $\boldsymbol{S}$ is $n-1$ since it is at most the rank of matrix $(\mathbf{I} - \frac{1}{n}\mathbf{J})$. However, in the case where $n \leq p$, the inverse of $\boldsymbol{S}$ does not exist even if the true population covariance matrix, $\boldsymbol{\Sigma}$, has an inverse.

In this report, the case when $n < p$ will be considered. In portfolio selection this is when the number of stocks, p, is greater than the number of observations or historic data available, n. With $n < p$, the estimation of the covariance matrix becomes a problem because a $p \times p$ covariance matrix with $p + \frac{p^2-p}{2}$ parameters, assuming the covariance matrix is symmetric, needs to be estimated. Since $n < p < p + \frac{p^2-p}{2}$, there are not enough observations to estimate these parameters. To be able to estimate $\boldsymbol{\Sigma}$, some structure on the estimator needs to be imposed. The type of structure is dependent on the problem at hand. With respect to portfolio selection and stock returns, a lower-dimensional factor model is often used to impose structure on an estimator.

2.3 Overview of Estimation Methods

A number of methods exist for the estimation of large dimensional covariance matrices. Stein [10] noted that the sample covariance matrix does not perform well as an estimator of the population covariance matrix, $\boldsymbol{\Sigma}$ when the ratio $\frac{p}{n}$ is large. Therefore, improvements on the sample covariance matrix, $\boldsymbol{S}$, as an estimator of $\boldsymbol{\Sigma}$, had to be made. Shrinkage estimation of covariance matrices is one such method. For $n < p$, when elements of $\boldsymbol{S}$ are calculated, the estimates may be inflated and, therefore, contain estimation error. This is due to the number of observation points available. Shrinkage estimation of $\boldsymbol{\Sigma}$, considers a weighted linear combination of $\boldsymbol{S}$ with some structured matrix, say $\mathbf{B}$. A structured matrix has fewer parameters to estimate compared to the parameters required for estimation of $\boldsymbol{S}$. The contribution of the unstructured $\boldsymbol{S}$ as an estimator of $\boldsymbol{\Sigma}$ is reduced by some percentage, say δ. Thus reducing the influence of estimation error. Consequently, $\delta\%$ of the structured matrix will contribute to the estimator of $\boldsymbol{\Sigma}$. Therefore, the resulting estimator of $\boldsymbol{\Sigma}$, under shrinkage estimation, is $\hat{\boldsymbol{\Sigma}} = (1-\delta)\boldsymbol{S} + \delta\boldsymbol{B}$. The amount by which we shrink this sample covariance matrix is determined by minimizing certain risk functions. Risk functions are defined by taking the expected value of the particular loss functions. Commonly used loss functions include Stein's loss function

$$L_1(\hat{\boldsymbol{\Sigma}}, \boldsymbol{\Sigma}) = tr(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}) - log\left|\hat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\right| - p \tag{5}$$

and the quadratic loss function

$$L_2(\hat{\boldsymbol{\Sigma}}, \boldsymbol{\Sigma}) = tr(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1} - \boldsymbol{I}) \tag{6}$$

where $\boldsymbol{I} : p \times p$ the identity matrix and tr represents the trace of the matrix. It can be seen that estimating the covariance matrix using such loss functions will fail when $n \leq p$, simply because their loss functions require calculation of the inverse of the covariance matrix.

Ledoit and Wolf [4] used an alternative loss function that does not include an inverse of the covariance matrix. This is explained in more detail in Section 2.3.1. In Section 2.3.2, the inverse of the covariance matrix

is estimated using the Moore-Penrose inverse of the sample covariance matrix, $\boldsymbol{S}$. Lastly, in Section 2.3.3, the method of principal components is discussed. This is a method widely used in the financial economics industry.

Another method that could be used for large dimensional covariance matrix estimation is the Bayesian approach, which also has a relation to shrinkage estimation. There is extensive theory available for the estimation of large dimensional covariance using the Bayesian approach [2, 9]. For example, Bai and Shi [2] give a review of some methods of estimation that have been developed to work for the large dimensional case, in addition to the Bayesian approach. This approach will not form part of the methods discussed in this report but it important to mention.

2.3.1 Shrinkage Estimation with Ledoit and Wolf's Shrinkage Constant

In portfolio selection, an invertible covariance matrix needs to be estimated in order to calculate efficient portfolio weights under the Markowitz mean-variance portfolio theory. Ledoit and Wolf get this invertible covariance matrix by shrinking the sample covariance matrix, $\boldsymbol{S}$, to Sharpe's single-index model covariance matrix estimator. The aim of Ledoit and Wolf's paper is to use Sharpe's single-index model to impose structure to the unstructured sample covariance matrix, and determine how much of this structured covariance matrix should be imposed on the estimator.

Sharpe's single-index model defines stock returns at time t as

$$x_{it} = \alpha_i + \beta_i x_{0t} + \varepsilon_{it} \quad for\ i = 1, 2, \ldots, p \tag{7}$$

where ε_{it} are the residuals that are uncorrelated with the market returns x_{0t} and α_i is a constant for asset i and β_i is the factor loading for the market returns for asset i. It is regarded as a single-factor model with the single-factor being the market returns.

The covariance matrix under Sharpe's single-index model shows risk in a systematic way and is defined as

$$\boldsymbol{\Phi} = \sigma_{00}^2 \boldsymbol{\beta}\boldsymbol{\beta}' + \triangle \tag{8}$$

where $\triangle$ is the diagonal matrix of residual variances, $var(\varepsilon_{it}) = \delta_{ii}$, $\boldsymbol{\beta}$ is the vector of slope estimates (factor loadings) and σ_{00}^2 is the variance of market returns. The covariance under Sharpe's index model breaks down risk into two components: $\sigma_{00}^2 \boldsymbol{\beta}\boldsymbol{\beta}'$ is the macroeconomic component representing market influences and $\triangle$ is the microeconomic component that represents the stock-specific random component. It is important to note that Ledoit and Wolf have assumed that $\boldsymbol{\Phi} \neq \boldsymbol{\Sigma}$. The estimate of the covariance matrix under this model is

$$\hat{\boldsymbol{\Phi}} = s_{00}^2 \mathbf{b}\mathbf{b}' + \mathbf{D} \tag{9}$$

where s_{00}^2 is the sample variance of market returns, each element of $\boldsymbol{b}$ is a least squares estimator of the corresponding element in $\boldsymbol{\beta}$ and $\mathbf{D}$ is a diagonal matrix of $\hat{\sigma}_{i,\varepsilon}^2$'s for $i = 1, 2, .., p$ where $\hat{\sigma}_{i,\varepsilon}^2$ is based on the ordinary least squares residuals. This estimate will be used as the shrinkage target matrix. The shrinkage target matrix is the matrix used to impose structure on the estimator of the covariance matrix. The percentage by which the sample covariance matrix is shrunk, i.e. the shrinkage intensity, is explicitly calculated in their paper [4].

Ledoit and Wolf's optimal shrinkage intensity is derived by minimizing the risk function corresponding to the loss function defined by $L_\alpha = \left\| (1-\alpha)\boldsymbol{S} + \alpha\hat{\boldsymbol{\Phi}} - \boldsymbol{\Sigma} \right\|^2$. After minimization of the risk function, the optimal shrinkage intensity is defined as

$$\alpha^* = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p} var(s_{ij}) - cov(\hat{\phi}_{ij}, s_{ij})}{\sum_{i=1}^{p}\sum_{j=1}^{p} var(\hat{\phi}ij - s_{ij}) + (\phi_{ij} - \sigma_{ij})^2} \tag{10}$$

The shrinkage intensity, α^*, is the percentage by which we "shrink" the sample covariance matrix, $\boldsymbol{S}$. It can also be seen as the weight of structure we want to impose on the estimator of the covariance matrix,

thus the percentage of the shrinkage target matrix that will be used in the weighted average calculation of the shrinkage target matrix and the sample covariance matrix.

Consequently, an optimal shrinkage intensity, α^*, is obtained that does not rely on a loss function that involves the inverse of the covariance matrix. Equation (10) converges to the following expression

$$\alpha^* = \frac{\kappa}{n}, \quad \kappa = \frac{(\pi - \rho)}{\gamma}, \tag{11}$$

π represents the estimation error on $\boldsymbol{S}$, ρ measures the covariance between the estimation errors of $\boldsymbol{S}$ and $\hat{\boldsymbol{\Phi}}$ and γ represents the squared difference in value between $\boldsymbol{\Phi}$ and $\boldsymbol{\Sigma}$. The shrinkage intensity placed on the shrinkage target increases with error on $\boldsymbol{S}$ (through parameter π). The intensity decreases when there is a misspecification of the shrinkage target (through the parameter γ). If $\rho > 0$ ($\rho < 0$) then the benefit of the weighted linear combination of $\mathbf{S}$ and $\hat{\boldsymbol{\Phi}}$ is smaller (larger). Using the sample data, values of κ can be estimated by constant $k = \frac{(c-d)}{g}$, assuming that $\gamma > 0$ (cf. [4]). This version of the formula for α^* is only true for the case when shrinking $\mathbf{S}$ towards the covariance estimator matrix of the single-index model, $\hat{\boldsymbol{\Phi}}$, since d is estimated using the market returns data. The value of d will have to be readjusted if a different shrinkage target is used. The weighted linear combination of $\mathbf{S}$ and $\hat{\boldsymbol{\Phi}}$ with optimal shrinkage intensity α^* can be expressed as $\hat{\boldsymbol{\Sigma}}_{L\&W} = (\frac{k}{n})\hat{\boldsymbol{\Phi}} + (1 - \frac{k}{n})\boldsymbol{S}$.

The inverse for $\hat{\boldsymbol{\Sigma}}_{L\&W}$ exists since $\hat{\boldsymbol{\Phi}}$ has an inverse. Using this invertible covariance matrix, optimal portfolio weights can be calculated as follows

$$w = \frac{\boldsymbol{C} - \bar{\boldsymbol{r}}\boldsymbol{B}}{\boldsymbol{AC} - \boldsymbol{B^2}}\boldsymbol{\Sigma^{-1}1} + \frac{\bar{\boldsymbol{r}}\boldsymbol{A} - \boldsymbol{B}}{\boldsymbol{AC} - \boldsymbol{B^2}}\boldsymbol{\Sigma^{-1}\mu} \tag{12}$$

where $\mathbf{A} = \mathbf{1'\Sigma^{-1}1}$, $\boldsymbol{B} = \mathbf{1'\Sigma^{-1}\mu}$, $\boldsymbol{C} = \boldsymbol{\mu'\Sigma^{-1}\mu}$, $\bar{\boldsymbol{r}} = \boldsymbol{w'\mu}$, $\mathbf{1}$ is a conformable vector of ones, $\bar{\boldsymbol{r}}$ is the expected rate of return that is required for the portfolio and $\boldsymbol{\mu}$ is a vector of the average rate of return per asset/stock.

An advantage of Ledoit and Wolf's method is that it can be generalized. Instead of shrinking the sample covariance matrix to the covariance matrix of the single-index model, another shrinkage target matrix can be used. A trade-off between the sample covariance matrix and its estimation error and asymptotic unbiasedness with another estimator matrix that has opposite properties occurs. Another advantage is that the shrinkage intensity is consistently optimal.

2.3.2 The Moore-Penrose Generalized Inverse

A generalized inverse of a matrix $\mathbf{Y} : n \times p$ of arbitrary rank is a matrix $\mathbf{G} : p \times n$ such that for a vector $\mathbf{b}$, $\mathbf{b} = \mathbf{Gx}$ is a solution of $\mathbf{Yb} = \mathbf{x}$ for any vector $\mathbf{x}$ for which the system of equations is consistent. The Moore-Penrose matrix, $\mathbf{M}$, is a generalized inverse, which satisfies the following four conditions:

$$\begin{aligned} (i) \quad & \mathbf{YMY} &&= \mathbf{Y} \\ (ii) \quad & \mathbf{MYM} &&= \mathbf{M} \\ (iii) \quad & (\mathbf{YM})' &&= \mathbf{YM} \\ (iv) \quad & (\mathbf{MY})' &&= \mathbf{MY} \end{aligned} \tag{13}$$

where $'$ represents the transpose of the matrix. This Moore-Penrose matrix, $\mathbf{M}$, is unique for matrix $\mathbf{Y}$.

The concept of the generalized inverse of a matrix was originally documented by Moore [6]. Penrose [8] later developed similar theory where both authors gave the same conditions to what is now known as the Moore-Penrose generalized inverse.

In the article by Pappas et al. [7] it is recommended that for large portfolio applications a generalized inverse (Moore-Penrose inverse) is used when a sample covariance matrix is not invertible, close to being non-invertible (i.e.$det(\mathbf{S}) \approx 0$) or is ill-conditioned. It is further proven in the article that the optimal portfolio weights can be calculated similarly to equation (12) with $\boldsymbol{\Sigma}^{-1}$ replaced by the Moore-Penrose inverse, $\mathbf{M}$.

2.3.3 The Method of Principal Components

The latent factor model with r common factors as defined in the book by Johnson and Wichern[3] is:

$$\boldsymbol{x_j} \;=\; \boldsymbol{\mu} + \boldsymbol{Lf} + \boldsymbol{\varepsilon} \;\; for \quad j = 1, 2, ..., n \;\; and \; \boldsymbol{x_j} : p \times 1, \;\; \boldsymbol{\mu} : p \times 1, \quad \mathbf{L} : p \times r \quad , \boldsymbol{f} : r \times 1, \quad \boldsymbol{\varepsilon} : p \times 1 \tag{14}$$

where x_{ij} is the variable (stock return) i at time j, μ_i is the mean of the variable i, ε_i is the i^{th} variable specific factor ($i = 1, 2, ..., p$), f_t is the t^{th} common factor ($t = 1, 2, ...r$) and l_{it} is the loading of the i^{th} variable on the t^{th} factor. The factor model in (14) is then referred to as an orthogonal factor model if the random vectors $\boldsymbol{f}$ and $\boldsymbol{\varepsilon}$ satisfy the following conditions:

1) $\boldsymbol{f}$ *and* $\boldsymbol{\varepsilon}$ *are independent* $\Rightarrow$ $COV(\boldsymbol{\varepsilon}, \boldsymbol{f'}) = \mathbf{0}$
2) $E(\boldsymbol{f}) = \mathbf{0}$, $COV(\boldsymbol{f}, \boldsymbol{f'}) = \boldsymbol{I}$ *where* $\boldsymbol{I}$ *is an* $r \times r$ *identity matrix.*
3) $E(\boldsymbol{\varepsilon}) = \mathbf{0}$, $COV(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}) = \boldsymbol{\Psi}$ *where* $\boldsymbol{\Psi}$ *is a diagonal matrix.*

The covariance matrix for this factor model is then,

$$\boldsymbol{\Sigma} = \boldsymbol{LL'} + \boldsymbol{\Psi} \;\; for \; r < p, \; where \; \boldsymbol{\Psi} \; is \; a \; diagonal \; matrix, \tag{15}$$

thus the covariance matrix can be broken up into a linear combination of a structure of communality (through $\boldsymbol{LL'}$) and a variable specific structure ($\boldsymbol{\Psi}$).
The principal component factor analysis of the sample covariance matrix, $\boldsymbol{S}$, is a special case of the factor analysis of the orthogonal factor model. Considering $\boldsymbol{S} = \sum_{i=1}^{p} \hat{\lambda}_i^2 \boldsymbol{\hat{e}_i \hat{e}_i'}$, the spectral decomposition of the sample covariance matrix and using the ordered eigenvalue-eigenvector pairs ($\hat{\lambda}_1$, $\boldsymbol{\hat{e}_1}$), ..., ($\hat{\lambda}_p$, $\boldsymbol{\hat{e}_p}$) of the sample covariance matrix $\boldsymbol{S}$, the estimated matrix of factor loadings is calculated as:

$$\boldsymbol{\hat{L}} = \left[\sqrt{\hat{\lambda}_1} \times \hat{\boldsymbol{e}}_1 \vdots \sqrt{\hat{\lambda}_2} \times \hat{\boldsymbol{e}}_2 \vdots \; \ldots \; \vdots \sqrt{\hat{\lambda}_r} \times \hat{\boldsymbol{e}}_r \right] \tag{16}$$

The estimated matrix for $\boldsymbol{\Psi}$ (also referred to as the matrix of specific variance) is calculated by the diagonal elements of the matrix $\boldsymbol{S - \hat{L}\hat{L}'}$, i.e.:

$$\boldsymbol{\hat{\Psi}} = diag(\boldsymbol{S - \hat{L}\hat{L}'}) \tag{17}$$

thus

$$\boldsymbol{\hat{\Psi}} = \begin{bmatrix} \hat{\psi_1} & 0 & \cdots & 0 \\ 0 & \hat{\psi_2} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \ldots & 0 & \hat{\psi_p} \end{bmatrix} \; where \; \hat{\psi_i} = sii - \sum_{t=1}^{r} l_{it}^2$$

The estimated covariance matrix is then calculated as:

$$\boldsymbol{\hat{\Sigma}_{PCM}} = \boldsymbol{\hat{L}\hat{L}'} + \boldsymbol{\hat{\Psi}} \;\; for \; r < p \tag{18}$$

The number of factors, r, needs to be estimated. One way to estimate the value for r is to look at the eigenvalues of $\boldsymbol{S}$ and then the number of factors (components) is taken to be the point at which the remainder of the eigenvalues tend to be about the same size and are relatively small. Another way is to increase the number of factors retained by accumulating the following formula:

$$\begin{matrix} proportion \; of \\ total \; sample \; variance \\ due \; to \; i^{th} \\ factor \end{matrix} \quad = \quad \frac{\hat{\lambda}_i}{\sum_{i=1}^{p} \hat{\lambda}_i} \tag{19}$$

until a "suitable proportion" of total sample variance ($\sum_{i=1}^{p} s_{ii} = \sum_{i=1}^{p} \hat{\lambda}_i$) has been explained. It is important to note that for the sample covariance matrix $\boldsymbol{S}$, $\hat{\lambda}_1 > \hat{\lambda}_2 > \hat{\lambda}_3 > \ldots \hat{\lambda}_{p-1} > \hat{\lambda}_p$.

An advantage of this method is that it can decrease the number of variables, p, to a value less than the number of observations, n. Consequently, an invertible covariance matrix can be estimated. The estimated covariance matrix under this method is also easy to compute. A disadvantage of this method is the difficulty in interpreting what these factors represent. Another disadvantage is that the selection of the number of factors r is subjective, which indirectly influences how the portfolio will be selected (through the calculation of portfolio weights).

3 Application of Estimation Methods

For purposes of application, the stock returns are selected from the 40 companies listed under the Johannesburg Stock Exchange's All Share Index (also referred to as the All Share Top 40 Index)[4] for the 36 working days before 23 of April 2016 (exclusive). Therefore $n = 36$ and $p = 43$ (Note that there are 40 companies listed but three of the companies break up into two separate entities, i.e. $p = 43$) . These companies are ranked by full-market capitalization. This index is market-capitalization weighted. These weights can be found in Table 1. This index also forms a benchmark to measure South Africa's stock market. Under each method developed in the previous section, an explanation of how the portfolio weights are calculated using that method is given. A comparison of the portfolio weights of the current method with those of the methods preceding that method is included under each section. A visual representation of the different weights in the form of a line graph is given in Figure 1.

Note that the specified average return, $\bar{r}$, is equal to 35. This was calculated by getting the average of the stock returns over the 36 working days.

3.1 Shrinkage Estimation with Ledoit and Wolf's Shrinkage Constant

The aim of Ledoit and Wolf's estimator for the covariance matrix was to impose structure to the sample covariance matrix, $\boldsymbol{S}$. Using SAS/IML^{TM}software[5] together with Ledoit and Wolf's method, an invertible estimated covariance matrix was found (cf. Appendix). The shrinkage intensity, α^*, was calculated using the estimated κ, $k = \frac{(c-d)}{g}$, derived from equation (11). The α^* was found to be 0.3148191 (≈32%). This means about 32 of the shrinkage target (the structured matrix) and 68%of the sample covariance matrix contributed to the estimated covariance matrix, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{L\&W}}$. Using the inverse of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{L\&W}}$, the weights were then calculated using equation (12). These weights are given in Table 1. It was found that some of the weights were negative, suggesting short-selling of stocks. Usually all weights under a market-capitalization weighted portfolio, like the All Share Top40 Index, are positive. Therefore, the negative weights obtained create a new perspective of how the index can be weighted.

3.2 The Moore-Penrose Generalized Inverse

The Moore-Penrose generalized inverse is calculated with the sample covariance matrix,$\boldsymbol{S}$. The sample covariance matrix, $\boldsymbol{S}$, is calculated using equation (4). The Moore-Penrose inverse is calculated using a function in SAS/IML^{TM} [6] (cf. Appendix). The weights calculated using this inverse are given in Table 1. Generally, it can be seen that where Ledoit and Wolf's method had negative weights, the Moore-Penrose method also gives negative weights. However, there are a few stocks in which the signs of the weights between

[4]Source: INET BFA accessed: 22/04/2016

[5]The data analysis for this report was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA.

[6]The SAS^{TM} software by default gives the Moore-Penrose generalized inverse when the function *ginv(.)* is used, thus satisfying conditions (13).

the two methods do not correlate, namely: APN, BHP, GRT, INL, MNP and OML. As stated previously, negative weights introduce the opportunity of short-selling stocks within a portfolio.

3.3 The Method of Principal Components

Using the sample covariance matrix $\boldsymbol{S}$, the eigenvalues and their corresponding normalized eigenvectors are calculated using SAS/IML^{TM} (cf. Appendix). Using the criterion (19), the number of factors were found to be three (i.e. $r = 3$). The three factors explained $0.8301995 (\approx 83$ of the total sample variance. Using equation (16) and (17) to estimate the covariance matrix under principal components analysis, i.e $\hat{\boldsymbol{L}}$ and $\hat{\boldsymbol{\Psi}}$, the estimated covariance matrix is obtained:

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{PCM}} = \hat{\boldsymbol{L}}\hat{\boldsymbol{L}}' + \hat{\boldsymbol{\Psi}} \tag{20}$$

$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{PCM}}$ is also found to be invertible. Using the inverse of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{PCM}}$ the portfolio weights were calculated using equation (12). These weights are given in Table 1. The weights given suggest fewer stocks to short-sell compared to the previous methods. Ledoit and Wolf's method and the Moore-Penrose generalized inverse method suggest to short-sell 21 and 23 stocks, respectively. The method of principal components suggests only 18 of the stocks should be short-sold.

Company - JSE Code	MC [7]	L&W [8]	MP [9]	PCM [10]
Anglo American plc - AGL	0.0253	0.0042703	0.0151125	0.0146484
Anglo American Plat Ltd - AMS	0.00360	-0.000608	-0.019323	0.0005753
Anglogold Ashanti - ANG	0.0136	0.0034621	0.0428483	0.0062537
Aspen Pharmacare Hldgs Ltd - APN	0.01760	0.0270044	-0.011731	0.0225694
Brait SE - BAT	0.00910	-0.002389	-0.026428	0.0149818
Barclays Africa Grp Ltd -BGA	0.008	-0.020119	-0.051203	-0.010971
BHP Billiton plc - BIL	0.05810	0.0267154	-0.001907	0.0225354
British American Tob plc - BTI	0.04330	0.0012627	0.0095342	0.0175517
Bidvest Ltd - BVT	0.01980	-0.03029	-0.058957	-0.007105
Capital&Counties Prop plc - CCO	0.0036	-0.12231	-0.00549	-0.082198
Compagnie Fin Richemont - CFR	0.0810	-0.049072	-0.068339	-0.041722
Capitec Bank Hldgs Ltd - CPI	0.0049	-0.002389	-0.026428	0.0149818
Discovery Ltd - DSY	0.0065	0.0166048	0.0374232	0.043187
Fortress Inc Fund Ltd A - FFA	0.0020	0.2892937	0.1926394	0.1991857
Fortress Inc Fund Ltd B - FFB	0.0035	0.21188	0.3292963	0.3409548
Firstrand Ltd - FSR	0.0243	-0.102561	-0.444776	-0.13838
Growthpoint Prop Ltd - GRT	0.0106	-0.191929	0.1565488	-0.062845
Investec Ltd - INL	0.0046	0.0308881	-0.060352	0.0193429
Investec plc - INP	0.0106	0.0626047	0.1388342	0.0380417
Intu Properties plc - ITU	0.0103	-0.074528	-0.172851	-0.079496
Mediclinic Int plc - MEI	0.0116	-0.008497	-0.074491	0.0086333
Mondi Ltd - MND	0.0056	-0.005384	-0.197504	-0.017689
Mondi plc - MNP	0.0172	-0.008038	0.2334727	-0.018148
Mr Price Group Ltd - MRP	0.0068	-0.019507	-0.014595	-0.006696
MTN Group Ltd - MTN	0.0393	-0.015053	-0.066062	0.0001248
Nedbank Group Ltd - NED	0.0063	-0.0468	-0.013054	-0.044923
Naspers Ltd -NPN	0.1466	0.0022677	0.0039263	0.0022086
Netcare Limited - NTC	0.0087	0.4098525	0.329101	0.2554168
Old Mutual plc - OML	0.0321	0.0877328	-0.127568	0.0164473
Redefine Properties Ltd - RDF	0.0087	0.361329	0.1601653	0.2650805
Reinet Investments S.C.A - REI	0.0077	0.1119173	0.0866355	0.1665838
Remgro Ltd - REM	0.0199	0.0072972	0.1025899	-0.001678
RMB Holdings Ltd - RMH	0.0067	-0.2073	-0.160008	-0.170705
Rand Merchant Inv Hldgs Ltd - RMI	0.0048	0.0225929	0.3686101	-0.010255
SABMiller plc - SAB	0.1458	-0.003493	-0.005866	-0.003262
Standard Bank Group Ltd - SBK	0.0264	0.0402402	0.1801614	0.0140749
Shoprite Holdings Ltd - SHP	0.0125	0.022401	0.0397192	0.0151718
Sanlam Limited - SLM	0.0199	0.1126068	0.2269389	0.1079946
Steinhoff Int Hldgs N.V. - SNH	0.0409	-0.037861	-0.02124	-0.022421
Sasol Limited - SOL	0.0405	-0.009143	-0.024752	-0.002607
Tiger Brands Ltd - TBS	0.0087	0.0234629	0.0280735	0.0198354
Vodacom Group Ltd - VOD	0.0087	0.1252443	0.0570971	0.1088403
Woolworths Holdings Ltd - WHL	0.0141	-0.043661	-0.085803	-0.014122

Table 1: Table of Top 40 Index market-capitalization weights vs weights produced by estimated covariance matrix using different methods

[7] Market-capitalization weights from Johannesburg Stock Exchange (JSE) as at 31/03/2016, Source:FTSE Group
[8] Weights produced by estimated covariance matrix using Ledoit and Wolf method
[9] Weights produced by estimated covariance matrix using Moore-Penrose Generalised Inverse method
[10] Weights produced by estimated covariance matrix using Method of Principal Components

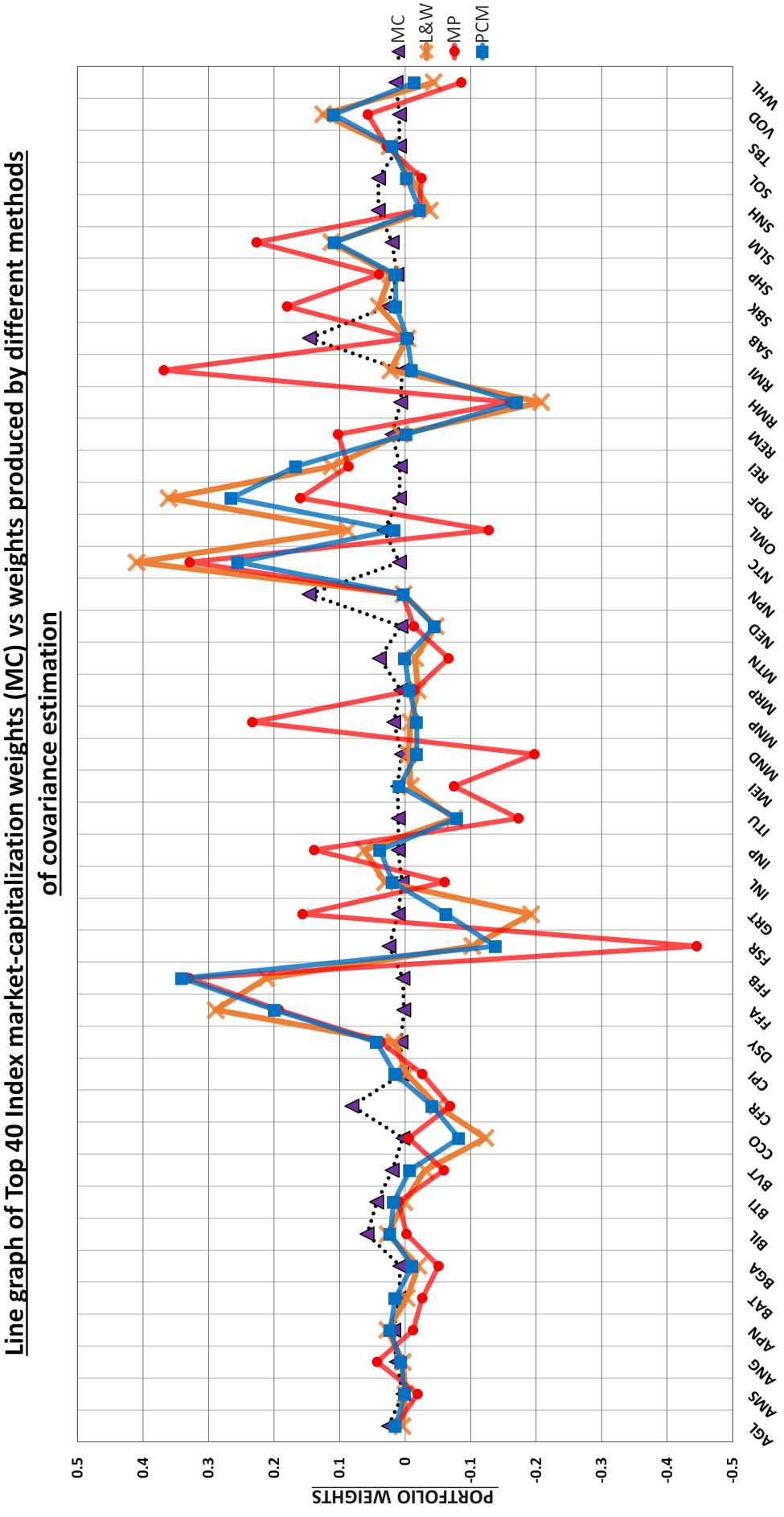


Figure 1: Line graph of Top 40 Index market-capitalization weights vs weights produced by estimated covariance matrix using different methods

4 Conclusion

Different methods of estimating large dimensional covariance matrices for the case where the number of observations, n, is fewer than the number of variables, p, are explored. For each method, the construction of the covariance matrix is discussed. For purposes of application, it is important that these covariance matrices are invertible. For example, in portfolio selection the inverse of the covariance matrix is needed for the calculation of portfolio weights under the Markowitz model. This forms part of the application section of this report. With the use of one dataset of stock returns over a certain time period, portfolio weights under each method of estimation were calculated and compared.

When comparing the three methods, fewer stocks agree on the short-selling of the stock, and more stocks are inconsistent throughout the three methods. Note that consistency is measured with respect to all methods suggesting either short-selling or buying the stock. Even with the consistent stocks, the weights given for the different methods are not similar or approximately equal. This is due to the different methods of calculating these large dimensional covariance matrices, showing the importance of choosing the best method. Many other methods, such as the Bayesian approach (cf. [2, 9]), are available in literature but are not discussed in this report of limited scope.

When comparing the market-capitalization weights with those calculated by the three methods, it can be seen that when the volatility of the stock is considered the opportunity for short-selling is introduced. An advantage of strictly positive values of the market-capitalization weights is that there is no gamble taken through the short-selling of a stock. However, the market-capitalization weights have the disadvantage that the volatility of the stock, i.e. the day-to-day variability of the stock return, is not considered in calculation of the weights. Under the Markowitz model (the model used by all three methods in the calculation of portfolio weights) the aim is to find the portfolio weights such that the variability is minimized.

In addition to application in portfolio selection, large dimensional covariances have applications in financial risk management. An example would be in the calculation of value at risk (VaR) models. These models give a measure of the risk in a particular investment instrument. The article by Alexander and Leigh [1] further examines the use of covariance matrix estimation in VaR models. A further study could be done to verify if methods developed for large dimensional covariance matrix estimation in portfolio selection can be used in VaR model calculations.

References

[1] C.O Alexander and C.T Leigh. On the covariance matrices used in value at risk models. *The Journal of Derivatives*, 4(3):50–62, 1997.

[2] J Bai and S Shi. Estimating high dimensional covariance matrices and its applications. Technical report, Columbia University, 2011.

[3] R.A Johnson and D.W Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall Inc, 1982.

[4] O Ledoit and M Wolf. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, 10(5):603–621, 2003.

[5] O Ledoit and M Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004.

[6] E. H. Moore. On the Reciprocal of the General Algebraic Matrix. *Bulletin of the American Mathematical Society*, 26:394–395, 1920.

[7] D Pappas, K Kiriakopoulos, and G Kaimakamis. Optimal Portfolio Selection with Singular Covariance Matrix. In *International Mathematical Forum*, volume 5, pages 2305–2318, 2010.

[8] R Penrose. A Generalized Inverse for Matrices. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 51, pages 406–413. Cambridge Univ Press, 1955.

[9] M Pourahmadi. *High-Dimensional Covariance Estimation*. Probability and Statistics. John Wiley & Sons, Inc, 2013.

[10] C Stein. Inadmissibility of the Usual Estimator for the Mean of a Multivariate Normal Distribution. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 197–206, Berkeley, Calif., 1956. University of California Press.

Appendix

```
/*Using Ledoit and Wolf Method to find Estimated Covariance Matrix*/

proc iml;
use sasuser.Top40;
read all into X1;
x=X1[,2:44];
n=nrow(x);
p=ncol(x);
meanx=mean(x);
mean2x=(1/p)*sum(meanx);
meannx=J(n,p,1)#meanx;
x=x-meannx;
xmkt=mean(x`)`;
xxmkt=x||xmkt;
sample=((n-1)/n)*cov(xxmkt);
covmkt=sample[1:p,p+1];
varmkt=sample[p+1,p+1];
prior = (covmkt*covmkt`)#(1/varmkt);
diagonalprior= diag(prior);
priorwithoutdiagonal=prior - diagonalprior;
diagsample=diag(sample[1:p,1:p]);/*D from equation (12)*/
newprior=priorwithoutdiagonal+diagsample; /*Shrinkage target*/
ranknewprior=round(trace(ginv(newprior)*newprior));
/* to prove that shrinkage target is invertible (therefore full rank)*/

print ranknewprior,
'thus shrinkage target is invertible since full rank (rank(shrinkage target)=p)';

sampleminusnewprior=sample[1:p,1:p]-newprior;
g=norm(sampleminusnewprior,'frobenius')**2;

y=x##2;
yty=y`*y;/*used to calculate first part of calculating c*/
sumsample2=sum(sample[1:p,1:p]##2);/*second part of calculating c*/
rauhat=(1/n)*sum(yty)-sumsample2;/*c*/

sumy2=sum(y##2);/*first part of calculating Ddiag*/
Ddiag=(1/n)*sumy2-sum(diag(sample[1:p,1:p])##2);/*Ddiag*/
z=x#xmkt;/*z*/
v1=(1/n)*y`*z-covmkt#sample[1:p,1:p];/*v1*/
v1c=v1#(covmkt`);
sumv1c=sum(v1c);
Doffdiag1=sumv1c/varmkt-sum(diag(v1)*covmkt)/varmkt;/*Doff1*/
v3=(1/n)*z`*z-varmkt*sample[1:p,1:p];/*v3*/
Doffdiag3=sum(v3#(covmkt*covmkt`))/(varmkt**2)-sum(diag(v3)#(covmkt##2))/(varmkt**2);/*Doff3*/
Doffdiag=2*Doffdiag1-Doffdiag3;/*roff*/
d=Ddiag+Doffdiag;/*r*/
k=(rauhat-d)/g;/*used to compute the shrinkage intensity*/
shrinkage=max(0,min(1,k/n));
estimatedcovariance=shrinkage*newprior+(1-shrinkage)*sample[1:p,1:p];
print 'Shrinkage intensity', shrinkage;
```

```
print 'Estimated covariance using Ledoit and Wolf method', estimatedcovariance;

rankE=round(trace(ginv(estimatedcovariance)*estimatedcovariance));
/*to prove that estimated covariance is invertible (thus full rank)*/

print rankE,
'thus estimated covariance is invertible since full rank (rank(estimated covariance)=p)';

/* calculating portfolio weights for Ledoit and Wolf method*/

ones=J(p,1,1);
invEstimatedCovariance=inv(estimatedcovariance);
A=ones`*invEstimatedCovariance*ones;
rbar=35;/*what is selected as the expected rate of return per day expected from portfolio*/
mu=meanx`;
B=ones`*invEstimatedCovariance*mu;
C=mu`*invEstimatedCovariance*mu;
wpart1=((C-rbar*B)/(A*C-B**2))*invEstimatedCovariance*ones;
wpart2=((rbar*A-B)/(A*C-B**2))*invEstimatedCovariance*mu;
weights=wpart1+wpart2;
print weights;


/*Using Moore-Penrose Generalised Inverse to find Estimated Covariance Matrix*/


use sasuser.Top40;
read all into Y1;
Y=Y1[,2:44];
meanY=mean(Y);
n1=nrow(Y);
p1=ncol(Y);
ones=J(n1,1,1);
deviationssquared=(Y-(1/n1)*ones*ones`*Y)`*(Y-(1/n1)*ones*ones`*Y);
SampleCov=(1/(n1-1))*deviationssquared;/*This would represent "Y" in equation (16)*/
M=ginv(SampleCov);/*Moore-Penrose Generalized Inverse of Y
calculated by default by function ginv() in SAS*/

/* calculating portfolio weights for Moore-Penrose method*/

ones2=J(p1,1,1);
A=ones2`*M*ones2;
rbar=35;/*The expected rate of return per day expected from portfolio*/
mu=meanY`;
n=nrow(mu);
B=ones2`*M*mu;
C=mu`*M*mu;
wpart11=((C-rbar*B)/(A*C-B**2))*M*ones2;
wpart21=((rbar*A-B)/(A*C-B**2))*M*mu;
weights1=wpart11+wpart21;
print weights1;
```

```
/*Using Method of Principal Components to find Estimated Covariance Matrix*/


use sasuser.Top40;
read all into Z1;
Z=Z1[,2:44];
meanZ=mean(Z);
n2=nrow(Z);
p2=ncol(Z);
ones=J(n2,1,1);
deviationssquared=(Z-(1/n2)*ones*ones`*Z)`*(Z-(1/n2)*ones*ones`*Z);
S=(1/(n2-1))*deviationssquared;
call eigen(values,vectors,S);/*function to get the eigenvalues and corresponding
eigenvectors of S*/

r=rank(values); /*ranks eigenvalues in ascending order*/
prop1=values[1]/sum(values); /*using equation (19)*/
prop2=values[2]/sum(values); /*...*/
prop3=values[3]/sum(values); /*...*/
prop4=values[4]/sum(values); /*...*/
prop5=values[5]/sum(values); /*...*/
cumulative=prop1+prop2+prop3; /*shown to equal 0.8301995*/
print prop1 prop2 prop3 prop4 prop5, cumulative;
L=(values[1:3]##(1/2))`#vectors[,1:3]; /*using equation (19)*/
EstimatedCov=L*L`+diag(S-L*L`); /*Estimated Covariance using method of Principal
Components (Using L calculated above and equation (20)) */
print L;
rankEC=round(trace(ginv(EstimatedCov)*EstimatedCov)); /*to prove that covariance is
invertible so that weights can be calculated the same as that in previous 2 methods*/
print rankEC ,'thus covariance is full rank (implies invertibility)';

/* calculating portfolio weights for Method of Principal Components method*/

InvEstimatedCov=inv(EstimatedCov);
ones2=J(p2,1,1);
A=ones2`*InvEstimatedCov*ones2;
rbar=35;/*The expected rate of return per day expected from portfolio*/
mu=meanZ`;
n=nrow(mu);
B=ones2`*InvEstimatedCov*mu;
C=mu`*InvEstimatedCov*mu;
wpart12=((C-rbar*B)/(A*C-B**2))*InvEstimatedCov*ones2;
wpart22=((rbar*A-B)/(A*C-B**2))*InvEstimatedCov*mu;
weights2=wpart12+wpart22;
print weights2;

/*Data matrix used to make line graph of portfolio weights in excel*/
use sasuser.Mcweights;
read all into A1;
MC=A1`;
Portfolioweights=MC||weights||weights1||weights2;
cn={'MC' 'L$W' 'MP' 'PCM'};
rn={'Anglo American plc - AGL' 'Anglo American Plat Ltd - AMS''Anglogold Ashanti - ANG'
```

```
'Aspen Pharmacare Hldgs Ltd - APN''Brait SE - BAT''Barclays Africa Grp Ltd -BGA'
'BHP Billiton plc - BIL''British American Tob plc - BTI' 'Bidvest Ltd - BVT'
'Capital&Counties Prop plc - CCO''Compagnie Fin Richemont - CFR''Capitec Bank Hldgs Ltd - CPI'
'Discovery Ltd - DSY''Fortress Inc Fund Ltd A - FFA''Fortress Inc Fund Ltd B - FFB'
'Firstrand Ltd - FSR''Growthpoint Prop Ltd - GRT''Investec Ltd - INL'
'Investec plc - INP''Intu Properties plc - ITU''Mediclinic Int plc - MEI'
'Mondi Ltd - MND''Mondi plc - MNP''Mr Price Group Ltd - MRP''MTN Group Ltd - MTN'
'Nedbank Group Ltd - NED''Naspers Ltd -N- - NPN''Netcare Limited - NTC''Old Mutual plc - OML'
'Redefine Properties Ltd - RDF''Reinet Investments S.C.A - REI''Remgro Ltd - REM'
'RMB Holdings Ltd - RMH''Rand Merchant Inv Hldgs Ltd - RMI''SABMiller plc - SAB'
'Standard Bank Group Ltd - SBK''Shoprite Holdings Ltd - SHP''Sanlam Limited - SLM'
'Steinhoff Int Hldgs N.V. - SNH''Sasol Limited - SOL''Tiger Brands Ltd - TBS'
'Vodacom Group Ltd - VOD''Woolworths Holdings Ltd - WHL' };

print Portfolioweights[colname=cn rowname=rn];

create Portfolioweights from Portfolioweights[rowname=rn colname=cn];
append from Portfolioweights [rowname=rn];
quit;

/*"Portfolioweights" data matrix was exported to excel where a line graph was compiled*/
```

Heavy tail distributions for claims data

Kuselo Ntsaluba 13108434

WST795 Research Report

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

Supervisor: Dr Fabris-Rotelli, I.

Department of Statistics, University of Pretoria

2 November 2016 (final)

Abstract

In this report heavy tail distributions for claims data are discussed. Predicting the event of observing very large or even extreme claims is done using the upper tail of the such distributions. The presence of a heavy upper tail suggests high risk and should be modeled as accurately as possible. Methods to describe/measure these extremes such as extreme value theory (EVT) and techniques to detect the presence of heavy tail occurrence of large claims are discussed and studied. The theory is applied to simulated data consisting of two underlying distributions. The results indicate that the splicing point can be estimated with these methods.

Declaration

I, *Kuselo Ntsaluba,* 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.

_____
Kuselo Ntsika Ntsaluba

_____
Dr Inger Fabris-Rotelli

_____
Date

Acknowledgments

I would like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a post graduate bursary.

To my supervisor, Dr. Inger Fabris-Rotelli, thank you for your time, guidance and invaluable input during the process of conducting this report.

To my family and in particular my brother, I would like to express my gratitude for your input.

Contents

1 Introduction **6**

2 Literature Review **7**

3 Background Theory **8**
3.1 The Claim Model 8
3.2 Tail Heaviness 8
3.3 Methods to detect and measure tail heaviness 9
3.4 Extreme Value Theory 13
3.5 Splicing 24

4 Application **24**
4.1 Unknown splicing point 35

5 Conclusion **44**

Appendix **47**

1 Introduction

Often we have few observations (or claims) from the upper tail of a distribution, however the occurrence of large claims needs to be modeled accurately for prediction purposes. When modeling claims data, we are often interested in two processes: 1) Describing claim sizes and, 2) claim frequency where the distributions for the two random variables of the respective processes are studied independently. In practice the normality assumption is often used due to the fact that it is easy to apply and most often used for inferential purposes. However when we are interested in the occurrence of extreme events (events resulting in large claims) the use of the normal distribution is no longer a reliable representation.

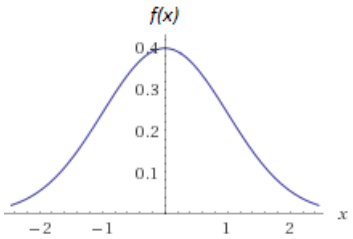


Figure 1: Standard normal density

From Figure 1 which is a normal distribution with mean $\mu = 0$ used as a loss distribution, however there is one problem amongst others with this representation of a loss distribution. The range of the claim amounts is $x \in (-\infty, \infty)$ which is incorrect since it is not possible to have a negative claim. A better (but not perfect) representation of an ideal loss distribution would be as described in Figure 2.

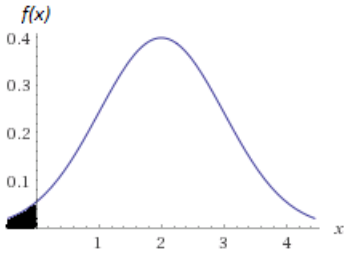


Figure 2: Standard normal density with shifted mean (μ=2)

Figure 2 is the same as the normal distribution used in Figure 1 except with $\mu > 0$. In Figure 2 there is a small area where range of the claim amounts (x) is negative but this becomes negligible as the number of claims tends to infinity ($N \to \infty$) so that the variance becomes small and the mean remains larger than 0. Therefore one can argue that the distribution in Figure 2 can be used as a loss distribution. Now consider the following representation of a loss distribution in Figure 2.

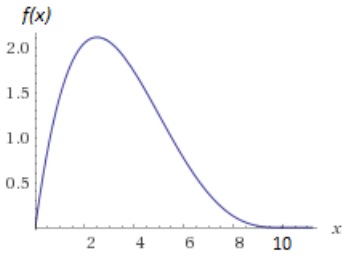


Figure 3: Beta distribution

Figure 3 presents an ideal loss distribution simply because range of claim amounts is $x \in (0, \infty)$, the distribution is positively skewed, indicating a higher probability of small claims, and the upper tail not too heavy which is a more realistic representation of a loss distribution.

In this report we will cover theory on how to measure heaviness of the tail of distributions, as well as extreme value theory, which includes distributions of extreme events such as large maxima, and the use of exponential and Pareto QQ-plots as well as mean excess values to predict the occurrence of a heavy tail or extreme event. The report will conclude with an application of the techniques studied.

2 Literature Review

With the main objectives stated above, the literature review will be conducted with the focus being on extreme value distributions and their domain of attraction as well as positively skewed distributions. A brief overview of the literature will be discussed below with structure being as follows:

- A literature review on modeling claims data.
- Then the focus will move to methods to measure tail heaviness.
- Then finally we give a brief background on research conducted on extreme value distributions and claim arrival of a portfolio of policies.

Literature on modeling claims data

In insurance the prediction of the event of a claim is of great importance. In [12] a hierarchical model is proposed of three components, corresponding to the frequency, type as well as severity of a claim. The first model is a negative binomial regression model used for the assessment of claim frequency. It also turns out that driver age, vehicle age, gender and claims discount are important variables for predicting the event of a claim. More methods for the modeling of claims data are discussed in [16, 27, 17].

Measuring tail heaviness

Different methods of measuring tail heaviness which deal with whether a distribution is exponentially bounded are introduced and discussed in [21] where exponentially bounded implies light-tailed. Further methods for measuring heavy tails are discussed in [11, 9, 20, 2, 24, 5, 1].

Extreme value distributions and claim arrival

It is widely acknowledged in the literature that there is a need for the consideration of heavy tailed distributions. In [22] the distribution of the maximum order statistic is considered, the reason for this is believed to be that a heavy tail is normally associated with a large number of extreme observations, which in insurance would be large claims. The convergence of the distribution function after normalization is also considered together with its possible limits.

In most of the literature on claims arrival an allowance is made for single claims to arrive at a time. In practice however a portfolio of policies presents the reality of several claims arriving within the same time period from different risks where these risks may be dependent in some cases [14].

In [4] a similar observation of the maximum order statistic was made and in addition methods related to the upper right tail of claim frequency or claim size distribution will be considered. Methods such as the Hill estimator, exponential and Pareto QQ-plots are discussed therein where the Hill estimator is a series of estimators that are considered to be unbiased.

Conclusion

The brief literature review above serves as motivation for this research report as we look into extreme value theory, tail heaviness and more methods of measuring tail heaviness including those already stated.

3 Background Theory

3.1 The Claim Model

Let the aggregate claim amount (claim model) be $S(t) = \sum_{i=1}^{N(t)} X_i$ where claim number process $\{N(t) : t \geq 0\}$ is a counting process satisfying the following three conditions with positive parameter λ:

- $N(0) = 0$ (no claims occur at time zero)
- $N(t)$ has independent increments
- $N(t)$ has stationary increments, i.e. $N(t) - N(s) \sim \text{Poisson}(\lambda(t-s))$, $\forall s < t$

Therefore $\{N(t) : t \geq 0\}$ is a Poisson process and $\{S(t) : t \geq 0\}$ is compound Poisson process [21].

The sequence of claim sizes $\{X_t : t \in \mathbb{N}\}$ are assumed to be independent and identically distributed random variables. It is often assumed that $\{N(t) : t \geq 0\}$ and $\{X_t : t \in \mathbb{N}\}$ are independent so that they can be studied independently. The practical advantage of this assumption is that factors affecting claim sizes and claim numbers may be different. Consider motor insurance as an example. A long spell of bad weather may have an effect on the claim numbers but only a slight effect on the distribution claim sizes. Conversely, inflation may have an effect on the cost of repairing a car, thus on the distribution of the claim sizes, but only a slight effect on the distribution of claim numbers. Now the following example is given to show a basic illustration of the claim model.

Example 1. Consider individual claim sizes $X \sim \text{Pareto}(\alpha, \beta)$ with probability density function:

$$f_X(x) = \frac{\alpha\beta^\alpha}{(\beta + x)^{\alpha+1}}, x > 0.$$

if we have annual aggregate claim numbers from a group of general insurance policies having a compound Poisson distribution with parameter λ. Then it follows that:

$$S(t) = \sum_{i=1}^{N(t)} X_i \sim \text{Comp.Poisson}(\lambda t, F_X(x)) \text{ i.e. } N(t) \sim \text{Poisson}(\lambda t)$$

where $F_X(x)$ is the distribution function of individual claim sizes.

3.2 Tail Heaviness

According to [9] the class of well-behaved distributions is defined as those distributions F which satisfy $1 - F_X(x) \leq ce^{-ax}$ where c, a are positive valued and this is true $\forall x \geq 0$, $F_X(x)$ is the distribution function of the claim sizes. To see why the inequality already stated makes sense, consider large claim size x^* arbitrarily chosen. Then we have

$$P(X > x^*) = 1 - P(X \leq x^*) = 1 - F_X(x^*) \leq ce^{-ax^*}.$$

It is desired that the probability stated above tend to zero as claim size becomes too large. If a particular loss distribution does satisfy this inequality, then

$$\lim_{x^* \to \infty} P(X > x^*) = \lim_{x^* \to \infty} (1 - F_X(x^*)) \leq \lim_{x^* \to \infty} ce^{-ax^*} = 0$$
$$\implies \lim_{x^* \to \infty} P(X > x^*) = 0.$$

Using this reasoning, it can be argued that a loss distribution is well defined if $1 - F_X(x) \leq ce^{-ax}$. We illustrate this concept with the following example from [24].

Example 2. Consider random variable $X \sim \text{Weibull}(\gamma, \beta)$ with probability density function :

$$F_X(x) = \begin{cases} 1 - e^{-(\frac{x}{\gamma})^{\beta}}, & y > 0 \\ 0 & \text{elsewhere.} \end{cases}$$

If we suppose that $0 < \beta < 1$ and further suppose that there exists a $\delta > 0$ such that

$$1 - F_X(x) \leq e^{-\delta x},$$

which implies $\left(\frac{x^{\beta-1}}{\gamma^{\beta}}\right) \geq \delta$. Thus if we have $0 < \beta < 1$, then

$$\lim_{x\to\infty} \delta \leq \lim_{x\to\infty} \left(\frac{x^{\beta-1}}{\gamma^{\beta}}\right) = 0$$

which is clearly a contradiction to the fact that there exists a value $\delta > 0$. This implies there does not exist a $\delta > 0$ and $c > 0$ such that $1 - F_X(x) \leq ce^{-\delta x}$. It can therefore be said that the Weibull distribution with parameter values for $0 < \beta < 1$ is not well-behaved since there does not exist a $\delta > 0$ and $c > 0$ such that $1 - F_X(x) \leq ce^{-\delta x}$.

3.3 Methods to detect and measure tail heaviness

Firstly consider two exponential distributions with the parameters α_1 and α_2 such that $0 < \alpha_2 < \alpha_1 < \infty$. The two probability density functions are given in Figure 4. The exp(3.5) is considered to have a heavier tail than the exp(2.5).

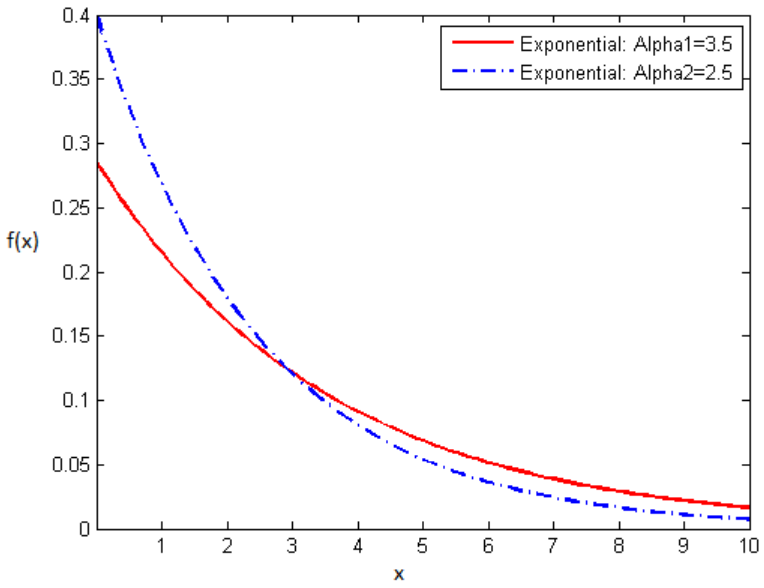


Figure 4: Comparison of two exponential probability density functions exp(α_1) and exp(α_2)

Let X be a non-negative and absolutely continuous random variable. Now we consider methods for detecting tail heaviness.

Method 1:

A distribution $F_X(x)$ is said to be heavy tailed if $\int_{\mathbb{R}} e^{\lambda x} F_X(x)dx = \infty, \forall \lambda \in \mathbb{R}$ and light tailed if $\int_{\mathbb{R}} e^{\lambda x} F_X(x)dx < \infty$ for some values $\lambda > 0$ [11].

Method 2:

An absolutely continuous distribution with distribution function, $F_X(x)$, is said to have an exponentially bounded tail if there exists a $b > 0$ and $c > 0$ such that

$$\bar{F}_X(x) = 1 - F_X(x) \leq be^{-cx}, \forall x > 0$$

A distribution with an exponentially bounded tail is said to have a light tail [21].

Method 3:

Consider two different distributions, $F(x)$ and $G(x)$ and suppose further we can find a very x^* situated in the upper tail such that $f(x^*) = g(x^*)$. Now consider a case where $\bar{F}(x^*) > \bar{G}(x^*)$ while $f(x^*) = g(x^*)$, implying that for a given x^* the weight in the upper tail of $F(x)$ is heavier than the weight of the upper tail of $G(x)$. Therefore the following implies that $F(x)$ has a heavier upper tail than $G(x)$:

$$\frac{1}{\bar{G}(x^*)} > \frac{1}{\bar{F}(x^*)} \Rightarrow \frac{f(x^*)}{\bar{G}(x^*)} > \frac{f(x^*)}{\bar{F}(x^*)} \Rightarrow \frac{g(x^*)}{\bar{G}(x^*)} > \frac{f(x^*)}{\bar{F}(x^*)} \text{ (since } f(x^*) = g(x^*) \text{)}$$

However this comparison does not hold $\forall x \geq x^*$, but only proven to hold at x^*[24].

Method 4:

An absolutely continuous distribution function, $F_X(x)$, is said to have a heavy-tail if $\forall t > 0$ the moment generating function $M_X(t)$ is finite i.e. $M_X(t) < \infty$ [21, 11].

Method 5:

Consider the following definitions:

Definition 3. The hazard function [2] of random variable X, with distribution function $F_X(x)$ is defined as

$$h_X(x) = -\ln(1 - F_X(x)) = -\ln\left(\bar{F}_X(x)\right) \tag{1}$$

and the hazard rate function of random variable X, defined for $F_X(t) < 1$, is given by

$$h_X^*(t) = \frac{f_X(t)}{1 - F_X(t)}. \tag{2}$$

Therefore it follows that if probability density function $f_X(x)$ is continuous, then $h_X(x)$ is differentiable and thus $\frac{d}{dx}(h_X(x)) = h_X^*(x)$ [from 1 and 2].

Definition 4. Residual hazard rate distribution function [20], $F_t(x)$, is defined by

$$F_t(x) = \frac{F_X(t+x) - F_X(t)}{1 - F_X(t)} = \frac{P(t < X \leq x + t)}{P(X > t)} = P(X \leq x + t | X > t) = P(X - t \leq x | X > t)$$

for $F_X(t) < 1$. The mean residual hazard function [6, 5, 8] is given by:

$$\mu_{F_t} = E[X - t | X > t] = \frac{\int_t^\infty (\bar{F}_X(x))dx}{1 - F_X(t)}, F_X(t) < 1.$$

Therefore mean residual function gives average amount by which the random variable X exceeds the value t. This is also known as the exceedance function.

Now consider the following:

$$h_X^*(t) = \frac{f_X(t)}{1 - F_X(t)} = \frac{-\frac{d}{dt}(1 - F_X(t))}{1 - F_X(t)} = -\frac{d}{dt}(\ln(1 - F_X(t))) \text{ [from 1 and 2]}$$

$$\Longrightarrow 1 - F_X(t) = e^{-\int_0^t h_X^*(y)dy}$$

$$\Longrightarrow \bar{F}_X(t) = e^{-\int_0^t h_X^*(y)dy}.$$

Therefore it follows that,

$$\begin{aligned}
\bar{F}_t(t) &= 1 - \frac{F_X(t+x) - F_X(t)}{1 - F_X(t)} \\
&= \frac{\bar{F}_X(t+x)}{1 - F_X(t)} \\
&= \frac{e^{-\int_0^{t+x} h_X^*(y)dy}}{e^{-\int_0^t h_X^*(y)dy}} \\
&= e^{-\int_t^{t+x} h_X^*(y)dy}.
\end{aligned}$$

Now the ratio of $\bar{F}_X(x+t)$ to $\bar{F}_X(t)$ is increasing if the hazard rate function $h_X^*(t) = \frac{f_X(t)}{1-F_X(t)}$ is decreasing [5]. Therefore mean residual hazard function defined above can be written as [21, 2]:

$$\mu_{F_t} = E[X - t | X > t] = \int_0^\infty e^{-\int_0^x h_X^*(y+t)dy} dx.$$

This implies that if hazard rate function is decreasing then mean residual function is increasing. The mean residual hazard function can be used to compare two distributions where the one with a heavier tail will have a mean residual hazard function that increases at a faster rate.

Method 6:

Let $\gamma_F = \limsup_{x\to\infty} \frac{h_X(x)}{x}$. If $\gamma_F = 0$, then $F_X(x)$ is heavy-tailed [21].

Method 7:

If a distribution function $F_X(x)$ defined on $\mathbb{R}^+$ has the following property

$$\lim_{x\to\infty} \frac{1 - F^{*2}(x)}{1 - F(x)} = 2,$$

where $F_X^{*2}(x)$ is the two-fold convolution[1] of $F_X(x)$, then the distribution is said to be a member of the subexponential class represented by S.

Now each distribution $F \in S$ is heavy-tailed [21]. Furthermore consider two distribution functions $F(x)$ and $G(x)$ where we know that the two distributions are tail equivalent and $F \in S$, it follows that $G \in S$ [1]. The example that follows will use the distribution in Figure 4 to make an illustration of method 3.

Example 5. Now from Figure 4 we have the following density and distribution functions:

$$f(x) = \frac{1}{\alpha_1} e^{-\left(\frac{x}{\alpha_1}\right)}, x \geq 0$$

$$F(x) = 1 - e^{-\left(\frac{x}{\alpha_1}\right)}, x \geq 0$$

and

$$g(x) = \frac{1}{\alpha_2} e^{-\left(\frac{x}{\alpha_2}\right)}, x \geq 0$$

[1] The n-fold convolution of $F_X(x)$ is defined as $F_X^{*n}(x) = F_X^{*n-1}(x) * F_X^*(x)$

$$G(x) = 1 - e^{-\left(\frac{x}{\alpha_2}\right)}, x \geq 0.$$

From Figure 4 it is clear that we can find a value x^* such that $f(x^*) = g(x^*)$ which is given by:

$$x^* = \frac{\alpha_2 \alpha_1 \ln\left(\frac{\alpha_2}{\alpha_1}\right)}{\alpha_2 - \alpha_1}$$

now we have,

$$\frac{1}{\alpha_2} > \frac{1}{\alpha_1} \Rightarrow \frac{-x^*}{\alpha_2} < \frac{-x^*}{\alpha_1} \Rightarrow e^{\frac{-x^*}{\alpha_2}} < e^{\frac{-x^*}{\alpha_1}} \Rightarrow \bar{G}(x^*) < \bar{F}(x^*) \Longrightarrow \frac{1}{\bar{G}(x^*)} > \frac{1}{\bar{F}(x^*)}, x^* > 0$$

$$\Rightarrow \frac{g(x^*)}{\bar{G}(x^*)} > \frac{f(x^*)}{\bar{F}(x^*)}$$

It therefore follows from Method 3 that $F(x)$ has a heavier tail than $G(x)$ proven to hold from the point x^* only.

Example 6. If we have $G(x) = 1 - e^{-\alpha_2 x}, x \geq 0$ then it follows that

$$\int_{\mathbb{R}} e^{\lambda x} G(x)\, dx = \int_{\mathbb{R}} e^{\lambda x} - e^{x(\lambda - \alpha_2)} dx = \left[\frac{1}{\lambda} e^{\lambda x}\right]_{\mathbb{R}} - \left[\frac{1}{\lambda - \alpha_2} e^{(\lambda - \alpha_2)x}\right]_{\mathbb{R}}, \lambda > 0$$

now since $\lambda > \lambda - \alpha_2 \Rightarrow e^{\lambda} > e^{\lambda - \alpha_2}$, it follows that

$$\int_{\mathbb{R}} e^{\lambda x} G(x)\, dx < \infty$$

by Method 1 $G(x)$ is light-tailed.

Example 7. Consider $X \sim \exp(\alpha_1)$ therefore X has moment generating function,

$$M_X(t) = \frac{\alpha_1}{\alpha_1 - t}, t < \alpha < \infty$$

therefore since $t > 0$

$$\Rightarrow \frac{\alpha_1}{\alpha_1 - t} < \infty.$$

So by Method 4, $F(x)$ is heavy-tailed.

Example 8. Consider the survival function of distribution $G(x)$, given by

$$\bar{G}(x) = 1 - G(x) = e^{-\alpha_2 x}$$

since $X \sim \exp(\alpha_2)$. Now if $b, c \in \mathbb{R}$ are chosen such that $c = \alpha_2$ and $b > c > 0$ then

$$\bar{G}(x) = e^{-\alpha_2 x} \leq b e^{-cx} \Longrightarrow \bar{G}(x) \leq b e^{-cx}, \forall x > 0$$

So by Method 2 $G(x)$ is exponentially bounded since $\exists\ b, c > 0$ such that $\bar{G}(x) \leq b e^{-cx}$, therefore implying that $G(x)$ is light-tailed.

Example 9. We have $h_X(x) = -\ln(1 - F_X(x))$, where $X \sim \exp(\alpha_2)$, now

$$\begin{aligned}
\gamma_F &= \limsup_{x\to\infty}\left(\frac{h_X(x)}{x}\right) \\
&= \limsup_{x\to\infty}\left(\frac{\frac{d}{dx}h_X(x)}{1}\right)[L'Hospital] \\
&= \limsup_{x\to\infty}\left(\frac{h_X^*(x)}{1}\right) \\
&= \limsup_{x\to\infty}(0) \\
&= 0
\end{aligned}$$

Therefore by Method 6, this is inductive of a heavy-tailed distribution.

Example 10. Consider $\lim_{x\to\infty}\left(\frac{1-G_X^{*2}(x)}{1-G_X(x)}\right)$, where $G_X^{*2}(x)$ is the two-fold convolution of $G_X(x)$.

$$\Rightarrow \lim_{x\to\infty}\frac{e^{-\alpha_2 x}(1+\alpha_2 x)}{e^{-\alpha_2 x}} = \lim_{x\to\infty}(1+\alpha_2 x) = \infty$$

So $G(x)$ is not a member of the subexponential class. $G(x)$ has been shown to be exponentially bounded and therefore lighted-tailed.

Example 11. Consider mean residual hazard rate functions of $F(x)$ and $G(x)$ given by

$$\mu_{F_t} = E[X-t|X>t] = \frac{\int_t^\infty(\bar{F}_X(x))dx}{1-F(t)} = \frac{1}{\alpha_1}$$

and

$$\mu_{G_t} = E[X-t|X>t] = \frac{\int_t^\infty(\bar{G}_X(x))dx}{1-G(t)} = \frac{1}{\alpha_2}$$

from [21] for an exponential distribution, now since $0<\alpha_2<\alpha_1<\infty$, $F(x)$ has a heavier tail than $G(x)$. It therefore follows that mean residual hazard function of $F(x)$ increases at a faster rate than mean residual hazard function of $G(x)$, so $F(x)$ has a heavier tail.

3.4 Extreme Value Theory

We normally associate a heavy tail with a large number of extreme observations i.e. large claims. This is why it is necessary to also look at extreme value theory.

Consider the distribution function of the maximum order statistic $X_{n:n}$ given by: $P(X_1\le x,...,X_n\le x) = P(X_{n:n}\le x) = F^n(x)$, where $X_1,\ldots,X_N$ are i.i.d and $F_X(x)$ is the distribution of each X_i [18].

Sometimes the distribution function stated above is difficult to obtain. To overcome this issue consider convergence of $\frac{(X_{n:n}-b_n)}{a_n}$ as $n\to\infty$ where $a_n, b_n\in\mathbb{R}$, sequence of real numbers otherwise known as normalising constants. The theorem that follows states that $X_{n:n}$, after being normalized, converges in distribution to only one of a possible three distributions [10, 13].

Theorem 12. *If $\exists$ sequence of real numbers $a_n, b_n > 0$ such that*

$$\lim_{n\to\infty} P\left(\frac{(X_{n:n}-b_n)}{a_n}\le x\right) = F^n(a_n x+b_n) = G(x), \text{ say}$$

where $F^n(x) = P(X_{n:n}\le x)$ and then $G(x)$ is a non-degenerate distribution function and belongs to one of the following 3 types:

1. Frechet: $\Phi_\alpha(x) = e^{\left(-x^{-\alpha}\right)}, (x,\alpha>0)$
2. Weibull: $\Phi_\alpha(x) = e^{(-(-x^\alpha))}, (x<0,\alpha>0)$

3. Gumbel: $\Phi_\alpha(x) = e^{\left(-e^{-x}\right)}, x \in \mathbb{R}$

This theorem is known as Fisher-Tippett-Gnedenko theorem [10, 13].

Remark 13. The three distributions above are particular cases (where $\alpha = \frac{1}{\beta}$ and first order moment does not exist if $\beta > 1$) of Generalized Extreme Value (GEV) distribution, where the distribution function is given by [10, 13]

$$G_\beta(x) = \begin{cases} e^{-(1+\beta x)^{-\left(\frac{1}{\beta}\right)}} & \text{for } 1+\beta x > 0, \beta \neq 0 \\ e^{\left(-e^{-x}\right)} & \text{for } x \in \mathbb{R}, \beta = 0 \end{cases}$$

where β is the tail index or extreme value index which is related to the tail weight of the distribution. If the result stated in the above theorem holds for a particular distribution F it is said that F belongs to the maximum-domain of attraction of the distribution function G_β which is denoted by $F \in D(G_\beta)$.

Consider Figure 5 with distributions where their tails are shown from lightest to heaviest where exponential is the lightest and Pareto the heaviest.

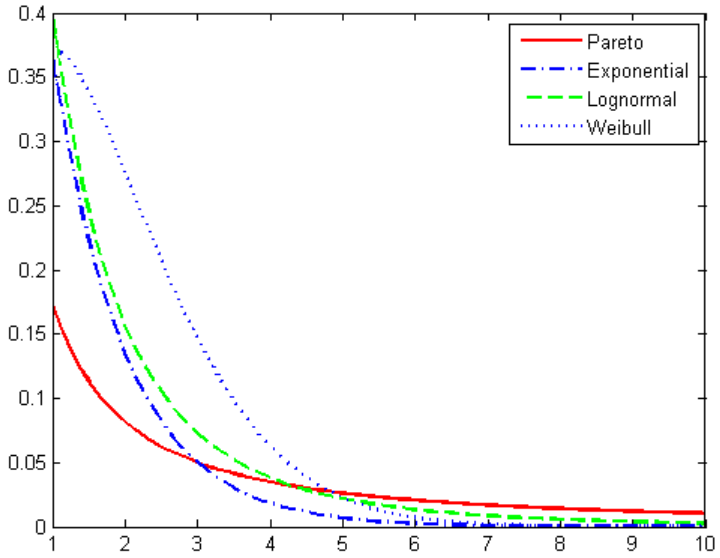


Figure 5: Distributions from "light" to "heavy" tailed

In light of this we consider more methods to measure tail heaviness.

Method 8 (Exponential QQ-plot and mean excess values):

Consider the quantile function defined as

$$Q(p) = \inf\{x : F(x) \geq p\} \text{ for } 0 < p < 1.$$

Now using the following relationship

$$Q(p) = F^{-1}(p) \text{ where } F(x) = 1 - e^{-x\alpha}$$

it follows that

$$F(x) = p \Rightarrow x = \frac{-1}{\alpha}\log(1-p) \Rightarrow F^{-1}(p) = \frac{-1}{\alpha}\log(1-p)$$

where $F(x) = p$ was solved for x in terms of p. We thus obtain

$$Q(p) = \frac{-1}{\alpha}\log(1-p) \text{ for } 0 < p < 1$$

which is known as the exponential quantile function where $p = \frac{j}{n+1}, j = 1, \ldots, n$. The non-parametric estimator of $Q\left(\frac{j}{n+1}\right)$ is given by $X_{j:n}$. Therefore we have exponential quantile plot given by the points:

$$\left(-\log\left(1 - \frac{j}{n+1}\right); X_{j:n}\right) \text{ for } j = 1, \ldots, n.$$

Now, if the exponential fit is linear then the data is exponential. However if the tail of the data distribution is heavier than that of the exponential then exponential QQ-plot will be convex.

Alternatively mean excess values,

$$e_{k:n} = \frac{1}{k}\left[\sum_{i=1}^{k} X_{n-i+1:n}\right] - X_{n-k:n}$$

of the exponential QQ-plot where $\left(-\log\left(\frac{k+1}{n+1}\right); X_{n-k:n}\right)$ is the right anchor point can be used. The mean excess plot is given by the points:

$$(X_{n-k:n}; e_{k:n}).$$

If the mean excess values stay horizontal with increasing k then data is exponential otherwise we have an indication of a heavy tail [4].

Method 9 (Pareto QQ-plot):

Recall the quantile function defined in method 8, now using the relationship

$$Q(p) = G^{-1}(p) \text{ where } G(x) = 1 - \frac{1}{x^{\lambda}}$$

note that $G(x)$ is the distribution function from a basic Pareto distribution. It follows that

$$G(x) = p \Rightarrow x = (1-p)^{\frac{-1}{\lambda}} \Rightarrow G^{-1}(p) = (1-p)^{\frac{-1}{\lambda}}$$

so that

$$\log Q(p) = \frac{-1}{\lambda}\log(1-p) \text{ for } 0 < p < 1.$$

Now similar to Method 8, the non-parametric estimator of $\log\left(Q\left(\frac{j}{n+1}\right)\right)$ is given by $X_{j:n}$ where $p = \frac{j}{n+1}, j = 1, \ldots, n$. The Pareto QQ-plot is given by the points:

$$\left(-\log\left(1 - \frac{j}{n+1}\right), \log(X_{j:n})\right) \text{ for } j = 1, \ldots, n.$$

If the Pareto QQ-plot is linear then the data is Pareto. However, if the Pareto QQ-plot is concave then the tail of the data distribution is less heavy than that of the Pareto.

Alternatively mean excess values (Hill's estimator 1975) [4],

$$H_{k:n} = \frac{1}{k}\left(\sum_{i=1}^{k} \log(X_{n-i+1:n})\right) - \log(X_{n-k:n})$$

of the Pareto QQ-plot where $\left(-\log\left(\frac{k+1}{n+1}\right); \log(X_{n-k:n})\right)$ is the right anchor point can be used. If the $H_{k:n}$ value remains horizontal with decreasing k then the Pareto fit is appropriate. Note also that if $H_{k:n}$ values decreases with decreasing k then the tail of the data distribution is lighter than Pareto. The Hill plot is given by the points [4]:

$$(\log(X_{n-k:n}); H_{k:n}).$$

We now present an example to illustrate Method 8 and 9.

Example 14. For this particular example 200 observations were simulated from the lognormal with parameters 0 and 1, Weibull with parameters 0.5 and 2, Pareto with parameters 2.5 and 1 and exponential with parameter 1 distributions. The tail heaviness of the respective distributions were then compared using method 8. This example is done in-order to demonstrate how the results produced by the exponential QQ-plot and the mean excess plot differ when the underlining distribution changes for the data generated. For the exponential data generated we have the exponential QQ-plot in Figure 6 followed by its mean excess plot in Figure 7.

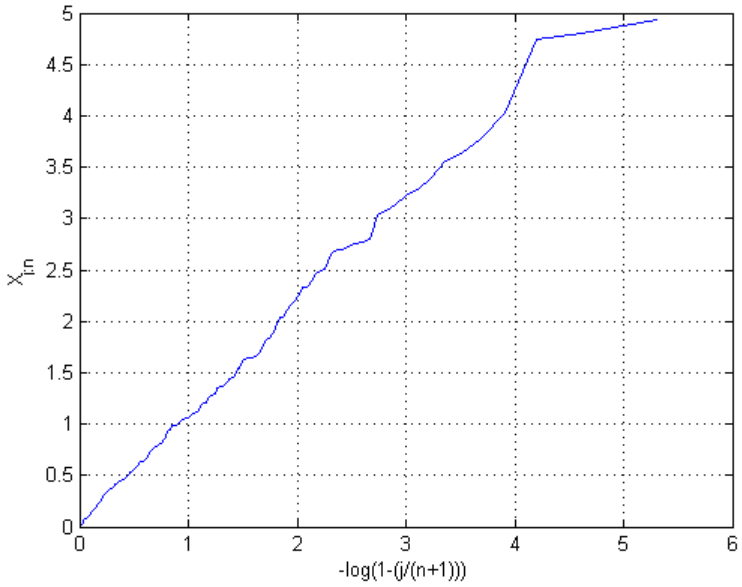


Figure 6: Exponential QQ-plot for simulated exp(1)

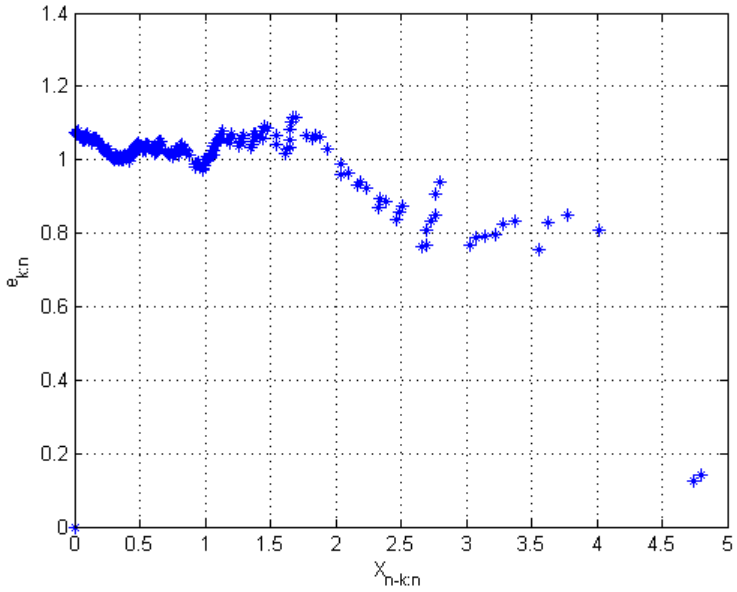


Figure 7: Mean excess plot for simulated exp(1)

Now using method 8, since exponential QQ-plot is linear the data is considered exponential and the mean excess value plot stays horizontal with increasing k, therefore the data is considered exponential as it should be.

For the Weibull data generated we have the exponential QQ-plot in Figure 8 followed by its mean excess plot in Figure 9.

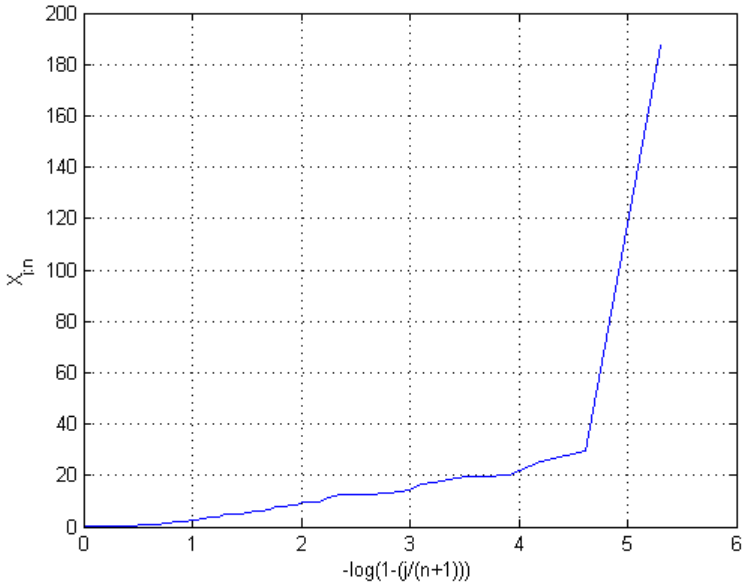


Figure 8: Exponential QQ-plot for simulated Weibull$(0.5, 2)$

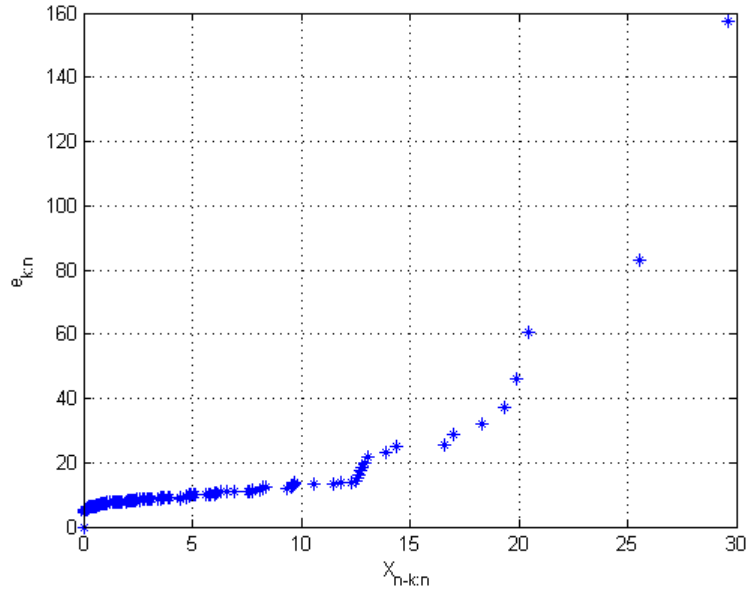


Figure 9: Mean excess plot for simulated Weibull$(0.5, 2)$

Now using method 8 since we see convex exponential QQ-plot in Figure 8 for the Weibull data generated,

therefore the tail of the Weibull data is heavier than the exponential distribution. The mean excess value plot does not stay horizontal with increasing k, which is an indication of a heavy tail.

For the lognormal data generated we have the exponential QQ-plot in Figure 10 followed by its mean excess plot in Figure 11.

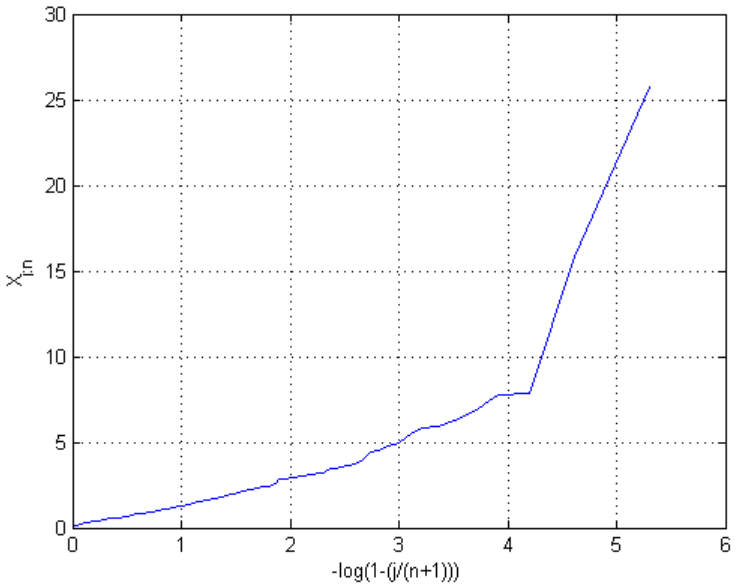


Figure 10: Exponential QQ-plot for simulated logNormal$(0, 1)$

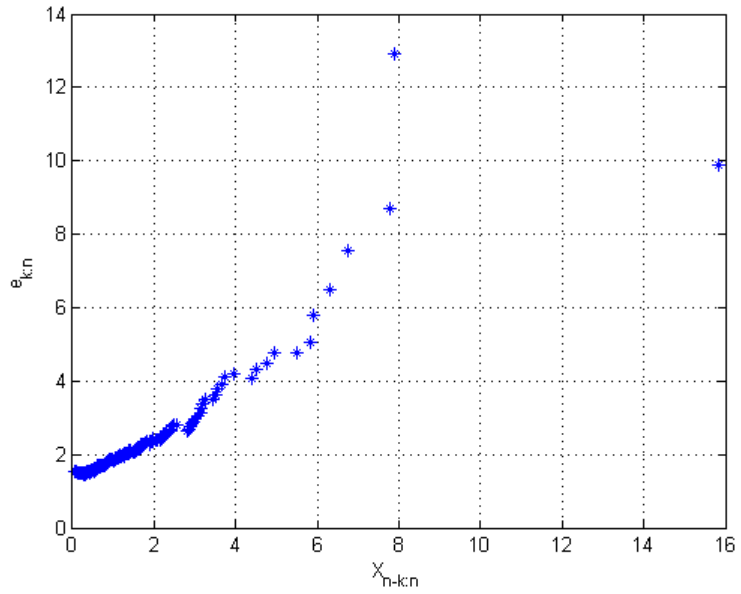


Figure 11: Mean excess plot for simulated logNormal$(0, 1)$

Similar to the Weibull data, using method 8 we have a convex exponential QQ-plot in Figure 10 for the

lognormal data generated therefore the tail of the lognormal data is heavier than the exponential distribution. The mean excess value plot does not stay horizontal with increasing k, which is an indication of a heavy tail.

For the Pareto data generated we have the exponential QQ-plot in Figure 12 followed by its mean excess plot in Figure 13.

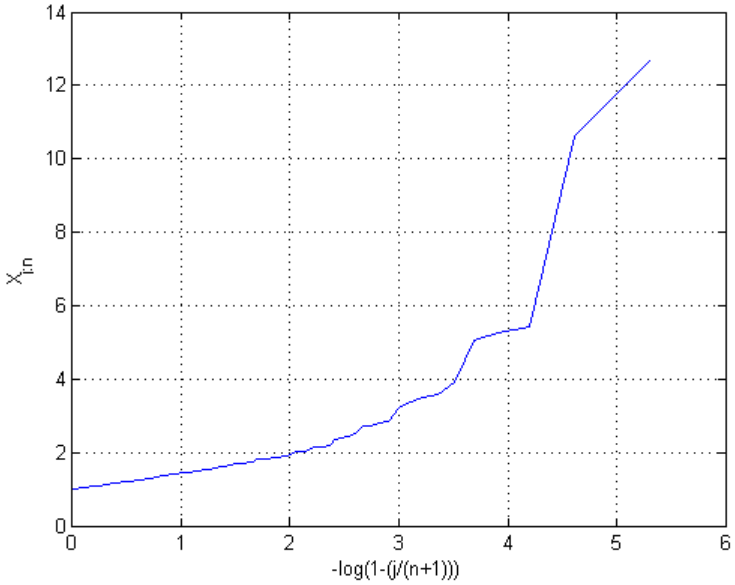


Figure 12: Exponential QQ-plot for simulated Pareto$(2.5, 1)$

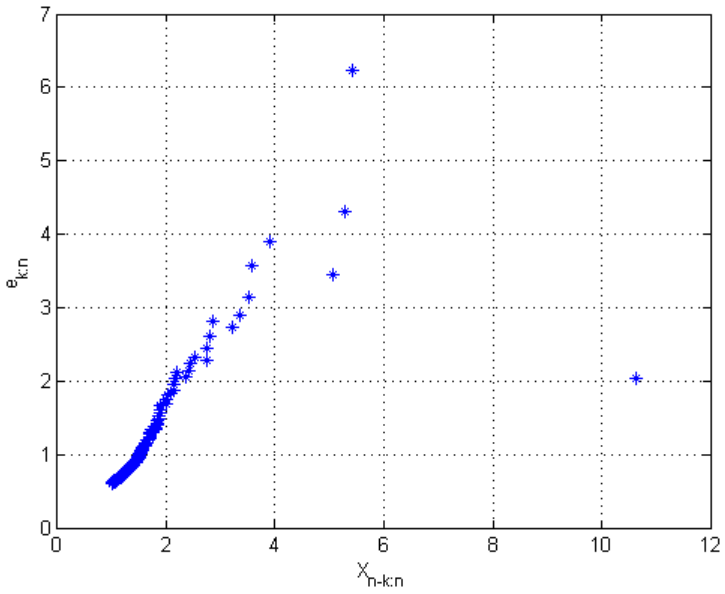


Figure 13: mean excess plot for simulated Pareto$(2.5, 1)$

Similar to the Weibull and lognormal data, using method 8 we have a convex exponential QQ-plot in Figure 12 for the Pareto data generated therefore the tail of the Pareto data is heavier than the exponential

distribution. The mean excess values do not stay horizontal with increasing k, which is an indication of a heavy tail.

Example 15. For this example, using the same simulated data from example 14, the tail heaviness of the respective distributions was then compared using method 9. For the Pareto data generated we have the Pareto QQ-plot in Figure 14 followed by its Hill plot in Figure 15.

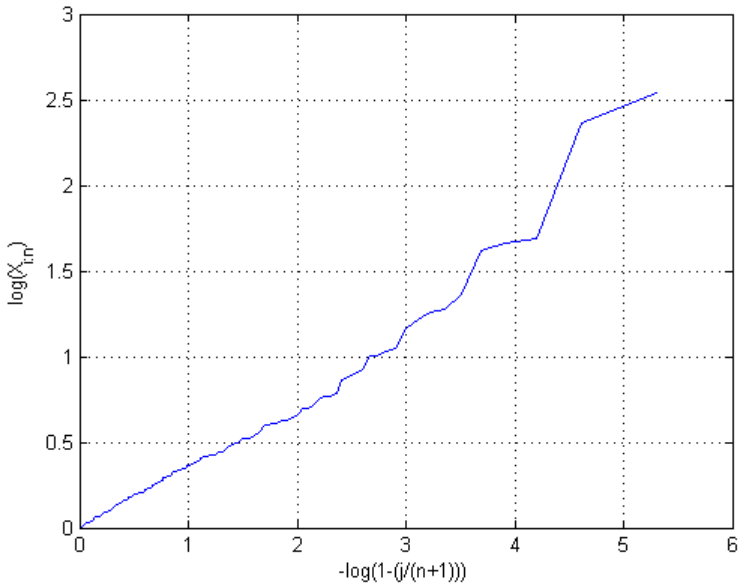


Figure 14: Pareto QQ-plot for simulated Pareto$(2.5, 1)$

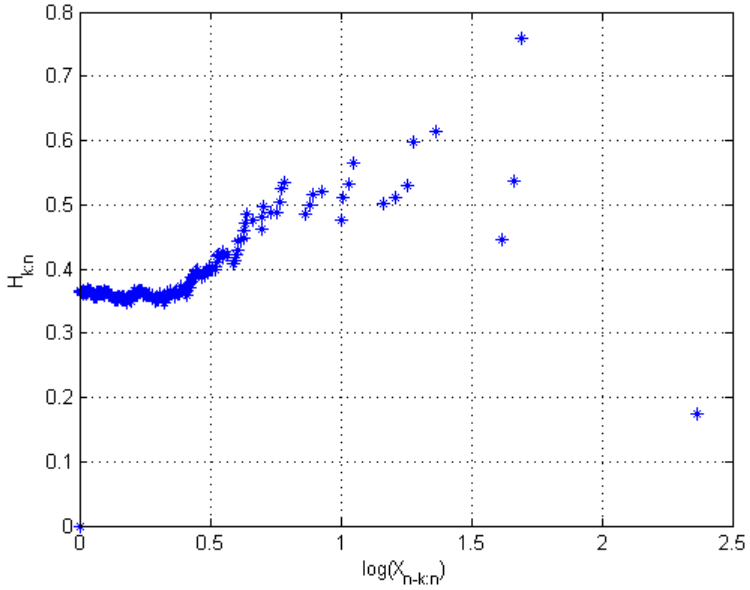


Figure 15: Hill plot for simulated Pareto$(2.5, 1)$

Now using method 9, since Pareto QQ-plot in Figure 14 is linear the data is Pareto and we have that the $H_{k:n}$ in Figure 15 stays relatively horizontal with decreasing k it follows that the Pareto fit is appropriate.

Now for the lognormal data generated we have the Pareto QQ-plot in Figure 16 followed by its Hill plot in Figure 17.

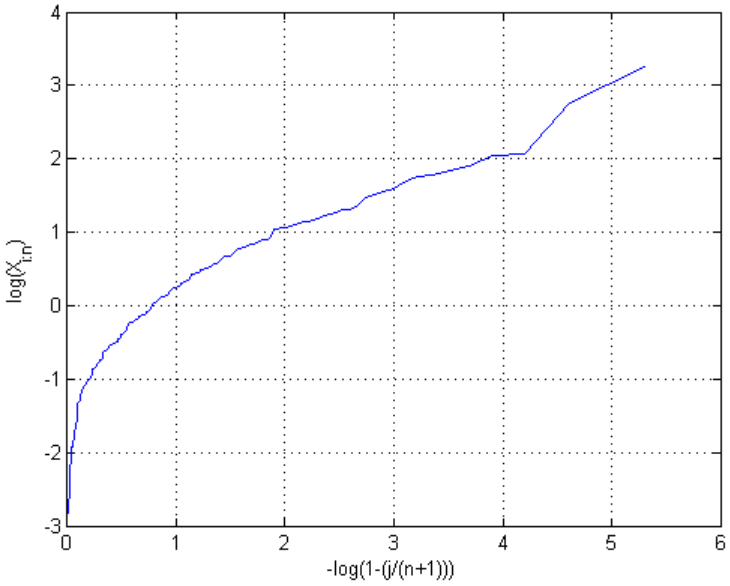


Figure 16: Pareto QQ-plot for simulated logNormal$(0, 1)$

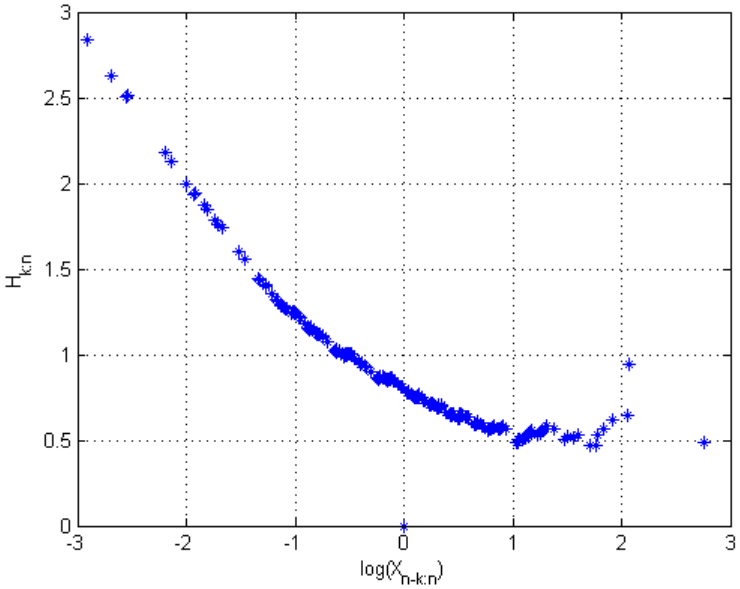


Figure 17: Hill plot for for simulated logNormal$(0, 1)$

Now using method 9 since we have a concave Pareto QQ-plot in Figure 16 for the lognormal data generated,

therefore the tail of the lognormal data is less heavy than that of the Pareto distribution. Since $H_{k:n}$ decreases with decreasing k it follows that the tail of the lognormal data is lighter than that of the Pareto.

For the weibull data generated we have the Pareto QQ-plot in Figure 18 followed by its Hill plot in Figure 19.

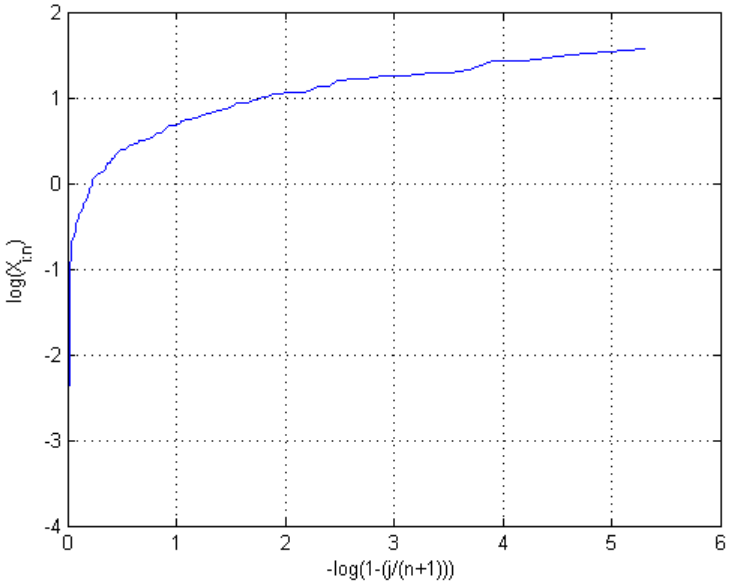


Figure 18: Pareto QQ-plot for simulated weibull$(0.5, 2)$

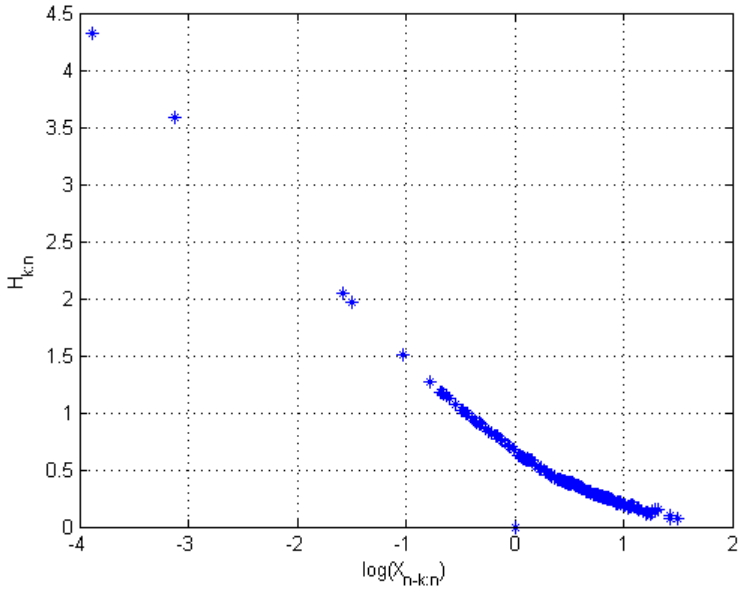


Figure 19: Hill plot for simulated Weibull$(0.5, 2)$

Similar to the lognormal data, the Pareto QQ-plot in Figure 18 of the weibull data is concave therefore the tail of the weibull data is less heavy than that of the Pareto. Note also that $H_{k:n}$ decreases with decreasing

k therefore it follows that the tail of the weibull data is lighter than that of the Pareto.

For the exponential data generated we have the Pareto QQ-plot in Figure 20 followed by its Hill plot in Figure 21.

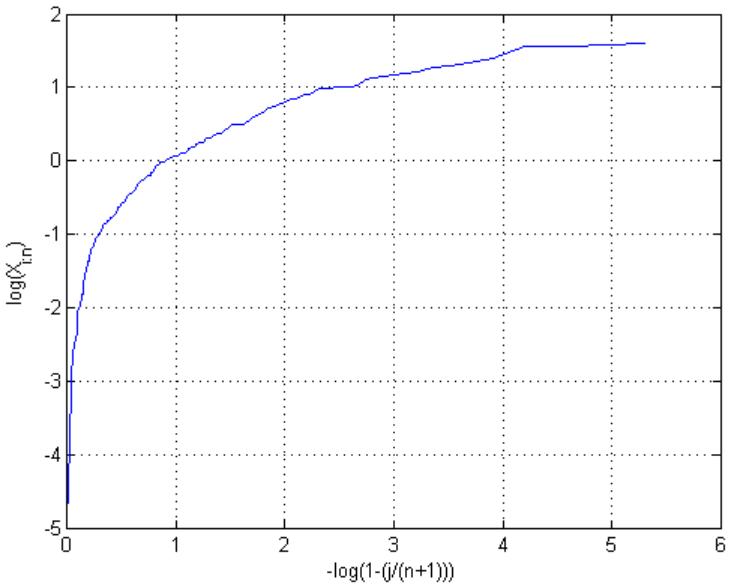


Figure 20: Pareto QQ-plot for simulated exp(1)

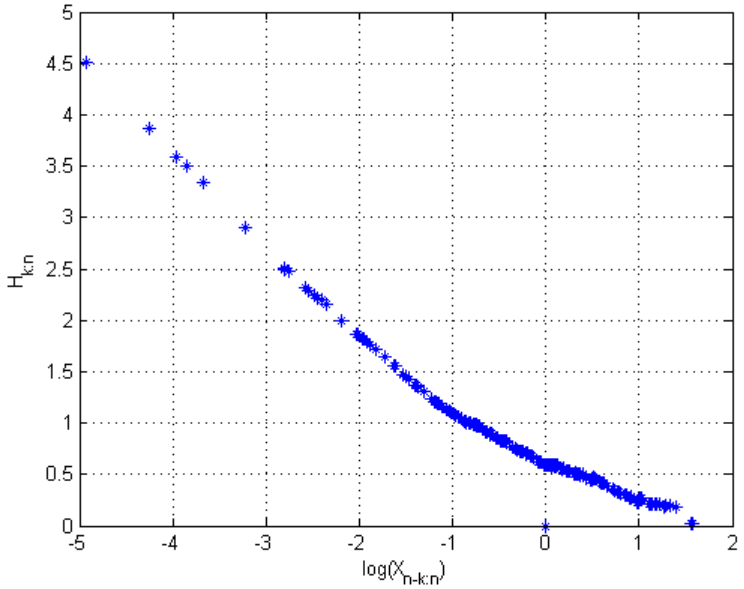


Figure 21: Hill plot for simulated exp(1)

Similar to the lognormal and Weibull data, the Pareto QQ-plot in Figure 20 of the exponential data is concave therefore the tail of the exponential data is less heavy than that of the Pareto. Note also that $H_{k:n}$

decreases with decreasing k in Figure 21 therefore it follows that the tail of the exponential data is lighter than that of the Pareto.

3.5 Splicing

It has often been seen that in the modeling of claims data that one model is sufficient to capture claims behavior over one interval where as other models are sufficient on the other intervals. This is why we consider the splicing method to create new distributions that take this into account.

Consider the two probability density functions f_1 and f_2 with corresponding distribution functions F_1 and F_2, now we have:

$$f_1^*(x) = \begin{cases} \frac{f_1(x)}{F_1(t)-F_1(t^l)} & , t^l \leq x \leq t \\ 0 & \text{otherwise} \end{cases}$$

and

$$f_2^*(x) = \begin{cases} \frac{f_2(x)}{F_2(T)-F_2(t)} & , t \leq x \leq T \\ 0 & \text{otherwise} \end{cases}$$

which is the transformation of f_1 and f_2 to valid densities on the intervals $\left[t^l; t\right]$ and $[t; T]$ where t^l and T are the lower and upper truncation but t is the splicing point. It follows that the splicing density is given by:

$$f(x) = \begin{cases} 0 & , x \leq t^l \\ \alpha f_1^*(x) & , t^l < x \leq t \\ (1-\alpha) f_2^*(x) & , t < x < T \\ 0 & , x \geq T \end{cases} \tag{3}$$

where α and $(1-\alpha)$ are constants such that $\alpha + (1-\alpha) = 1$, making $f(x)$ a legitimate density function [4].

4 Application

A simulation was performed using MATLAB R2016a [26] in which 5000 observations were generated from 3 initial distributions, namely, N(190093.2401, 81585.08159), Gamma(2, 73779) and exp(110000) where the minimum claim amount is zero and a maximum of 350000 (i.e. truncation is applied) is set for the 3 initial distributions. From the claim amount of 350000 onward 100 observations were generated from 4 distributions exp(0.5), Weibull(0.5, 0.8), LogNormal(0.01, 1.5) and Pareto(0.0001, 1.8) respectively, therefore forming a total of 12 spliced distributions.

The value of α from Equation 3 was not determined as it is mathematically technical. Rather the data was simulated to match visually, and in this section the value of α was set to be approximately equal to $\frac{5000}{5250} = 95.2\%$ such that most of data generated comes from the initial distribution. Therefore $1 - \alpha = 4.8\% \implies \alpha + (1-\alpha) = 1$, it follows that approximately 5% of the data generated comes from the tail distribution., see Figure 31-42.

The aim of the application section is to demonstrate methods 8 and 9 using real simulated data, where the data is from a spliced distribution. The 3 initial distributions and the 4 tail distributions are shown in Figures 22 and 23.

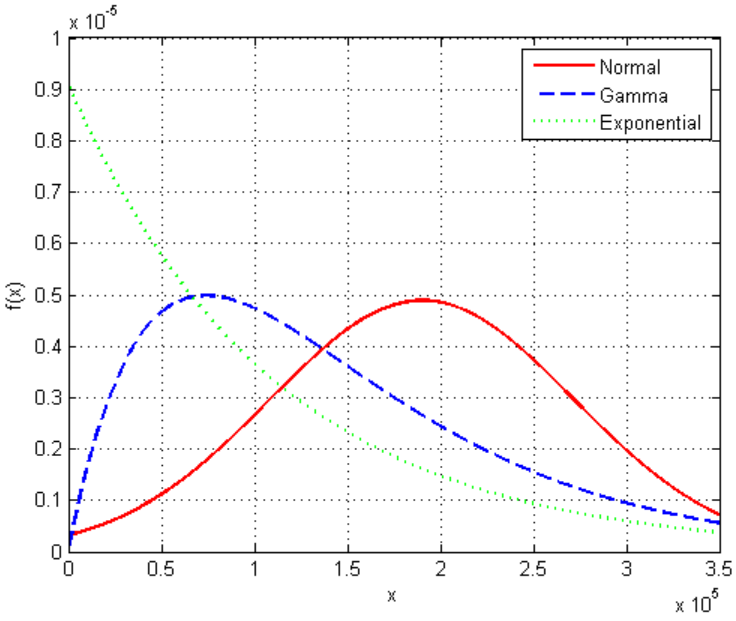


Figure 22: Initial distributions (X < 350000)

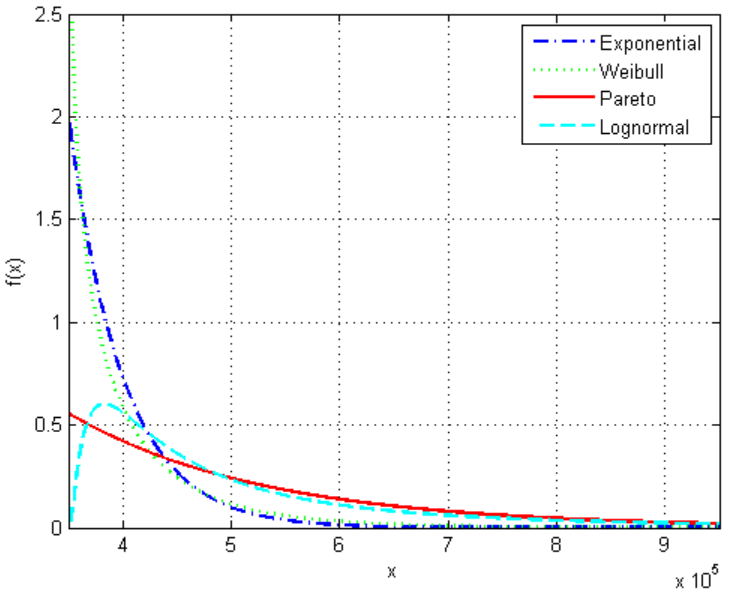


Figure 23: Tail distributions (X < 350000)

The simulated data for the 3 initial distributions, is shown in Figures 24, 25 and 26.

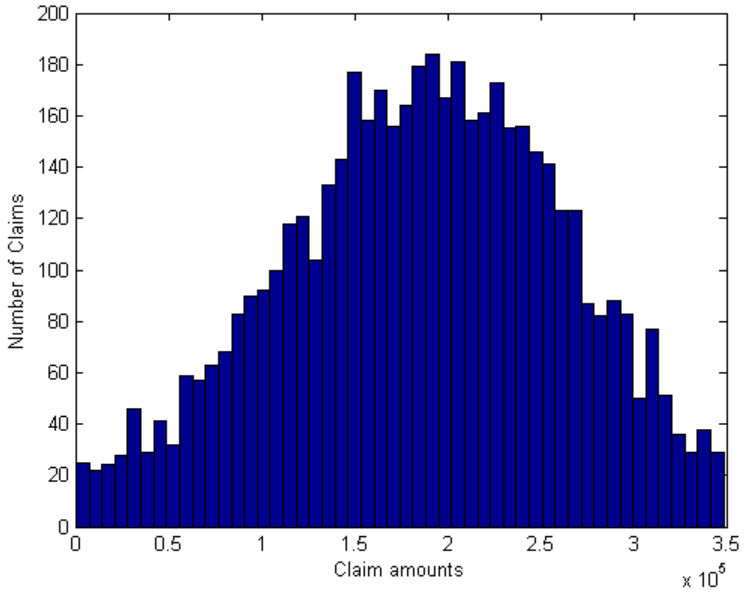


Figure 24: Simulated data for N(190093.2401, 81585.08159)

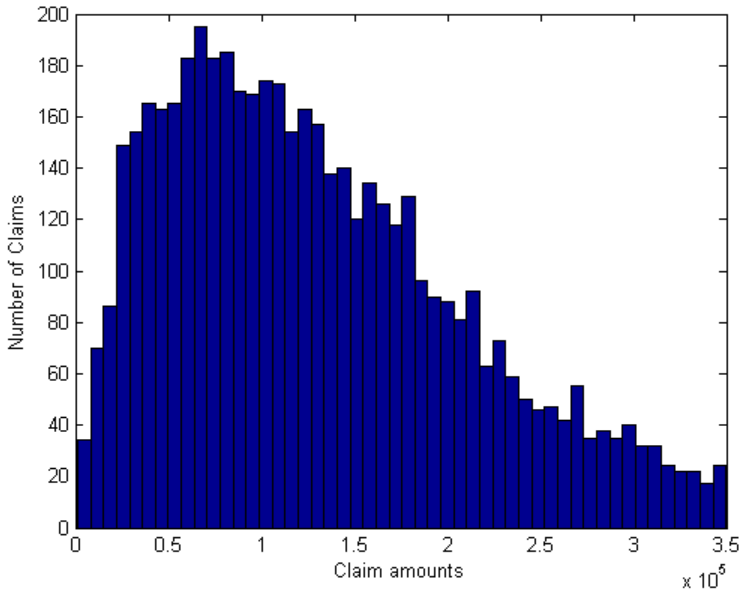


Figure 25: Simulated data for Gamma(2, 73779)

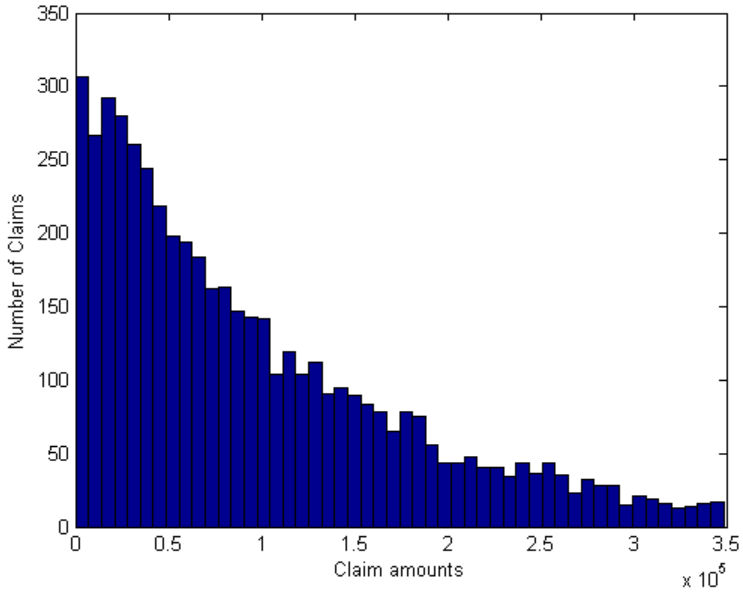


Figure 26: Simulated data for exp(110000)

The simulated data for the 4 tail distributions is shown in Figures 27, 28, 29 and 30.

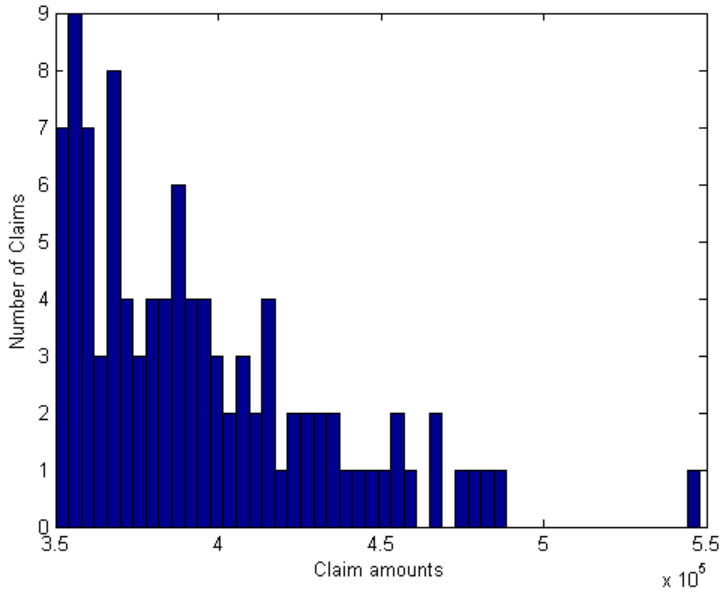


Figure 27: Simulated data for exp(0.5)

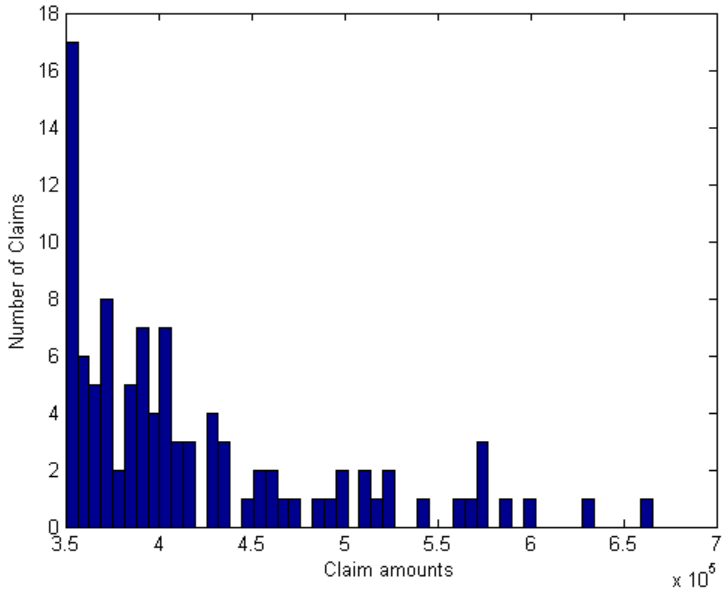


Figure 28: Simulated data for Weibull$(0.5, 0.8)$

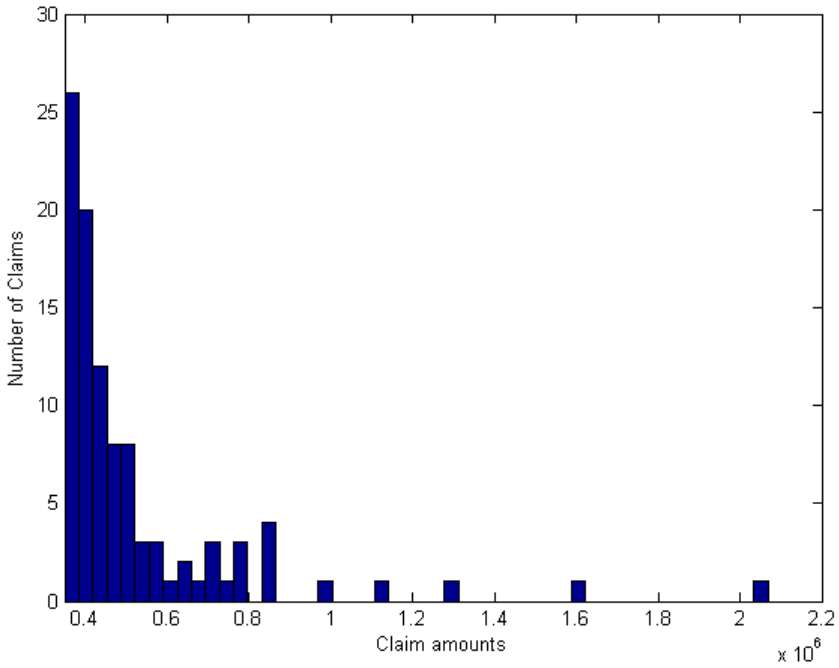


Figure 29: Simulated data for LogNormal$(0.01, 1.5)$

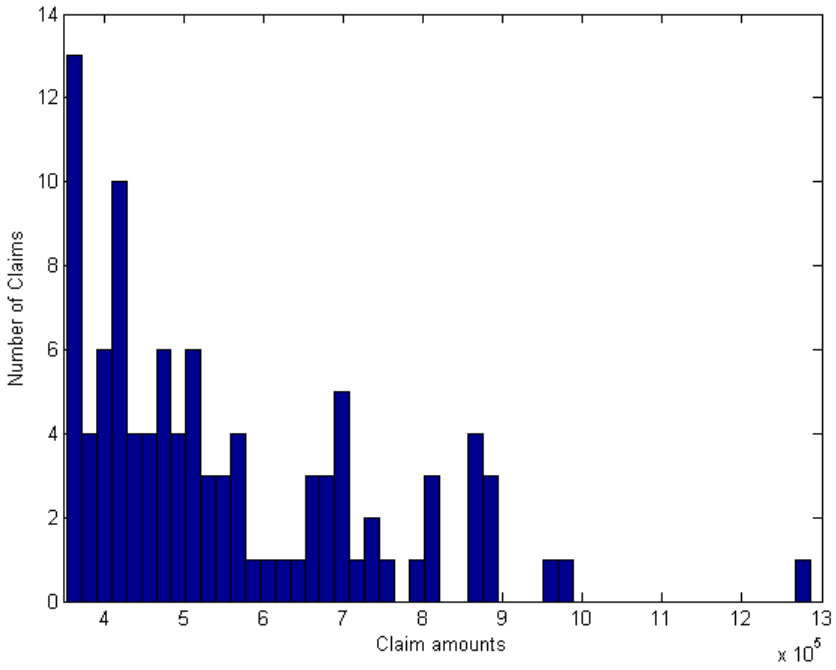


Figure 30: Simulated data for Pareto$(0.0001, 1.8)$

Figures 31-42 show the 12 simulated data from the 12 spliced distributions were obtained.

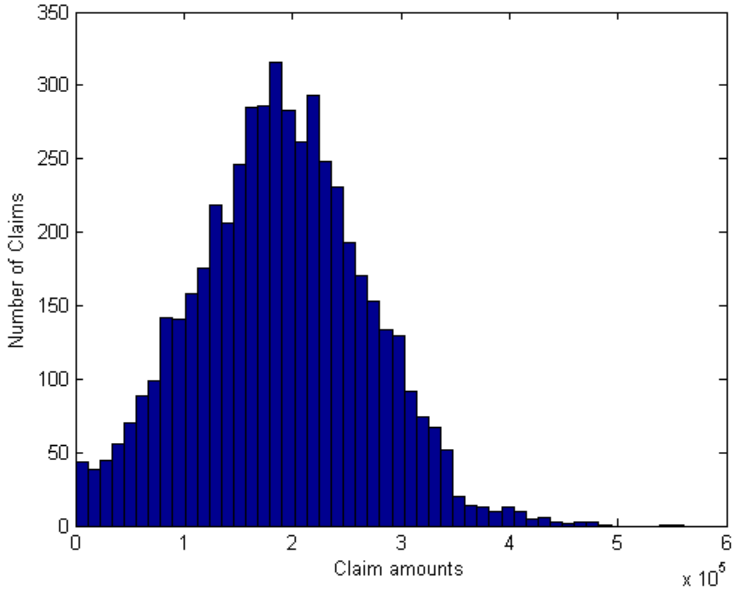


Figure 31: N$(190093.2401, 81585.08159)$ spliced with exp(0.5)

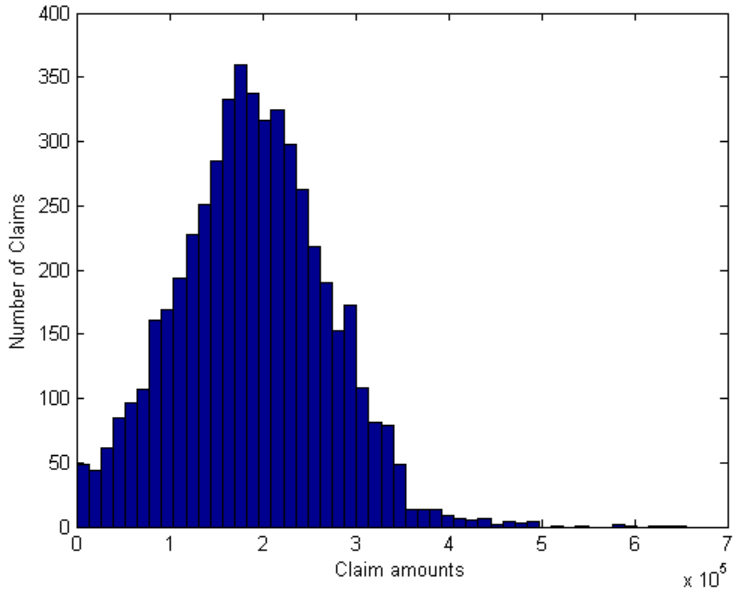


Figure 32: N(190093.2401, 81585.08159) spliced with Weibull(0.5, 0.8)

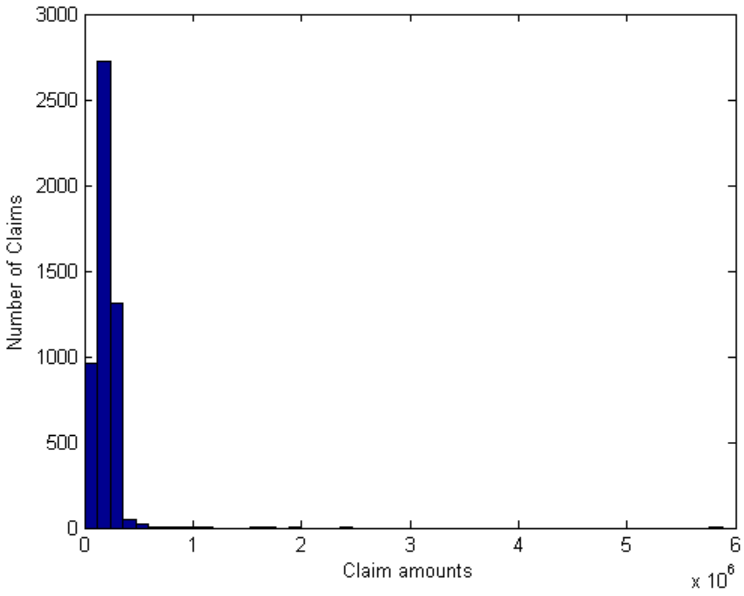


Figure 33: N(190093.2401, 81585.08159) spliced with LogNormal(0.01, 1.5)

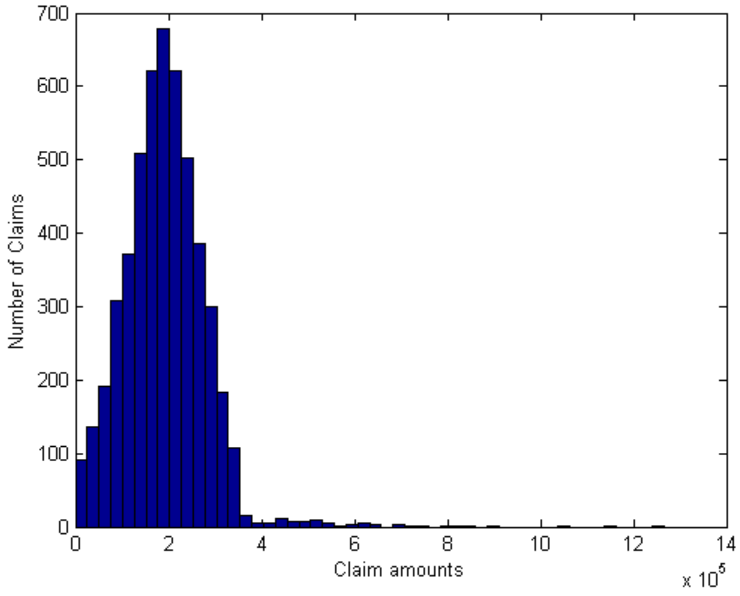


Figure 34: N(190093.2401, 81585.08159) spliced with Pareto(0.0001, 1.8)

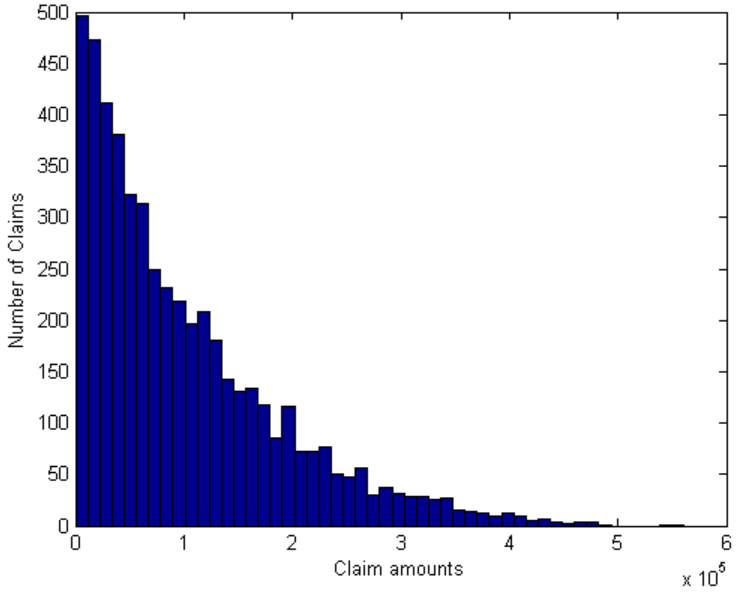


Figure 35: exp(110000) spliced with exp(0.5)

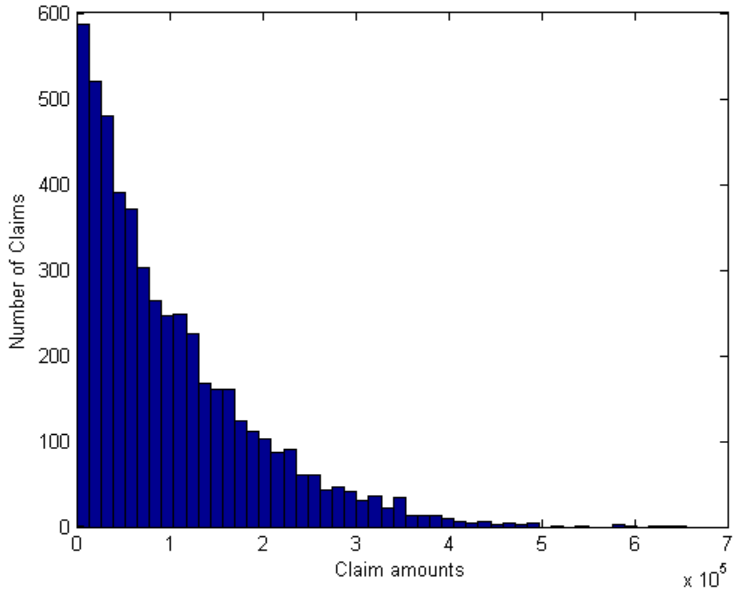


Figure 36: exp(110000) spliced with Weibull$(0.5, 0.8)$

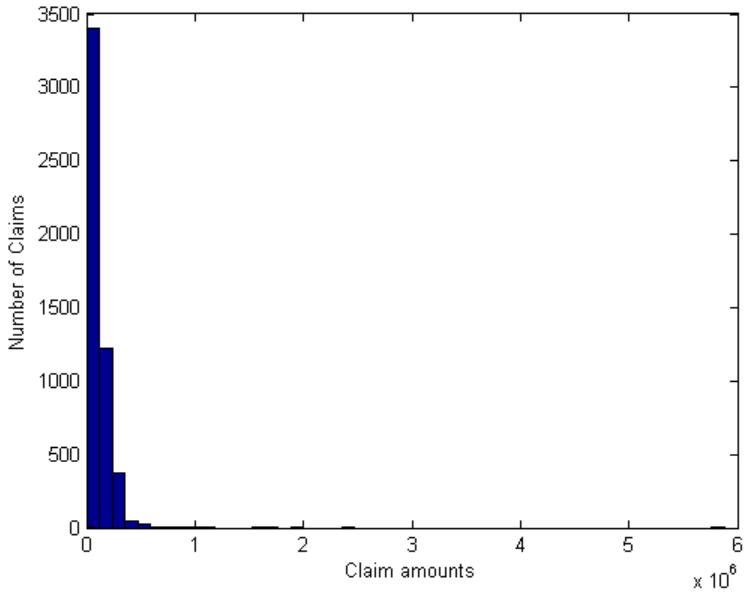


Figure 37: exp(110000) spliced with LogNormal$(0.01, 1.5)$

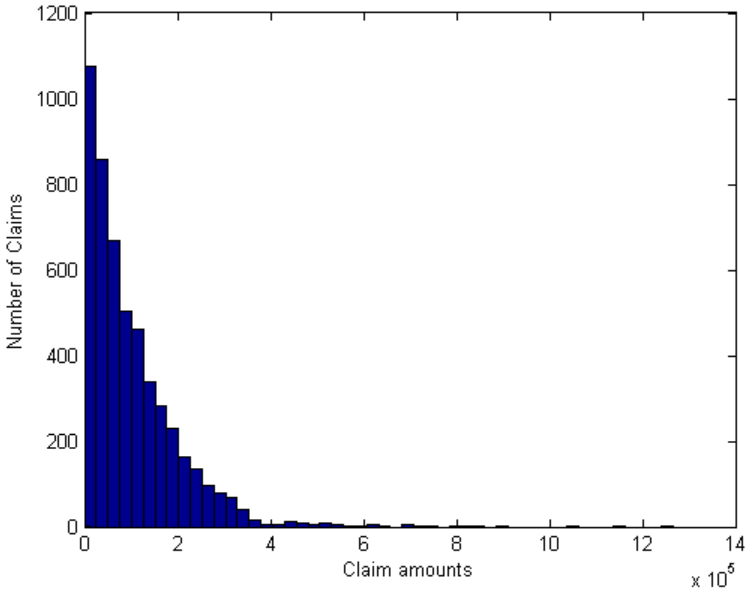


Figure 38: exp(110000) spliced with Pareto$(0.0001, 1.8)$

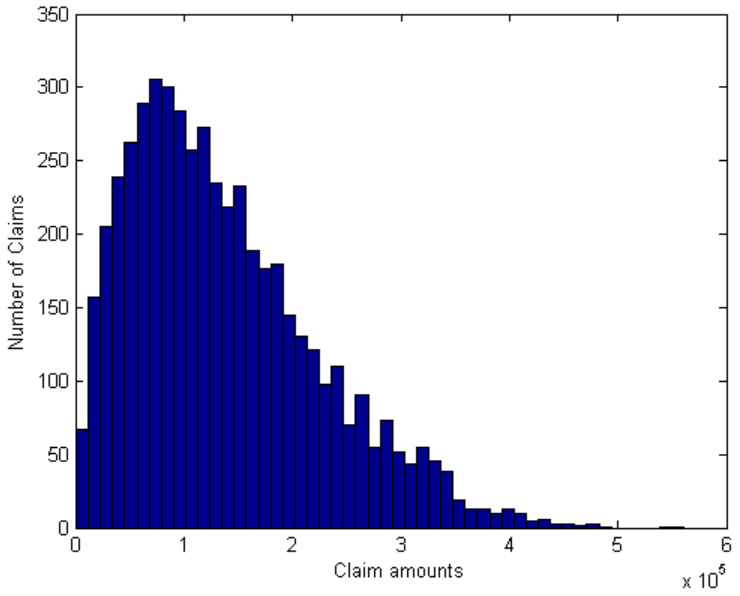


Figure 39: Gamma$(2, 73779)$ spliced with exp(0.5)

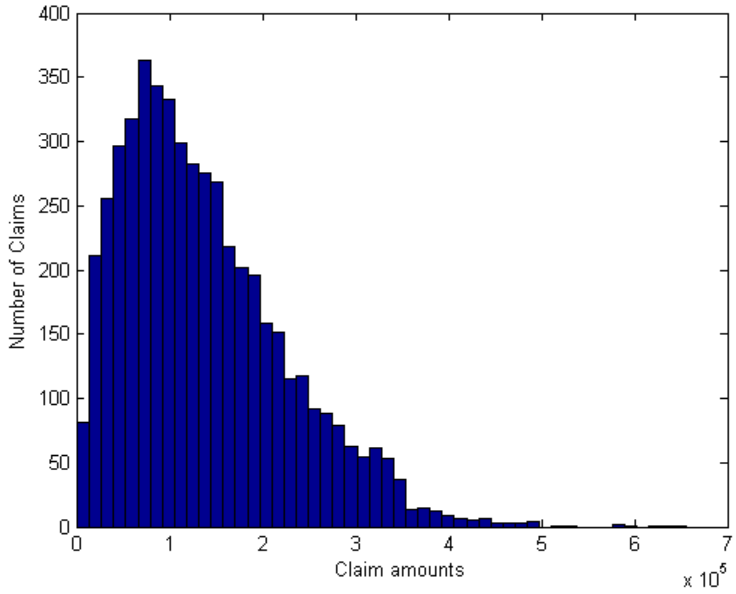


Figure 40: Gamma$(2, 73779)$ spliced with Weibull$(0.5, 0.8)$

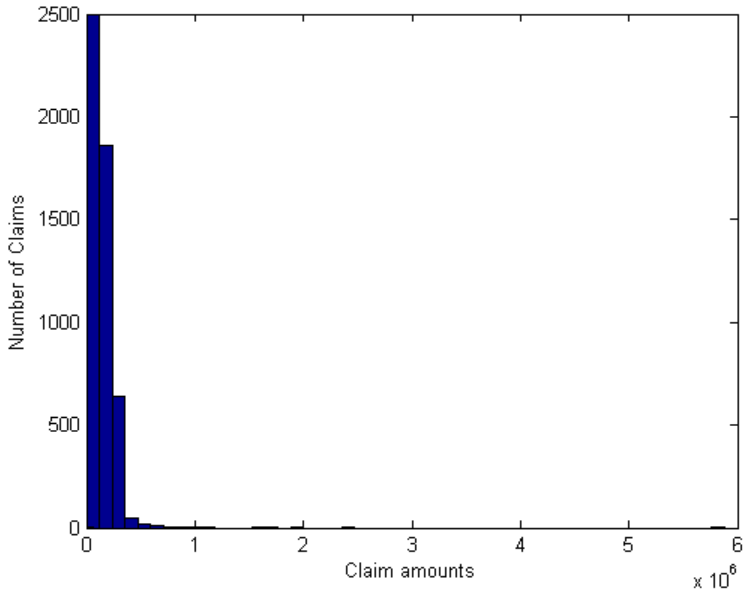


Figure 41: Gamma$(2, 73779)$ spliced with LogNormal$(0.01, 1.5)$

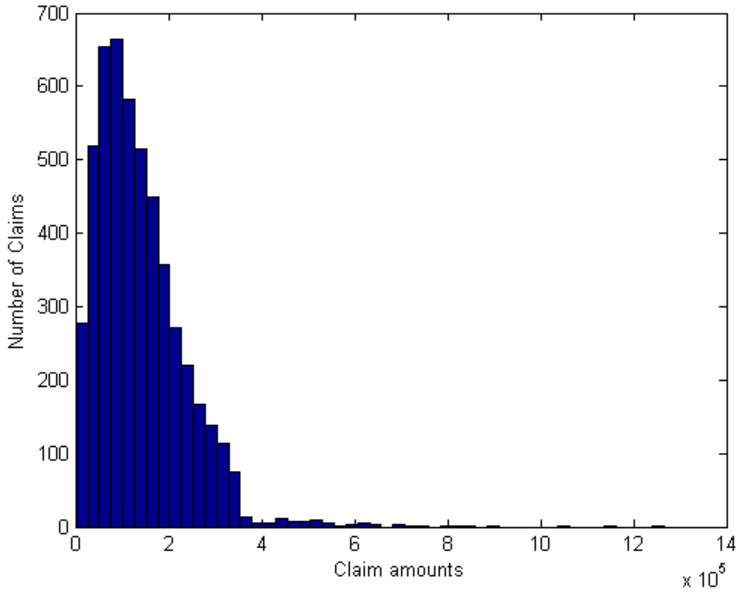


Figure 42: Gamma$(2, 73779)$ spliced with Pareto$(0.0001, 1.8)$

4.1 Unknown splicing point

Now we look at a situation where the splicing point is assumed unknown and we would like to know when a heavy tail has occurred in the 12 spliced distributions above. Using the simulated data from the 12 spliced distributions and applying method 8 the following exponential QQ-plots and mean excess plots and were obtained.

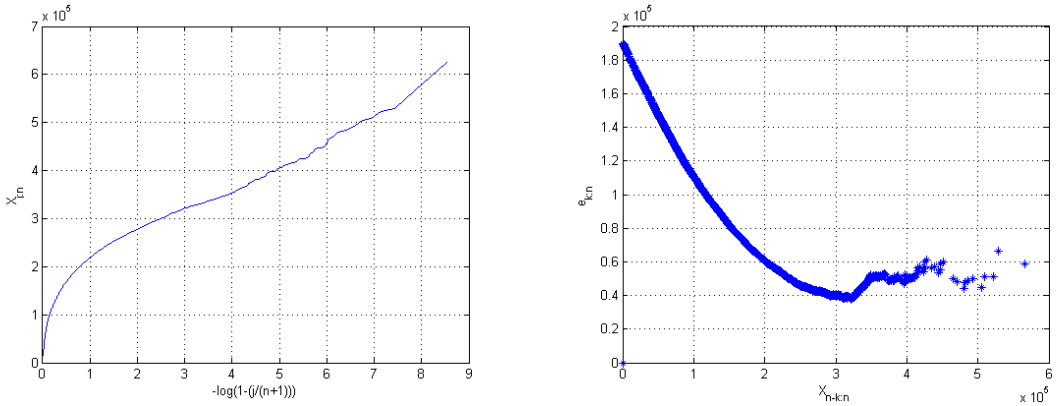


Figure 43: Exponential QQ-plot and mean excess plot for simulated N$(190093.2401, 81585.08159)$ spliced with exp(0.5)

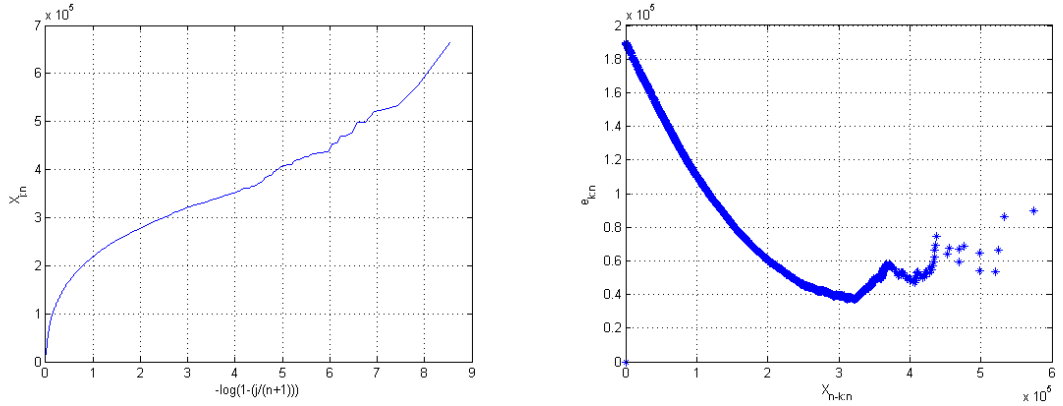


Figure 44: Exponential QQ-plot and mean excess plot for simulated N(190093.2401, 81585.08159) spliced with Weibull(0.5, 0.8)

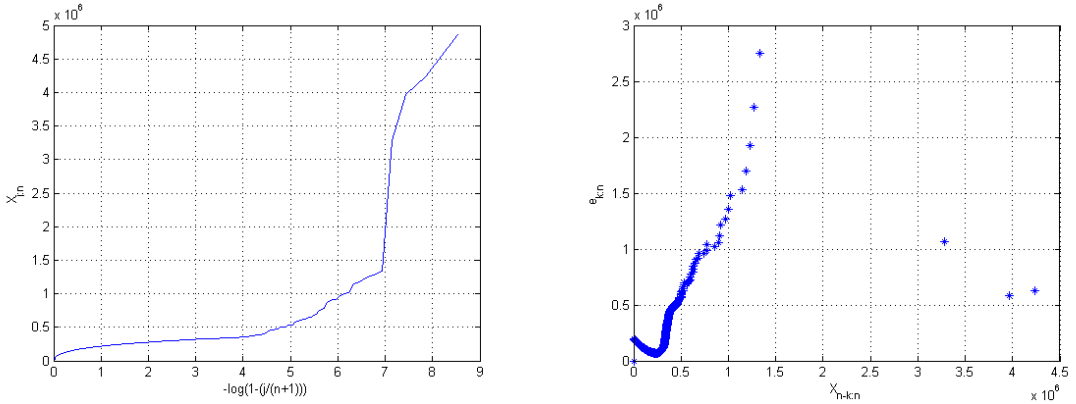


Figure 45: Exponential QQ-plot and mean excess plot for simulated N(190093.2401, 81585.08159) spliced with LogNormal(0.01, 1.5)

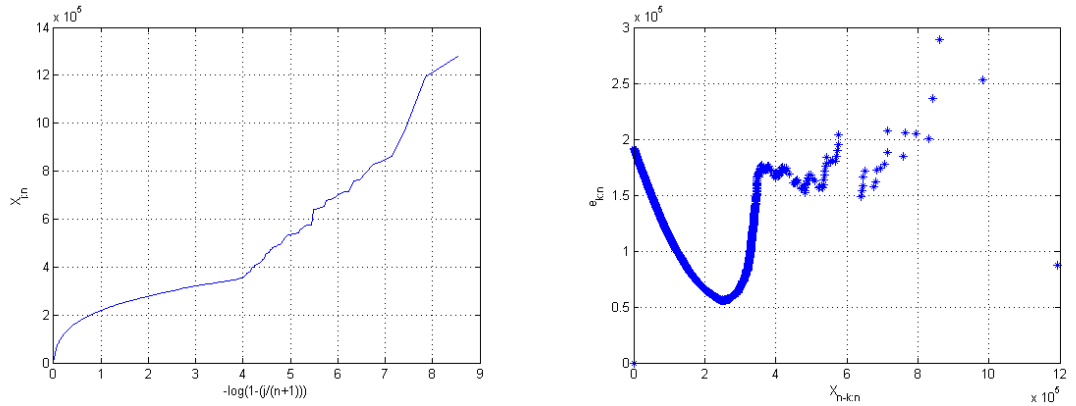


Figure 46: Exponential QQ-plot and mean excess plot for simulated N(190093.2401, 81585.08159) spliced with Pareto(0.0001, 1.8)

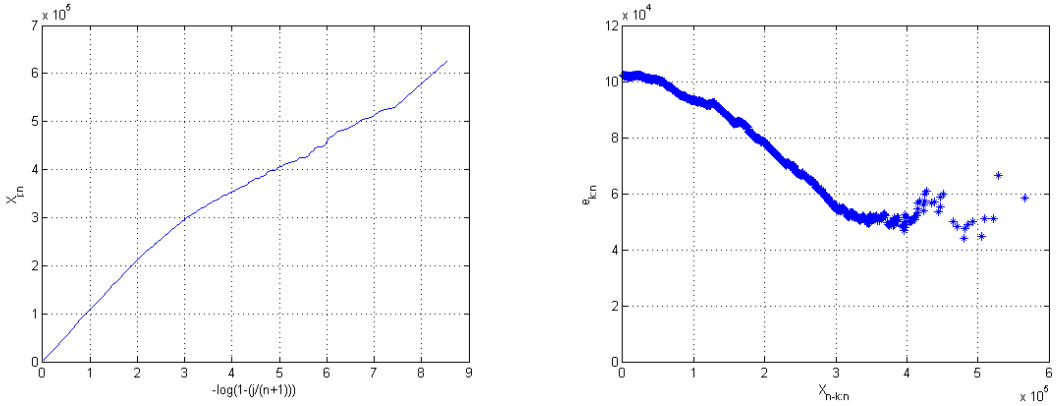


Figure 47: Exponential QQ-plot and mean excess plot for simulated exp(110000) spliced with exp(0.5)

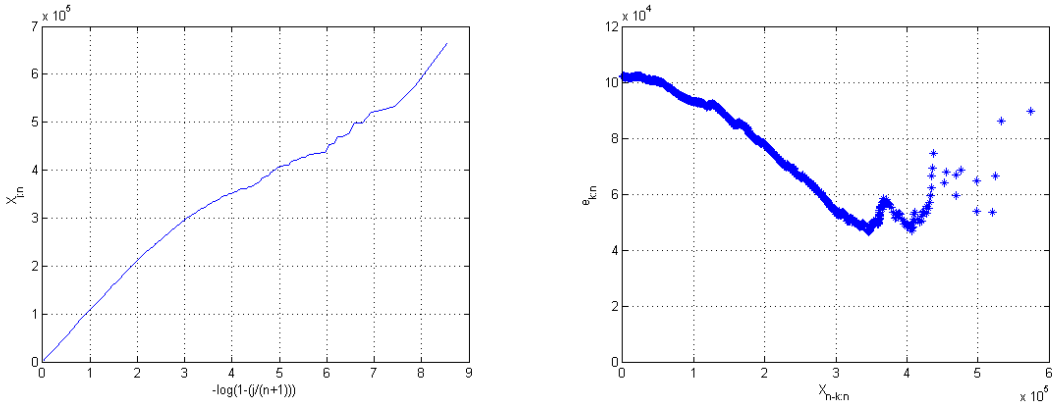


Figure 48: Exponential QQ-plot and mean excess plot for simulated exp(110000) spliced with Weibull$(0.5, 0.8)$

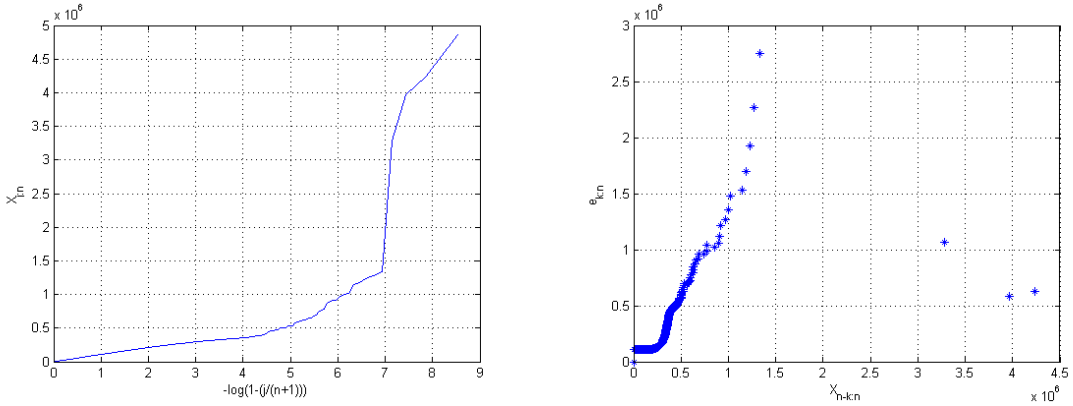


Figure 49: Exponential QQ-plot and mean excess plot for simulated exp(110000) spliced with LogNormal$(0.01, 1.5)$

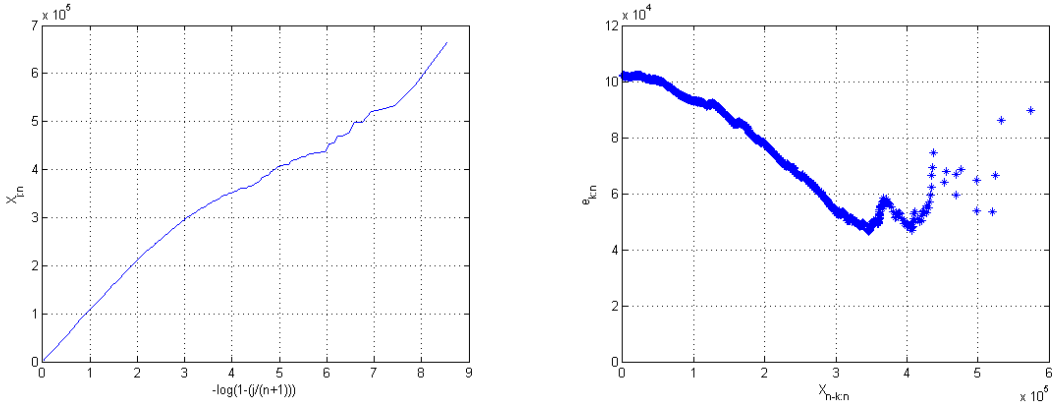


Figure 50: Exponential QQ-plot and mean excess plot for simulated exp(110000) spliced with Pareto$(0.0001, 1.8)$

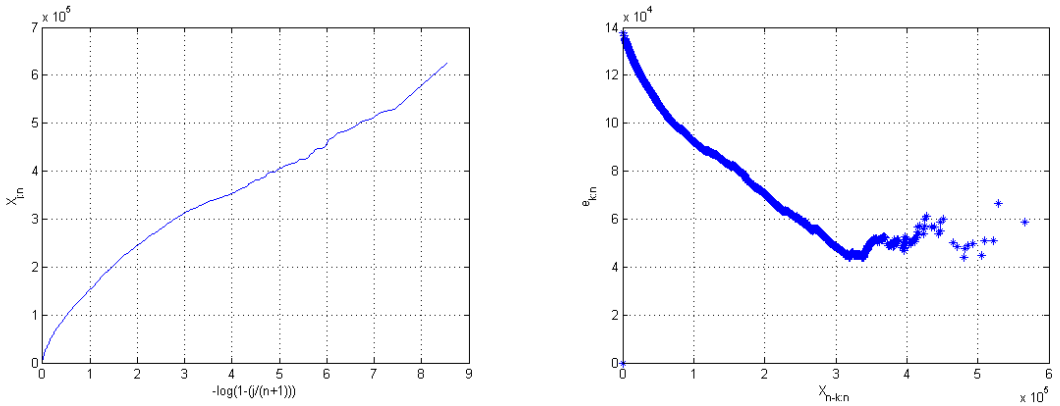


Figure 51: Exponential QQ-plot and mean excess plot for simulated Gamma$(2, 73779)$ spliced with exp(0.5)

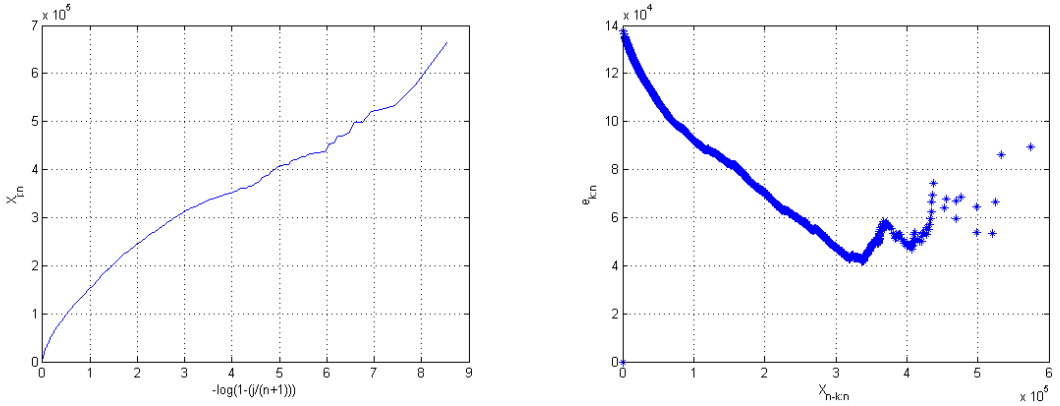


Figure 52: Exponential QQ-plot and mean excess plot for simulated Gamma$(2, 73779)$ spliced with Weibull$(0.5, 0.8)$

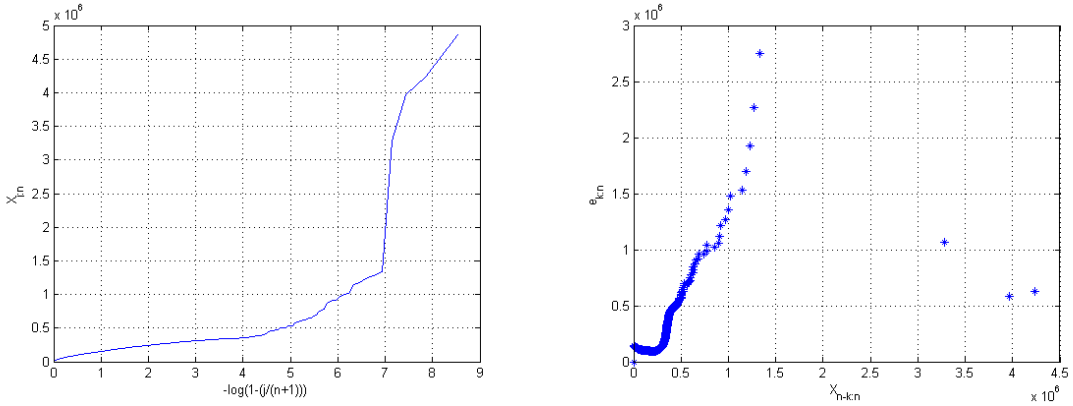


Figure 53: Exponential QQ-plot and mean excess plot for simulated Gamma$(2, 73779)$ spliced with LogNormal$(0.01, 1.5)$

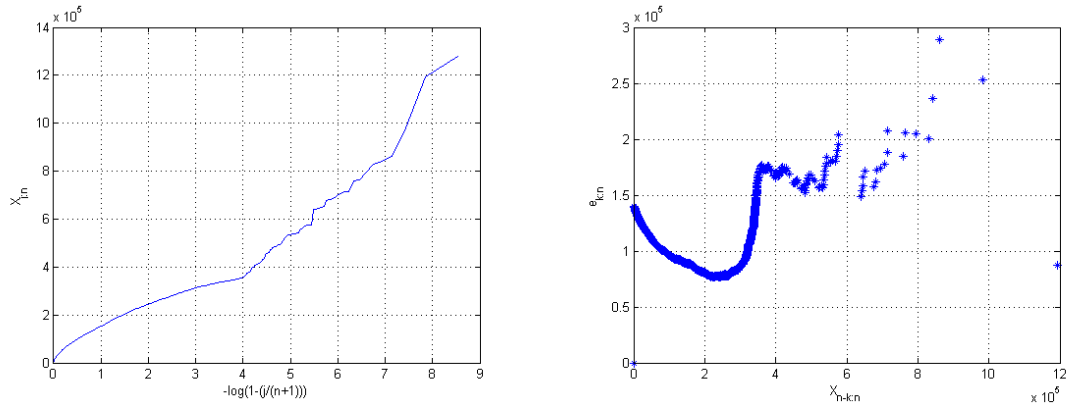


Figure 54: Exponential QQ-plot and mean excess plot for simulated Gamma$(2, 73779)$ spliced with Pareto$(0.0001, 1.8)$

Now consider the exponential QQ-plot in Figure 48, by looking at the horizontal axis $\left[-\log(1-\frac{j}{n+1})\right]$ we see that from 0 to 4 we have a relatively linear line, from 4 onward we see the line taking a convex shape. Furthermore $\log(4) = 1.386$ on the log scale. This sudden change in the behavior of the line is caused by the splicing of two different distributions at some point, in this case the point is 350000. Since the mean excess plot is convex from 4 onward by Method 8, the spliced distribution is heavy tailed. Similar reasoning can be used for the other spliced distributions using their exponential QQ-plots.

This can be useful in practice because if we have claims data from particular insurance company and would like to determine the splicing point, looking at the exponential QQ-plot for the claims data we can determine the splicing point then split the claims data at that particular point. This is done so that a further analysis can be performed on the two pieces separately.

Now using the simulated data from the 12 spliced distributions and applying Method 9 Figures 55 to 66 Pareto QQ-plots and Hill plots were obtained.

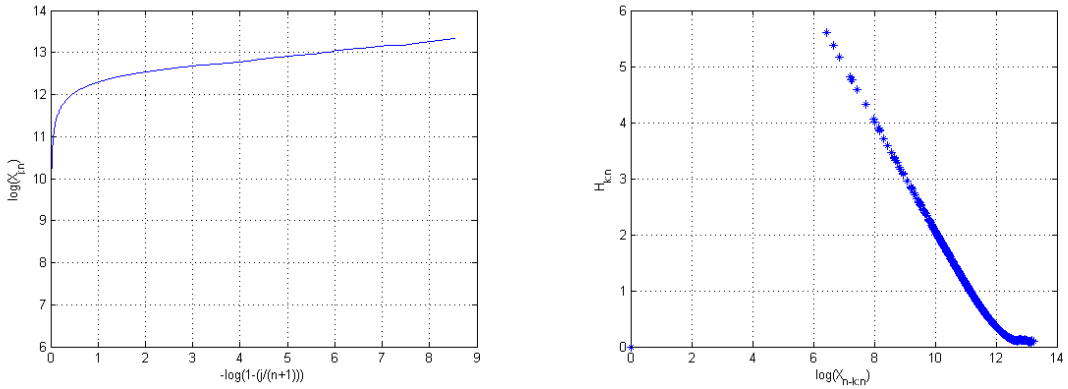


Figure 55: Pareto QQ-plot and Hill plot for simulated N(190093.2401, 81585.08159) spliced with exp(0.5)

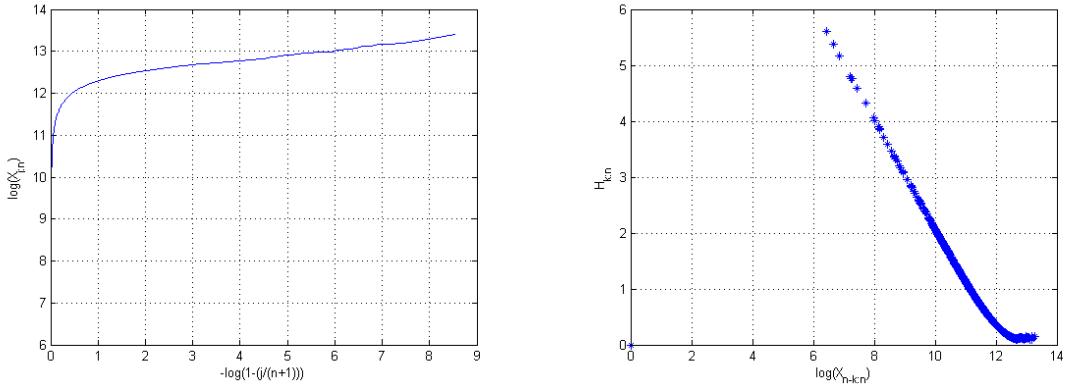


Figure 56: Pareto QQ-plot and Hill plot for simulated N(190093.2401, 81585.08159) spliced with Weibull(0.5, 0.8)

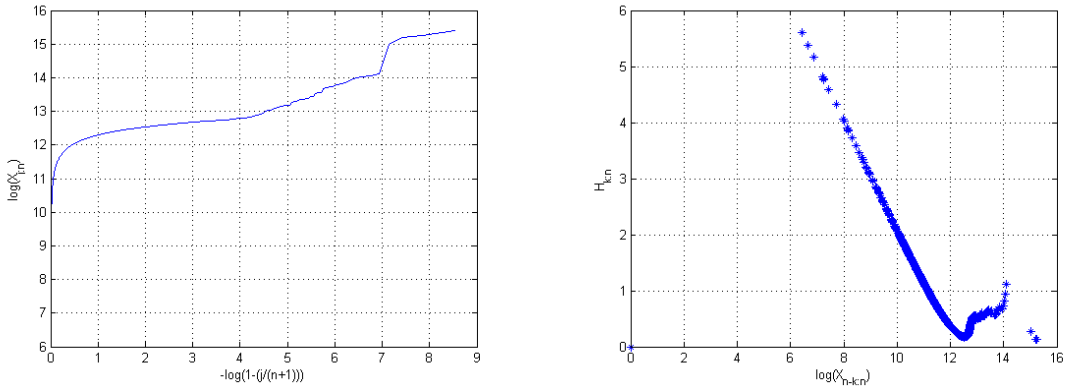


Figure 57: Pareto QQ-plot and Hill plot for simulated N(190093.2401, 81585.08159) spliced with LogNormal(0.01, 1.5)

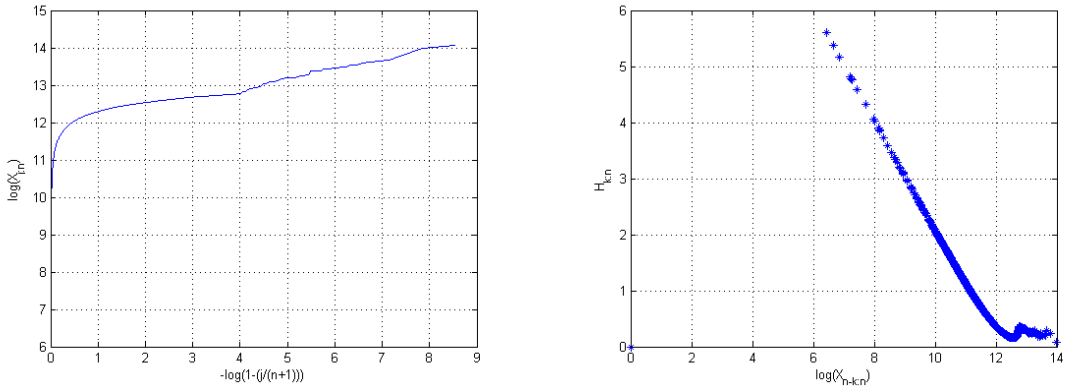


Figure 58: Pareto QQ-plot and Hill plot for simulated $N(190093.2401, 81585.08159)$ spliced with $\text{Pareto}(0.0001, 1.8)$

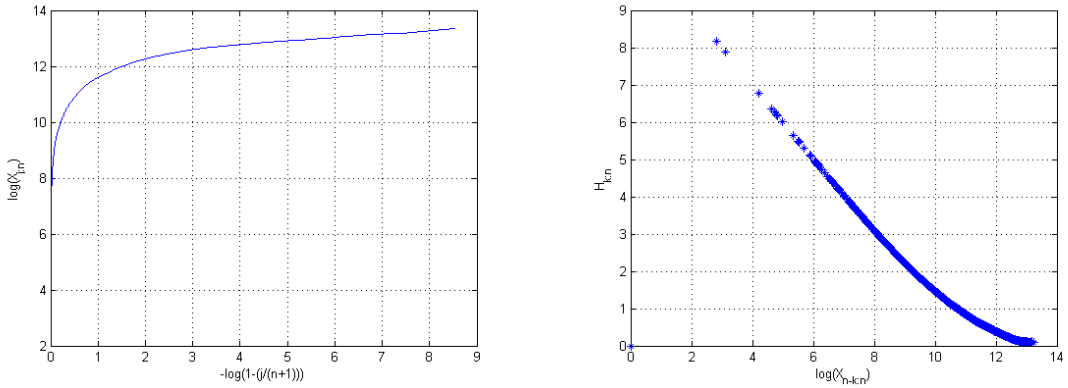


Figure 59: Pareto QQ-plot and Hill plot for simulated exp(110000) spliced with exp(0.5)

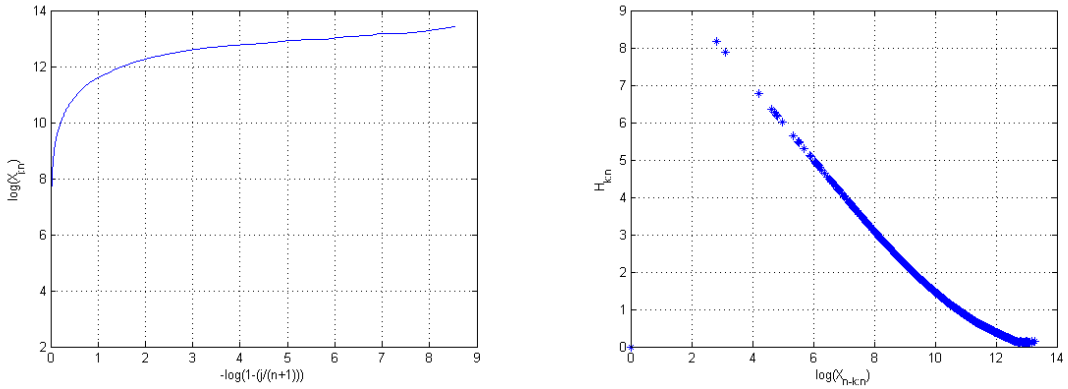


Figure 60: Pareto QQ-plot and Hill plot for simulated exp(110000) spliced with Weibull$(0.5, 0.8)$

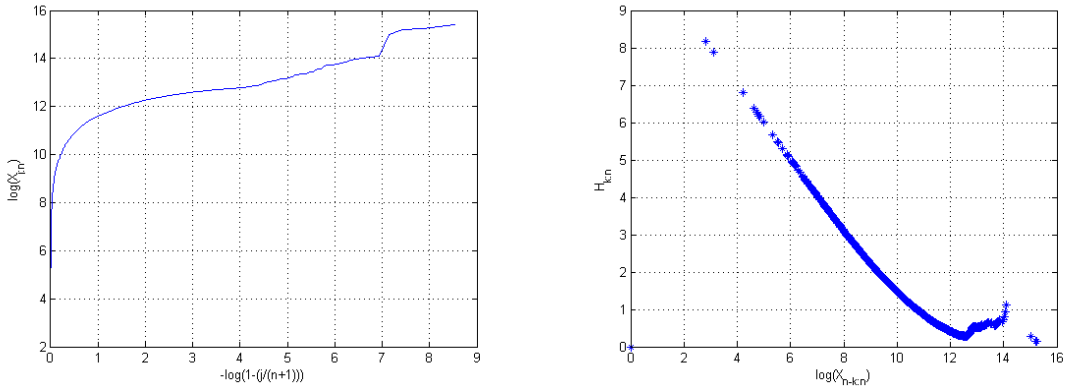


Figure 61: Pareto QQ-plot and Hill plot for simulated exp(110000) spliced with LogNormal$(0.01, 1.5)$

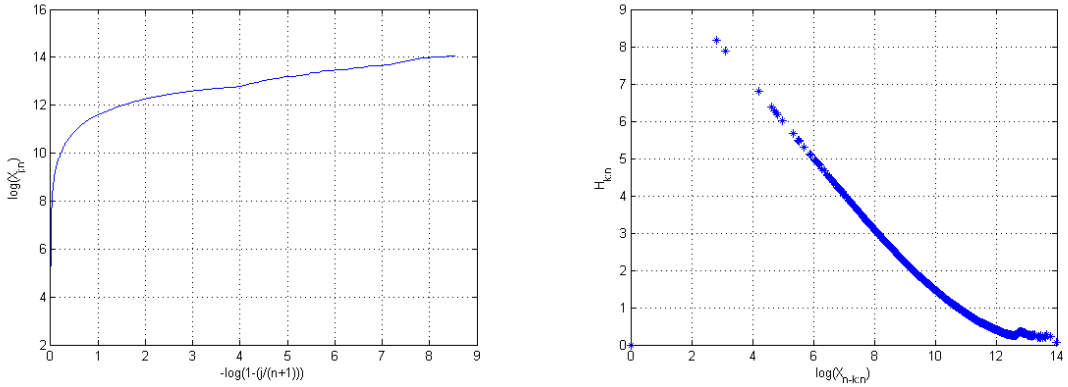


Figure 62: Pareto QQ-plot and Hill for simulated exp(110000) spliced with Pareto$(0.0001, 1.8)$

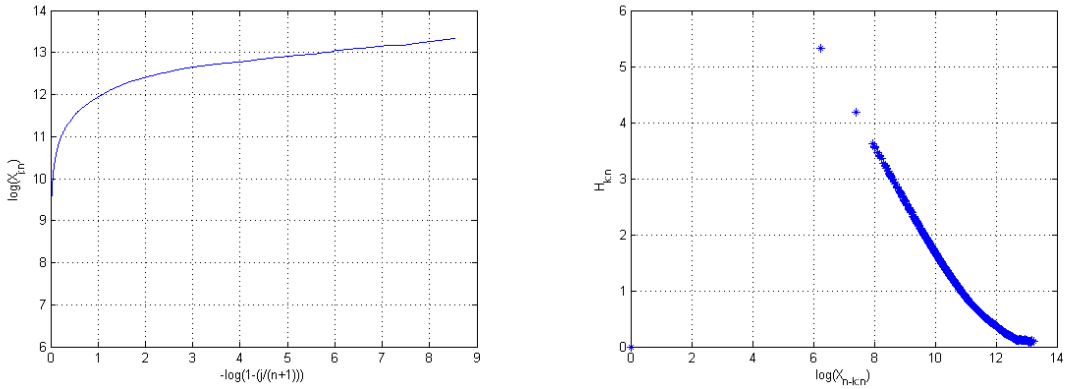


Figure 63: Pareto QQ-plot and Hill plot for simulated Gamma$(2, 73779)$ spliced with exp(0.5)

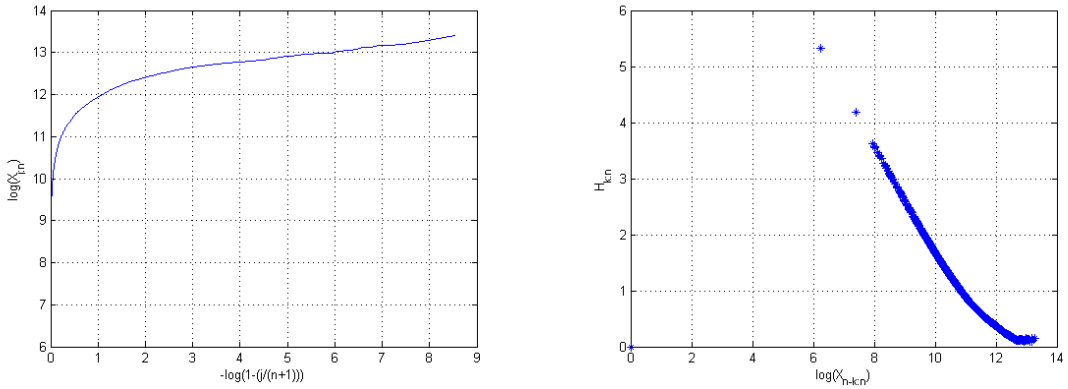


Figure 64: Pareto QQ-plot and Hill plot for simulated Gamma$(2, 73779)$ spliced with Weibull$(0.5, 0.8)$

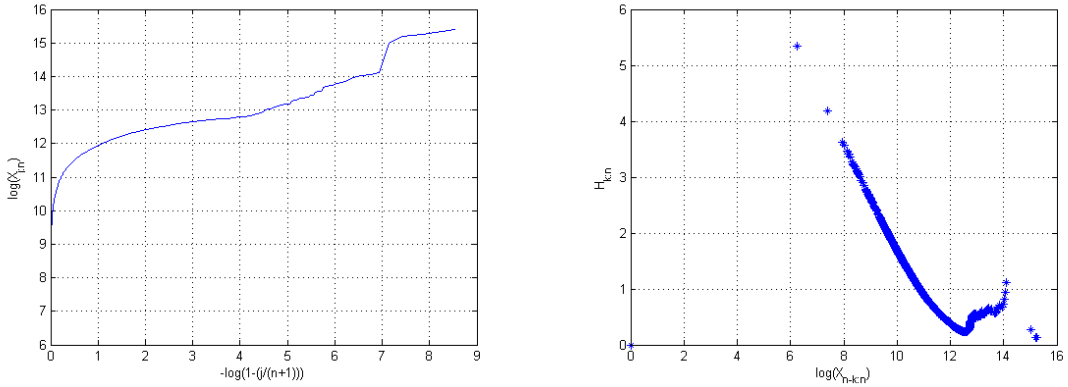


Figure 65: Pareto QQ-plot and Hill plot for simulated Gamma$(2, 73779)$ spliced with LogNormal$(0.01, 1.5)$

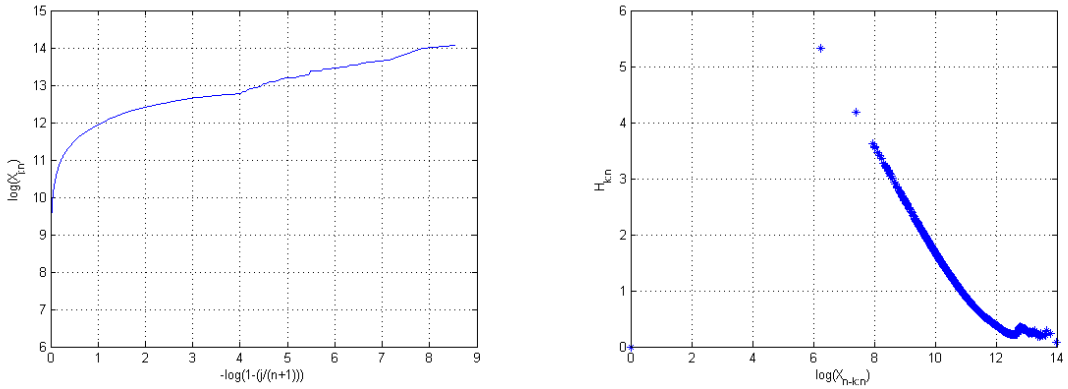


Figure 66: Pareto QQ-plot and Hill plot for simulated Gamma$(2, 73779)$ spliced with Pareto$(0.0001, 1.8)$

We can see that the Pareto QQ-plots do not illustrate the splicing point as effectively as the exponential QQ-plots. However, the Hill plots do indicate the tails from the point $\log(X_{n-k:n}) = 12.7$ onward on the horizontal axis. For example if we look at Figures 58, 62 and 66 where the tail distribution is a Pareto

distribution, we see that from the point $\log(X_{n-k:n}) = 12.7$ onward we have a relatively horizontal plot whereas the other Hill plots do not behave in this nature. Furthermore $\log(350000) = 12.766$, which is very close to 12.7 where 350000 is the approximated splicing point.
The application of methods $1 - 7$ will not be illustrated in this section since it is illustrated in the M.Sc. thesis "Statistical distributions in general insurance stochastic processes"[24].

5 Conclusion

On the basis of the literature review it is evident that a huge amount of research is available on the modeling of claims data and measuring of tail heaviness.

The research conducted revealed that when the application techniques, which were gathered in the study, were applied to the simulated claims data it is often difficult to find a single distribution to describe the distribution of the whole range of observed claims. It was later realized in the application phase that in-order to overcome this a segmentation based on claim size can be performed so that a specific distribution may be fitted on the upper range or tail of the observed claims data while another distribution is fitted on the lower range of the observed claims data. These distributions can then be considered as one single distribution known as a spliced distribution. As a result one would have to incorporate these two distributions and the splicing point between the two into the algorithm to fit the model.

It was also found that some methods for detecting tail heaviness are easier to implement then others and that some methods obtain results more accurately than others.

With the above in mind, it is suggested that judgment not be based solely on a theoretical perspective but also on a expert analyst's opinion on whether the fitted distributions of the model make sense or not.

References

[1] S Asmussen, H Schmidli, and V Schmidt. Tail probabilities for non-standard risk and queueing processes with subexponential jumps. *Advanced Applied Probability*, 31:422–477, 1999.

[2] R Barlow, A Marshall, and F Proschan. Properties of probability distributions with monotone hazard rates. *The Annals of Mathematical Statistics*, 34(2):375–389, 1963.

[3] J Beirlant, C Bouquiaux, and BJM Werker. Semiparametric lower bounds for tail index estimation. *Journal of Statistical Planning and Inference*, 136(3):705–729, 2006.

[4] J Beirlant, Y Goegebeur, J Segers, and J Teugels. *Statistics of Extremes: Theory and Applications*. John Wiley & Sons, 2006.

[5] M Brown. Further monotonicity properties for specialised renewal processes. *The Annals of Probability*, 9(5):891–895, 1981.

[6] M Bryson. Heavy-tailed distributions: Properties and tests. *Technometrics*, 16(1):61–68, 1974.

[7] H Drees. Minimax risk bounds in extreme value theory. *The Annals of Statistics*, 29(1):266–294, 2001.

[8] P Embrechts, R Frey, and H Furrer. *Stochastic Processes in Insurance and Finance*. To find, 1999.

[9] P Embrechts, C Kluppelberg, and T Mikosch. *Modelling Extremal Events: for Insurance and Finance*, volume 33. Springer Science & Business Media, 2013.

[10] Ronald Aylmer Fisher and Leonard Henry Caleb Tippett. Limiting forms of the frequency distribution of the largest or smallest member of a sample. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 24, pages 180–190. Cambridge Univ Press, 1928.

[11] S Foss, D Korshunov, and S Zachary. *An Introduction to Heavy-Tailed and Subexponential Distributions*. Springer, 2011.

[12] Edward W Frees and Emiliano A Valdez. Hierarchical insurance claims modeling. *Journal of the American Statistical Association*, 103(484):1457–1469, 2008.

[13] B Gnedenko. Sur la distribution limite du terme maximum d'une serie aleatoire. *Annals of mathematics*, 44(3):423–453, 1943.

[14] Dominique Guegan and Jing Zhang. Change analysis of a dynamic copula for measuring dependence in multivariate financial data. *Quantitative Finance*, 10(4):421–430, 2010.

[15] P Hall. On estimating the endpoint of a distribution. *The Annals of Statistics*, 10(2):556–568, 1982.

[16] Carrie N Klabunde, Joan L Warren, and Julie M Legler. Assessing comorbidity using claims data: an overview. *Medical Care*, 40(8):4–26, 2002.

[17] Christopher A Powers, Christina M Meyer, M Christopher Roebuck, and Baze Vaziri. Predictive modeling of total healthcare costs using pharmacy claims data: a comparison of alternative econometric cost modeling techniques. *Medical Care*, 43(11):1065–1072, 2005.

[18] Ronald H Randles and Douglas A Wolfe. *Introduction to the Theory of Nonparametric Statistics*, volume 1. Wiley New York, 1979.

[19] S I Resnick. Heavy tail modeling and teletraffic data: special invited paper. *The Annals of Statistics*, 25(5):1805–1869, 1997.

[20] J Rolski, H Schmidli, V Schmidt, and J Teugels. *Stochastic Processes for Insurance and Finance*. John Wiley and Sons, 1999.

[21] J Rolski, H Schmidli, V Schmidt, and J Teugels. *Stochastic Processes for Insurance and Finance*. John Wiley and Sons, 2008.

[22] Paulo Jose Araujo dos Santos. *Excesses, durations and forecasting value-at-risk*. PhD thesis, University of Lisbon, 2011.

[23] SAS Institute. *SAS 9.4*. SAS Institute, 2016.

[24] JHH Steenkamp. Statistical distributions in general insurance stochastic processes. Master's thesis, University of Pretoria, 2014.

[25] J Teugels. The class of subexponential distributions. *The Annals of Probability*, 3(6):1000–1011, 1975.

[26] The MathWorks Inc. *MATLAB R2016a version 9*. The MathWorks Inc., Natick, Massachusetts, 2016.

[27] Karen CH Yip and Kelvin KW Yau. On modeling claim frequency data in general insurance with extra zeros. *Insurance: Mathematics and Economics*, 36(2):153–163, 2005.

Appendix

MATLAB Code [26]

```
%Code illustrating tail heaviness;
x=1:0.1:10;
y1=exppdf(x,1);
y2=wblpdf(x,2,1.5);
y3=lognpdf(x,0,1);
y4=gppdf(x,2.5,1,0);

figure
plot(x,y1,'Linestyle','-.','Color','b','Linewidth',2)
hold on
plot(x,y2,'Linestyle','--','Color','g','Linewidth',2)
hold on
plot(x,y3,'Linestyle',':','Color','b','Linewidth',2)
hold on
plot(x,y4,'color','r','Linewidth',2)
legend({'Exponential','Weibull','Lognormal','Pareto'},'Location','NorthEast');
hold off
grid on
```

SAS Code [23]

```
/*--------------------------------SAS Code start-----------------------------------*/
proc iml;
n=200;
/*seed=10;*/

wbl=J(n,1);
a=2;
b=0.5;
exp=J(n,1);

logn=J(n,1);
Par=J(n,1);

call randgen(Par,'PARETO',2.5,1); /* fill 200 observations generated from Pareto(2.5,1) */
call randgen(exp,'EXPONENTIAL',1); /* fill 200 observations generated from exp(1) */

do j= 1 to n;
 wbl[j,1] = RAND('WEIBULL', b, a); /* fill 200 observations generated from Weibull(a,b) */
 logn[j,1] = RAND('LOGNORMAL'); /* fill 200 observations generated from lognormal(0,1) */
end;

print exp wbl logn Par;

/*Sort column vectors for order statistics*/
expOrder=exp;
b = expOrder;
expOrder[rank(expOrder),] = b;
```

```
wblOrder=wbl;
b = wblOrder;
wblOrder[rank(wblOrder),] = b;


lognOrder=logn;
b = lognOrder;
lognOrder[rank(lognOrder),] = b;


ParOrder=Par;
b = ParOrder;
ParOrder[rank(ParOrder),] = b;

print expOrder wblOrder lognOrder ParOrder;
OrderedData=expOrder||wblOrder||lognOrder||ParOrder;

create dataExpQQ from OrderedData;
append from OrderedData;

create dataParQQ from OrderedData;
append from OrderedData;

create datawblNew from wblOrder; /*Just added new for WBL col vector 200 observations*/
append from wblOrder;
quit;
proc export data=dataExpQQ outfile="C:\Users\Kuselo Kusi Ntsaluba\Dropbox\Research\
Proposal and Final  report drafts\SAS\data1" dbms=xlsx replace;
run;

proc export data=dataParQQ outfile="C:\Users\Kuselo Kusi Ntsaluba\Dropbox\Research\
Proposal and Final  report drafts\SAS\data2" dbms=xlsx replace;
run;

/*Just added new for WBL col vector 200 observations*/
proc export data=datawblNew outfile="C:\Users\Kuselo Kusi Ntsaluba\Dropbox\Research\
Proposal and Final  report drafts\SAS\dataWbl" dbms=xlsx replace;
run;

/*-----------------------------------------------------------------------------*/
/*--------------For simulation of 5000 values for applications-----------------*/

proc iml;
m=5000;
k=100;
NormalVec=J(m,1);
GammaVec=J(m,1);
ExpVec=J(m,1);

ExpVecGreater=J(k,1);
WblVecGreater=J(k,1);
lognVecGreater=J(k,1);
ParVecGreater=J(k,1);
```

```
NormPercentile=Quantile('NORMAL',0.95,3.5,2);
GammaPercentile=Quantile('Gamma',0.95,2,1);
ExpPercentile=Quantile('Exponential',0.95,1);
print NormPercentile GammaPercentile ExpPercentile;

/*For simulation of 5000 values for Normal*/

do i=1 to 5000;
 find=0;

 do until (find=1);
  x=RAND('NORMAL',3.5 , 2);
  if (x>0 & x<=NormPercentile) then find=1;
 end;

NormalVec[i,1]=x;
end;
print NormalVec;

/*For simulation of 5000 values for Gamma*/

do i=1 to 5000;
 find=0;

 do until (find=1);
  x=RAND('GAMMA',2 , 1);
  if (x<=GammaPercentile) then find=1;
 end;

GammaVec[i,1]=x;
end;
print GammaVec;

/*For simulation of 5000 values for Exponential*/

do i=1 to 5000;
 find=0;

 do until (find=1);
  x=RAND('EXPONENTIAL', 1);
  if (x<=ExpPercentile) then find=1;
 end;

ExpVec[i,1]=x;
end;
print ExpVec;

/*For simulation of 100 values for Exponential tail against Normal*/

do i=1 to 100;
 find=0;
```

```
 do until (find=1);
  x=RAND('EXPONENTIAL', 3);
  if (x>=NormPercentile) then find=1;
 end;

ExpVecGreater[i,1]=x;
end;
print ExpVecGreater;

/*For simulation of 100 values for Weibull tail against Normal*/

do i=1 to 100;
 find=0;

 do until (find=1);
  x=RAND('WEIBULL', 0.5, 2);
  if (x>=NormPercentile) then find=1;
 end;

WblVecGreater[i,1]=x;
end;
print WblVecGreater;

/*For simulation of 100 values for Lognormal tail against Normal*/

do i=1 to 100;
 find=0;

 do until (find=1);
  x=RAND('LOGNORMAL');
  if (x>=NormPercentile) then find=1;
 end;

lognVecGreater[i,1]=x;
end;
print LognVecGreater;

/*For simulation of 100 values for Pareto tail against Normal*/

do i=1 to 100;
 find=0;

 do until (find=1);
  call randgen(x,'PARETO',2.5,1);
  if (x>=NormPercentile) then find=1;
 end;

ParVecGreater[i,1]=x;
end;
print ParVecGreater;

quit;
/*-------------------------------SAS Code end-----------------------------------*/
```

MATLAB Code [26]

```
function ExpQQPlot(xdata)
%xdata is already ordered;
y = xdata;
n = size(xdata,1);
x = zeros(n,1);
for i = 1 : n
    x(i) = -log(1-i/(n+1));
end
plot(x,y);
 xlabel('-log(1-(j/(n+1)))')
 ylabel('X_j_:_n')
 title('exponential QQ plot')
 grid on
%Mean excess Values:
e = zeros(n,1);
for k = 1 : n-1
    e(k) = 0;
    for j = 1 : k
        e(k) = e(k) + xdata(n-j+1);
    end
    e(k) = e(k)/k - xdata(n-k);
end
   yreverse = zeros(n,1);
for k = 1 : n-1
  yreverse(k) = y(n-k);
end
figure
plot(yreverse,e,'*');
xlabel('X_n_-_k_:_n')
 ylabel('e_k_:_n')
 title('mean excess plot')
 grid on
end


function ParetoQQPlot(xdata)
%xdata is already ordered;
y = log(xdata);
n = size(xdata,1);
x = zeros(n,1);
for i = 1 : n
    x(i) = -log(1-i/(n+1));
end
plot(x,y);
 xlabel('-log(1-(j/(n+1)))')
 ylabel('log(X_j_:_n)')
 title('Pareto QQ plot')
 grid on
%Hill plot Values:
e = zeros(n,1);
for k = 1 : n-1
    e(k) = 0;
```

```
    for j = 1 : k
        e(k) = e(k) + log(xdata(n-j+1));
    end
    e(k) = e(k)/k - log(xdata(n-k));
end
   yreverse = zeros(n,1);
for k = 1 : n-1
  yreverse(k) = (y(n-k));
end
figure
plot(yreverse,e,'*');
 xlabel('log(X_n_-_k_:_n)')
 ylabel('H_k_:_n')
 title('Hill plot')
 grid on
end
```

```
%Code for data of Initial distributions (3 densities and parameters);
%MATLAB
x=0:0.1:350000;
z=0:0.1:700000;

y1=(normpdf(x,190093.2401,81585.08159));

syms betaGam
% BetaG=double(solve(0.95==gamcdf(350000,2,betaGam)))
BetaG=73779; % such that probability=0.95 when a=2
probability=gamcdf(350000,2,73779)
y2=gampdf(x,2,BetaG);

syms l
% lamda=double(solve(0.95==1-exp(-350000*l)))
y3=exppdf(x,110000);

figure
plot(x,y1,'Linestyle','-','Color','r','Linewidth',2)
hold on
plot(x,y2,'Linestyle','--','Color','b','Linewidth',2)
hold on
plot(x,y3,'Linestyle',':','Color','g','Linewidth',2)
legend({'Normal','Gamma','Exponential'},'Location','NorthEast');
hold off
title('Initial distributions (<350000)')
xlabel('x')
ylabel('f(x)')
grid on

% B=unifrnd(0,1,5,1)
```

```
%Code for data tail distributions (4 densities and parameters);MATLAB
z=0:0.000005:6;
w=0:0.5:600000;
size(z)
size(w)
```

```
%Exponential Tail Distrubtion%
Mean=0.5;
y2 = pdf('exponential', z, Mean);
figure
plot(w+350000,y2,'Linestyle','-.','Color','b','Linewidth',2)

%Weibull Tail Distribution%
lambda = 0.5;
k = 0.8;
y3 = pdf('weibull',z,lambda,k);
hold on
plot(w+350000,y3,'Linestyle','-.','Color','r','Linewidth',2)


%Pareto Tail Distribution%
minn = 1.8;
alpha = 0.0001;
y4 = pdf('Generalized Pareto',z,alpha,minn,0);
hold on
plot(w+350000,y4,'Linestyle','-.','Color','g','Linewidth',2)


%Lognormal Tail Distribution%
sigma = 1.1;
mu = 0.1;
y5 = pdf('Lognormal',z,mu,sigma);
plot(w+350000,y5,'Linestyle','-.','Color','c','Linewidth',2)



figure
plot(w+350000,y2,'Linestyle','-.','Color','b','Linewidth',2)
hold on
plot(w+350000,y3,'Linestyle',':','Color','g','Linewidth',2)
hold on
plot(w+350000,y4,'Linestyle','-','Color','r','Linewidth',2)
hold on
plot(w+350000,y5,'Linestyle','--','Color','c','Linewidth',2)
legend({'Exponential','Weibull','Pareto','Lognormal'},'Location','NorthEast');
hold off
title('Comparison of tails')
xlabel('x')
ylabel('f(x)')
grid on
axis([350000 950000 0 2.5]);

% ///Code for Random number generation (3 initial distributions) MATLAB///
n=5000;
count=0;
randUniform=zeros(n,1);
randExp=zeros(n,1);
randGamma=zeros(n,1);
```

```
randNorm=zeros(n,1);
while count<n
    uni=unifrnd(0,1,1,1); % generate 1 value from uniform(0,1)
    if uni<=0.958
        count=count+1;
        randUniform(count)=uni;
        randExp(count)=expinv(uni,110000); %generate Exp(110000) values
    end
end

count=0;
while count<n
    uni=unifrnd(0,1,1,1); % generate 1 value from uniform(0,1)
     if uni<=0.95
        count=count+1;
        randUniform(count)=uni;
        randGamma(count)=gaminv(uni,2,73779); %generate gamma(2,73779) values
     end
end

count=0;
while count<n
    uni=unifrnd(0,1,1,1); % generate 1 value from uniform(0,1)
    if uni<=0.974
        count=count+1;
        randUniform(count)=uni;
        if norminv(uni,190093.2401,81585.08159)<0
        randNorm(count)=abs(norminv(uni,190093.2401,81585.08159));
       %generate N(190093.2401,81585.08159) values
        else randNorm(count)=norminv(uni,190093.2401,81585.08159);
        end
    end
end

% %Initial Histograms
% figure
% hist(randExp,50)
% title('Exponential simulated data');
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% figure
% hist(randGamma,50)
% title('Gamma simulated data');
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% figure
% hist(randNorm,50)
% title('Normal simulated data');
% xlabel('Claim amounts')
% ylabel('Number of Claims')

%///////Code for Random number generation (4 tail distributions)///////////
m=100;
```

```
count=0;
randUniform=zeros(m,1);
randExp2=zeros(m,1);
randWbl=zeros(m,1);
randLogn=zeros(m,1);
randPar=zeros(m,1);
while count<m
   uni=unifrnd(0,1,1,1); % generate 1 value from uniform(0,1)
   count=count+1;
   randUniform(count)=uni;
    randExp2(count) = 100000*icdf('exponential', uni, 0.5)+350000;
    %generate scaled Exp(0.5) value
end

count=0;
while count<m
    uni=unifrnd(0,1,1,1); % generate 1 value from uniform(0,1)
    count=count+1;
    randUniform(count)=uni;
    randWbl(count) = 100000*icdf('Weibull', uni, 0.5, 0.8)+350000;
    %generate scaled wbl(0.5,0.8) value
end

count=0;
while count<m
    uni=unifrnd(0,1,1,1); % generate 1 value from uniform(0,1)
    count=count+1;
    randUniform(count)=uni;
    randLogn(count) = 100000*icdf('Lognormal', uni, 0.01, 1.5)+350000;
    %generate scaled Logn(0.01,1.5) value
end

count=0;
while count<m
    uni=unifrnd(0,1,1,1); % generate 1 value from uniform(0,1)
    count=count+1;
    randUniform(count)=uni;
    randPar(count) = 100000*icdf('Generalized Pareto', uni, 0.0001, 1.8, 0)+350000;
    %generate GP(0.0001,1.8,0) value
end


% %Tail Histograms
% figure
% hist(randExp2,50)
% title('Exponential simulated data');
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% xlim auto
% figure
% hist(randWbl,50)
% title('Weibull simulated data');
% xlabel('Claim amounts')
```

```
% ylabel('Number of Claims')
% xlim auto
% figure
% hist(randLogn,50)
% title('Lognormal simulated data');
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% xlim auto
% figure
% hist(randPar,50)
% title('Pareto simulated data');
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% xlim auto

%Normal spliced vectors
NormExp=sort(vertcat(randNorm,randExp2));
NormWbl=sort(vertcat(randNorm,randWbl));
NormLogn=sort(vertcat(randNorm,randLogn));
NormPar=sort(vertcat(randNorm,randPar));

%Exponential spliced vectors
ExpExp=sort(vertcat(randExp,randExp2));
ExpWbl=sort(vertcat(randExp,randWbl));
ExpLogn=sort(vertcat(randExp,randLogn));
ExpPar=sort(vertcat(randExp,randPar));

%Gamma spliced vectors
GammaExp=sort(vertcat(randGamma,randExp2));
GammaWbl=sort(vertcat(randGamma,randWbl));
GammaLogn=sort(vertcat(randGamma,randLogn));
GammaPar=sort(vertcat(randGamma,randPar));

% %Normal Spliced Histograms
% figure
% hist(NormExp,50)
% title('Normal spliced with Exp')
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% figure
% hist(NormWbl,50)
% title('Normal spliced with Wbl')
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% figure
% hist(NormLogn,50)
% title('Normal spliced with LogN')
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% figure
% hist(NormPar,50)
% title('Normal spliced with Par')
% xlabel('Claim amounts')
```

```
% ylabel('Number of Claims')
%
% %Exponential Spliced Histograms
% figure
% hist(ExpExp,50)
% title('Exp spliced with Exp')
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% figure
% hist(ExpWbl,50)
% title('Exp spliced with Wbl')
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% figure
% hist(ExpLogn,50)
% title('Exp spliced with LogN')
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% figure
% hist(ExpPar,50)
% title('Exp spliced with Par')
% xlabel('Claim amounts')
% ylabel('Number of Claims')

% %Gamma Spliced Histograms
% figure
% hist(GammaExp,50)
% title('Gamma spliced with Exp')
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% figure
% hist(GammaWbl,50)
% title('Gamma spliced with Wbl')
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% figure
% hist(GammaLogn,50)
% title('Gamma spliced with LogN')
% xlabel('Claim amounts')
% ylabel('Number of Claims')
% figure
% hist(GammaPar,50)
% title('Gamma spliced with Par')
% xlabel('Claim amounts')
% ylabel('Number of Claims')
```

The analysis of multilevel models for longitudinal data

Mbali C Ntuli 13155165

STK795 Research Report

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

Supervisor: Dr. G Crafford

Department of Statistics, University of Pretoria

2 November 2016 (final)

Abstract

The analysis of multilevel models is based on looking at different regression models with different explanatory variables at different levels. There are various ways to analyse the multilevel models, one of which is the use of SAS PROC MIXED. There are 2 types of multilevel analysis. Hierarchical data found commonly in educational and clinical research settings as well as longitudinal data, which represents models of individuals over time. The level 1 units can be explained by regression over time, which are commonly dependent, however, other there may exist secondary, level 2 units, which are additional explanatory variables and influence the regression outcome.

Declaration

I, *Mbali Chantelle Ntuli,* 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.

_ _

Mbali Chantelle Ntuli

_ _

Dr.G Crafford

_ _

Date

Acknowledgements

Mbali Chantelle Ntuli would like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a post graduate bursary.

The data analysis for this essay was performed using SAS software, Version 9.4 of the SAS system for Windows. Copyright © 2016 SAS Institution Inc. Cary, NC, USA.

Contents

1 Introduction **6**

2 Background Theory **6**

3 Application **7**
3.1 Concepts 7
3.1.1 Unconditional means model: completely random, no time or explanatory variables influence 7
3.1.2 Unconditional linear growth model 9
3.1.3 Conditional linear growth model 10
3.1.4 Conditional linear growth model : With fixed slopes 11
3.1.5 SAS Output 12

4 Conclusion **13**

References **14**

List of Figures

1 Unconditional means model 8
2 Unconditional linear growth model 10
3 Conditional linear growth model 11
4 **Conditonal growth model with fixed slopes** 12

List of Tables

1 Introduction

Multilevel models are models which can be analyzed using different regression models, considering both fixed and random variables and eventually creating one regression equation which has integrated all variables. The aim of the research paper is to answer the research question: What is the influence of the level 2 explanatory variables on the slopes and intercepts over time? The research paper will be looking at 2 level growth model analysing the depression levels of individuals that are unemployed between certain periods of time. It is essential to look at the detailed levels because ignoring details can lead to inflated type 1 error rates, as well as errorneous reading and interpretation of statistical significance tests[1]. The models that are generally formulated from mulitlevel models are models with varying intercepts or models with varying intercepts and slopes.

This paper is investigating the level of depression level (CESD score) between a sample of people who have not been employed for 1 month, 5 months, and 11 months. At level 1 we look at the effect on the CESD score over time, while at level 2, the explanetory variable, in this case is whether a person is unemployed or not will be added in order to crate a combined model. This paper will discuss 4 types of models for longitudinal data. First, the Unconditional means model, the Unconditional growth model, and the Conditional growth model, and lastly, the Conditional growth model with fixed slopes.. Each model will have level 1 and level 2 influences and we will produce combined model estimates with both fixed and random effects from the explanatory variables. For all of the models we make the following assumptions:

$$\boldsymbol{\epsilon}_i \quad \sim \quad N(\mathbf{0}, \sigma^2 I_3)$$

,and

$$u_{ij} \quad \sim \quad N(0, \tau_{00})$$

These assumptions are based on [11, 10].

The paper will then calculate and interprete the variance components σ^2 and Φ, which are the "between-person" and "within-person" variances respectively.

2 Background Theory

Origin of Multilevel analyis The first step concerning multilevel analysis was in the United States of America in the 1940's. It was referred to as contextual analysis and the first statistical techniques on contextual analysis were by [7] where and [8] . In the early 1970's the development of multilevel analysis started and it took part in schools [9]. The innovation was to analyse each school seperately and the dependent variable would be marks for a certain subject with explanatory variables such as gender or parents socioeconomic status. An estimation for a identical regression models for each school therefore yeilded a set of intercepts and regression coefficients that showed the systematic variation between schools[ESS99]. This led to the slopes-as-approach was looked at as a two-stage multiple regression.

Innovation of Multilevel model According to [10] the analysis of multilevel models is essential in order to know he effects of a model with different levels. Not everything is as simple and linear as it seems. Most cases have other influences that form part of the building blocks that make up the main or combined model. [10] shows that time is a predictor of regression models, especially growth models dependent on the lapse of time.

[11] further explains how using SAS PROC MIXED can analyse multilevel hierarchical linear models as well as individual growth models. A simple explainantion of fixed effects can be effects that cannot be altered and are constant, such as time whereas the random effects can be influenced and usually change. These random effects are gerally in the form of dummy variables. One of the comparative techniques [10] uses is two-stage generalized least squares model in order to estimate the linear regression of an individual outcome

on a group in studies of multilevel data. The difference from ordinary least squares (OLS) estimation is that they assume an intra-class correlation among the errors within the group. Further research by [6] shows the different methods used to perform a multilevel analysis, which includes analysing residuals, examining slope variation using ordinary least squares, doing intraclass correlations, as well as hypothesis testing and other estimation methods. [6] analyses multilevel regression by using different softwares, such as HLM, MPlus, Stata, and SAS to name a few. He uses the approach of multilevel structural equation modelling (MSEM). First, he takes a sample of units from the higher level (e.g. schools), and next he samples the sub-units from the available units (e.g. sample pupils from the schools). In these samples, the individual observations are generally not independent. [6] states: "Pupils in the same school tend to be similar to each other, because of selection processes and because of common history the pupils share by going to the same school." The resulting intraclass correlation between variables measured on pupils from the same schools is higher than the average correlation measured on pupils from different schools.

[2] States that multilevel analysis in sociology is individual analysis based on different levels caused by context. Multilevel models are also referred to as contextual models. They identify the link by showing different social science research used to analyse multilevel models while [5] states that nonlinear multilevel models exist and usually occur when modelling discrete data. He shows how to linearize nonlinear multilevel data and do estimations on the data by using iterative generalized least squares estimation which was taken to be equivalent to maximum likelihood.

The innovation of this paper This paper will extract the unemployment data from [10] notes, where [10] captures the depression levels of individuals who are unemployed over time. specifically on the Chapter 5 analysis in her book. The months of the dataset have been altered for the purpose of clearly defining the timeline of the investigation. The dataset was provided in the form of a person-period dataset, with 674 observations. In this paper, we manipulated the dataset to make it more simpler to analyse by setting 3 time intervals, 1 month, 5 months, and 11 months. We also want to work with complete data so we changed the dataset from a person-period dataset to a period-level dataset. Once the dataset was changed, we deleted all missing fields as well as individuals who do not have CESD scores for all 3 months. Once the data was made complete, we reverted the dataset back to a person-period dataset and we eventually had 579 complete observations. The SAS coding is provided in the appendix.

We then use SAS Proc Mixed to analyse the the random and fixed effects of time on the level 1 regression model, the unemployment effect and time on the level 2 regression, and finally combine level 1 and level 2 model to create a multilevel regression model with all the effects. The results in this paper will be displayd in a matrix format and the parameters and variances will be discussed.

The paper will also investigate the intra class correlation between the CESD scores, months, and unemployment in the Unconditional means model.

3 Application

3.1 Concepts

3.1.1 Unconditional means model: completely random, no time or explanatory variables influence

The Unconditional means model is a multilevel model, with level 1 and level 2 effects, however, all the variables are completely random. The model equation is:

Level 1:

$$\mathbf{y}_i = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} b_{0i} + \boldsymbol{\epsilon}_\mathbf{i}$$

Level 2:

$$b_{0i} = \gamma_{00} + u_{0j}$$

Combined model:

$$\hat{\mathbf{y}}_i = \mathbf{1}(\gamma_{00} + u_{0j}) + \epsilon_i$$

$$\hat{\mathbf{y}}_i = \gamma_{00}\mathbf{1} + u_{0j}\mathbf{1} + \boldsymbol{\epsilon}_i$$

Estimates:
•Fixed
$y_i = \hat{\gamma_{00}}\mathbf{1} = (14.9223)\mathbf{1}$

The results show that on average, an unemployed person has a CESD score of 14.92.

•Variance components
$\hat{\Phi} = \tau_{00} \hat{=} 58$
$\hat{\sigma^2} = 80$

•Intraclass Correlation Coefficient
$\hat{\rho} = \frac{\hat{\tau_{00}}}{\hat{\tau_{00}}+\hat{\sigma^2}} = \frac{58}{58+80} = 0.42$

Figure 1 represents individual models that have been fitted to explain the CESD scores over time. The y-axis represents the time in months.

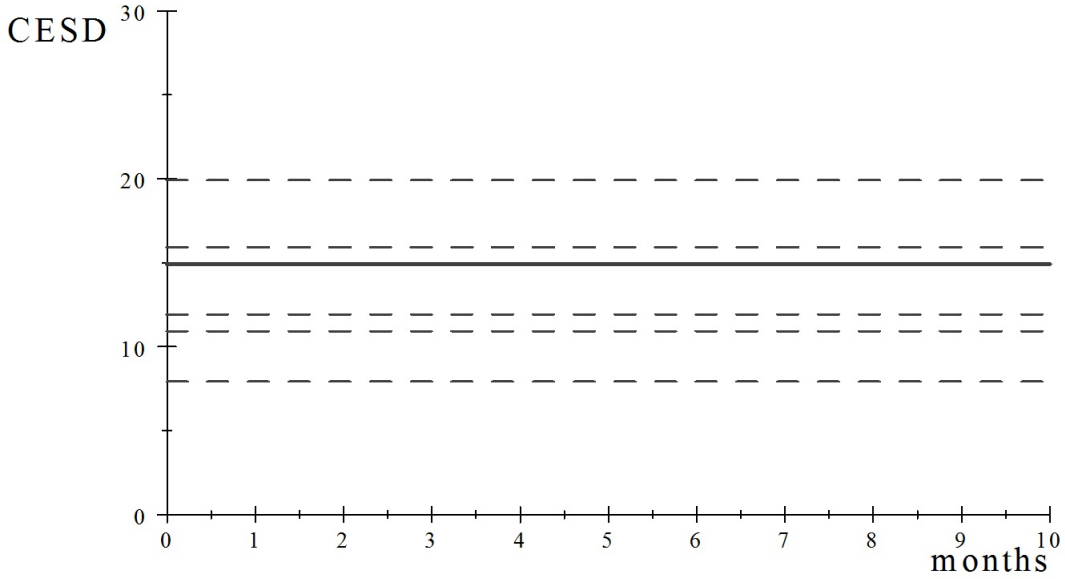


Figure 1: Unconditional means model

3.1.2 Unconditional linear growth model

In this model, we only consider the influence of time. It is essential to know that $\mathbf{X}_i = 1, 5, 11$. And in order to enable a starting point for the intercepts, we create the variable $\mathbf{t}_i = 0, 4, 10$, therefore for the rest of this paper we will consider $\mathbf{t_i}$.

Level 1:

$$y_i = \begin{pmatrix} 1 & 0 \\ 1 & 4 \\ 1 & 10 \end{pmatrix} \begin{pmatrix} b_{0i} \\ b_{1i} \end{pmatrix} + \boldsymbol{\epsilon}_i$$

Level 2:
$\begin{pmatrix} b_{0i} = \gamma_{00} + u_{0j} \\ b_{1i} = \gamma_{10} + u_{1j} \end{pmatrix}$
, where

$$\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix}\right)$$

Combined model:

$$\mathbf{y_i} = \begin{pmatrix} \mathbf{1} & \mathbf{t_i} \end{pmatrix} \begin{pmatrix} \gamma_{00} + u_{0j} \\ \gamma_{00} + u_{1j} \end{pmatrix} + \boldsymbol{\epsilon_i}$$

$$\mathbf{y_i} = \gamma_{00}\mathbf{1} + \gamma_{10}\mathbf{t_i} + u_{0j}\mathbf{1} + u_{1j}\mathbf{t}_i + \boldsymbol{\epsilon}_i$$

Estimates:
•Fixed
$\hat{\boldsymbol{y}}i = \hat{\gamma_{00}}\mathbf{1} + \hat{\gamma_{10}}\boldsymbol{t}_i = (16.97)\mathbf{1} - 0.439\boldsymbol{t}_i$
The results show that on average, on the first day of unemployment, the CESD score is 16.97 and the CESD score declines by 0.439. We notice that when we include the influence of time to the unconditional means model, the within person variation decreases from 80 to 67.

•Variance components
$\hat{\Phi} = \begin{pmatrix} \hat{\tau_{00}} & \hat{\tau_{01}} \\ \hat{\tau_{10}} & \hat{\tau_{11}} \end{pmatrix} = \begin{pmatrix} 76.25 & -2.29 \\ -2.29 & 0.359 \end{pmatrix}$
$\hat{\sigma^2} = 67$

Figure 2 displays the fitted models fitted above. We can see that the intercepts are randomised and the slopes are different.

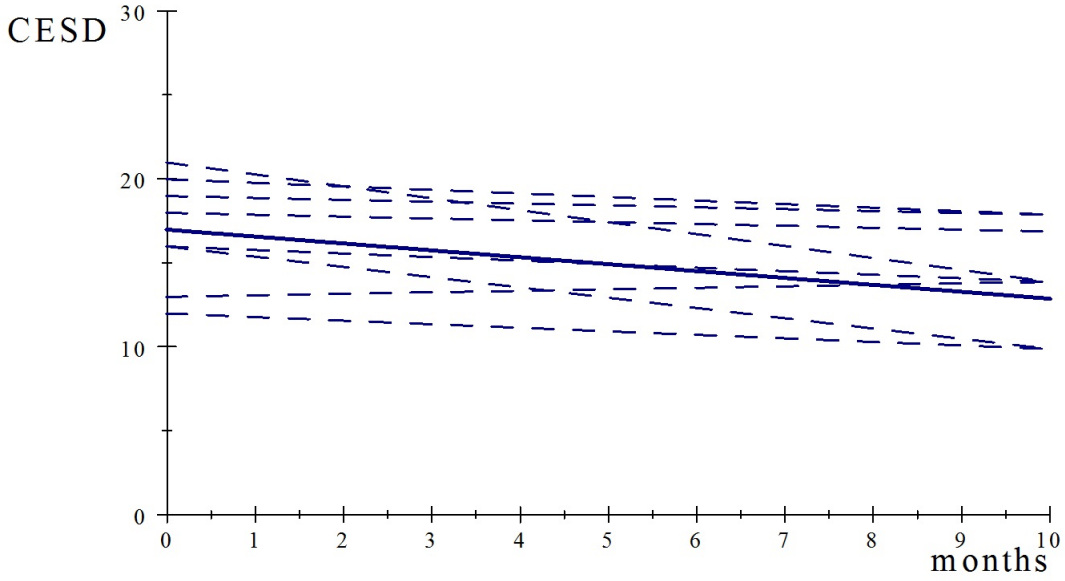


Figure 2: Unconditional linear growth model

3.1.3 Conditional linear growth model

This model allows for the effect of unemployment to vary over time.

Level 1:

$$y_i = \begin{pmatrix} 1 & 0 \\ 1 & 4 \\ 1 & 10 \end{pmatrix} \begin{pmatrix} b_{oi} \\ b_{1i} \end{pmatrix} + \boldsymbol{\epsilon}_i$$

Level 2: $\begin{pmatrix} b_{0i} = \gamma_{00} + z_i\gamma_{01} + u_{0j} \\ b_{0i} = \gamma_{10} + z_i\gamma_{11} + u_{1j} \end{pmatrix}$, where $\begin{pmatrix} u_{oj} \\ u_{0j} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{pmatrix}\right)$

Combined model:

$$\mathbf{y_i} = \begin{pmatrix} \mathbf{1} & \mathbf{t}_i \end{pmatrix} \begin{pmatrix} \gamma_{00} + z_i\gamma_{01} + u_{0j} \\ \gamma_{10} + z_i\gamma_{11} + u_{1j} \end{pmatrix} + \boldsymbol{\epsilon}_i$$

$$\mathbf{y_i} = \gamma_{00}\mathbf{1} + \gamma_{01}\mathbf{1} + \gamma_{10}\mathbf{t_i} + \gamma_{11}z_i\mathbf{t}_i + u_{0j}\mathbf{1} + u_{1j}\mathbf{t}_i + \boldsymbol{\epsilon}_i$$

Estimates:
•Fixed
$\mathbf{y}_i = \hat{\gamma_{00}}\mathbf{1} + \hat{y_{01}z_i}\mathbf{1} + \hat{y_{11}z_i}\mathbf{t}_i$
$= 16.91 + 0.09z_i\mathbf{1} - 0.593\mathbf{t}_i + 0.307z_i\boldsymbol{t}_i$

Unemployed: $z_i = 1$
$\hat{\mathbf{y}}_i = 16.991 - 0.286\mathbf{t_i}$
Not Unemployed: $z_i = 0$
$\hat{\mathbf{y}}_\mathbf{i} = 16.991 - 0.593\mathbf{t}_i$

The results show that the CESD score of people who are unemployed is greater than the people who are employed. The purple lines in Figure 3 represent the unemployed individuals, while the blue lines show the CESD scores of employed individuals. The estimates also show that for an employed individual the CESD score, on average, decreases by 0.593, while for the unemployed, it decreases by 0.286.

It is important to notice that both employed and unemployed models start at the same intercept, 16.991 which can be explained by assuming that all individuals were unemployed in the first month. The within person variation is 67, and insignificant.

$$\hat{\Phi} = \begin{pmatrix} \hat{\tau_{00}} & \hat{\tau_{01}} \\ \hat{\tau_{10}} & \hat{\tau_{11}} \end{pmatrix} = \begin{pmatrix} 81.2 & -2.65 \\ -2.65 & 0.359 \end{pmatrix}$$

$$\hat{\sigma^2} = 67$$

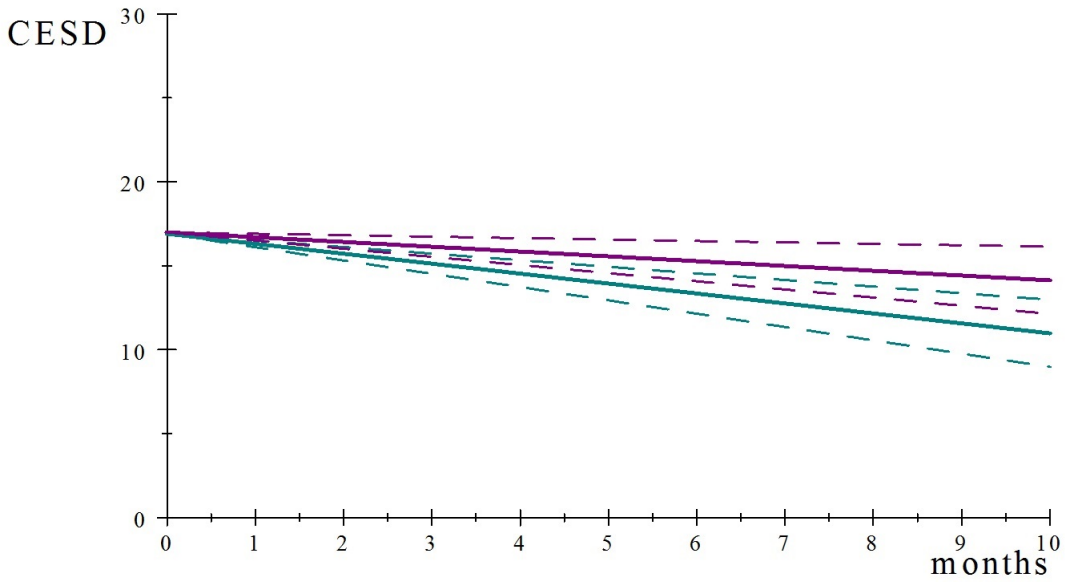


Figure 3: Conditional linear growth model

3.1.4 Conditional linear growth model : With fixed slopes

This model also allows for the effect of unemployment to vary over time, we then decided to hold the slopes fixed.

Level 1:

$$\mathbf{y_i} = \begin{pmatrix} 1 & 0 \\ 1 & 4 \\ 1 & 10 \end{pmatrix} \begin{pmatrix} b_{0i} \\ b_{1i} \end{pmatrix} + \boldsymbol{\epsilon}_i$$

Level 2:

$$\begin{pmatrix} b_{0i} = \gamma_{00} + zi\gamma_{01} + u_{0j} \\ b_{1i} = \gamma_{10} \end{pmatrix}$$

Combined model:

$$\mathbf{y_i} = \begin{pmatrix} \mathbf{1} & \mathbf{t_i} \end{pmatrix} \begin{pmatrix} \gamma_{00} + z_i\gamma_{01} + u_{0j} \\ \gamma_{10} + z_i\gamma_{11} + u_{1j} \end{pmatrix} + \boldsymbol{\epsilon_i}$$

$$= \gamma_{00}\mathbf{1} + \gamma_{01}z_i\mathbf{1} + \gamma_{10}\mathbf{t}_i + u_{0j}\mathbf{1} + \boldsymbol{\epsilon}_i$$

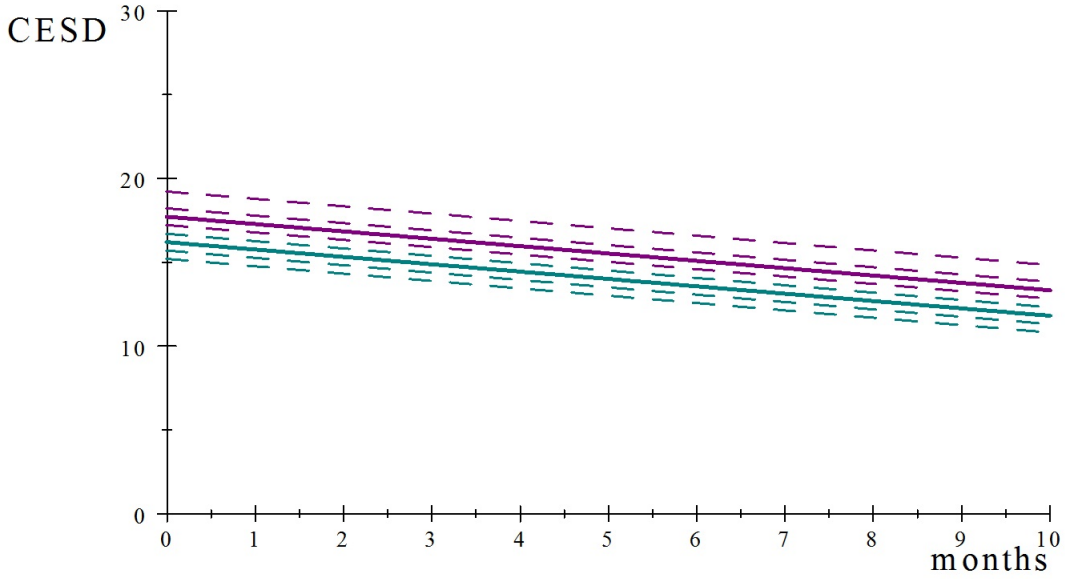


Figure 4: Conditonal growth model with fixed slopes

3.1.5 SAS Output

```
Model 1:Unconditional Means model
    Covariance Parameter Estimate
                                    Standard          Z
Cov Parm     Subject    Estimate       Error      Value      Pr > Z
UN(1,1)      id          58.0485      8.8741       6.54      <.0001
Residual                 80.4525      5.7911      13.89      <.0001

                                  Model 1:Unconditional Means model                    2
                                     Solution for Fixed Effects
                         Standard
Effect       Estimate       Error      DF    t Value    Pr > |t|
Intercept     14.9223      0.6631     192      22.50      <.0001
                              Model 2: Unconditional Linear Growth Model
     Covariance Parameter Estimates
                        Standard          Z
Cov Parm     Subject    Estimate       Error      Value        Pr Z
UN(1,1)      id          76.2509     13.9706       5.46      <.0001
UN(2,1)      id          -2.2975      1.3679      -1.68      0.0930
UN(2,2)      id           0.3598      0.2172       1.66      0.0488
Residual                 66.6687      6.7867       9.82      <.0001
```

Model 2: Unconditional Linear Growth Model

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	16.9725	0.8116	192	20.91	<.0001
months	-0.4393	0.09318	385	-4.71	<.0001

Model 3: Conditional Linear Growth Model

Covariance Parameter Estimates

Cov Parm	Subject	Standard Estimate	Z Error	Value	Pr Z
UN(1,1)	id	77.3084	14.1239	5.47	<.0001
UN(2,1)	id	-2.4257	1.3792	-1.76	0.0786
UN(2,2)	id	0.3598	0.2172	1.66	0.0488
Residual		66.6687	6.7867	9.82	<.0001

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	16.1376	1.0517	191	15.34	<.0001
months	-0.4393	0.09318	385	-4.71	<.0001
unemp	1.6611	1.3226	191	1.26	0.2107

Model 4: Conditional Linear Growth Model:With fixed slopes

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z
UN(1,1)	id	59.4704	8.8587	6.71	<.0001
Residual		75.7593	5.4604	13.87	<.0001

Model 4: Conditional Linear Growth Model:With fixed slopes

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	16.2064	1.0253	191	15.81	<.0001
months	-0.4393	0.08802	385	-4.99	<.0001
unemp	1.5242	1.3251	191	1.15	0.2515

4 Conclusion

The research question has been answered and displayed showing that both level 1 and level 2 effects have a influence on the combined outcome of the individual. In this paper it shows the basic intuition that the CESD score of an individual is expected to decrease, however, if the person is unemployed then they will have a higher CESD score, which in turn, are more depressed than an individual who is employed. There are exceptions however, where even though the individual is employed their depression levels still increase and that can be caused by other factors. This paper proves that it is essential to consider other underlying effects on a regression model in both hierarchical and longitudinal data.

References

[1] B.A Bell, M Ene, W Smiley, and J.A Schoeneberger. A multilevel model primer using sas proc mixed. In *SA Global Forum*, pages 0–19. Citeseer, 2013.

[2] TA DiPrete and JD Forristal. Multilevel models: methods and substance. *Annual Review of Sociology*, 20:331–357, 1994.

[3] C Duncan, K Jones, and G Moon. Context, composition and heterogeneity: using multilevel models in health research. *Social Science and Medicine*, 46(1):97–117, 1998.

[4] C Duncan, K Jones, and G Moon. Context, composition and heterogeneity: using multilevel models in health research. *Social Science and Medicine*, 46(1):97–117, 1998.

[5] Harvey Goldstein. Nonlinear mulilevel models, with an application to discrete response data. *Biometrika*, 78(1):45–51, March 1991.

[6] JJ Hox, M Moerbeek, and R Van de Schoot. *Multilevel Analysis: Techniques and Applications*. Routledge, 2010.

[7] P.F Lazarsfeld. *Problems in methodology*. 1959.

[8] P.F Lazarsfeld. On the relation between indicidual and collective properties. *Complex Organisations: A Sociological Reader*, 1961.

[9] C Schnaudt, M Weinhardt, R Fitzgerald, and S Liebig. A short history of multilevel analysis. *Schmollers Jahrbuch/Journal of Applied Social Science Studies*, 2014.

[10] J Singer and J.B Willet. *Applied Longitudinal Data Analysis: Modelling Change and Event Occurance*. Oxford university press, 2003.

[11] Judith D Singer. Using sas proc mixed to fit multilevel models, hierarchical models, and individual growth models. *Journal of Educational and Behavioural Statistics*, 23(4):323–355, 1998.

Graphics Program

```
libname research 'G:\Judith Singer\ALDA'; **Create new permanent library;
run;
data research.unemployment;
infile "G:\Judith Singer\ALDA\Research\unemployment.csv"
pad missover dsd dlm=',' lrecl=300 firstobs=2;
input id
months
cesd
unemp
;
run;
data research.depression;
set research.unemployment;
interaction=months*unemp;
run;
proc sgplot data=research.depression (rename=(unemp=unemployment)) noautolegend ;
yaxis min = 0 max = 4;
reg x=months y=unemployment
/ group = id nomarkers LINEATTRS = (COLOR= gray PATTERN = 1 THICKNESS = 1) ;
reg x=months y=unemployment
/ nomarkers LINEATTRS = (COLOR= red PATTERN = 1 THICKNESS = 3) ;
run;
quit;
```

Data manipulation

```
data clean;
array tvar[3] t1-t3;
array cesdvar[3] cesd1-cesd3;
do i=1 to 3 until (last.id);
set a;
by id;
tvar [i]=t;
cesdvar[i]=cesd;
end;
drop i  cesd ;
run;
data research.b;
set clean;
if nmiss(cesd1,cesd2,cesd3)>0 then delete;
run;
data research.c;
set research.b;
array tvar [3] t1-t3;
array cesdvar[3] cesd1-cesd3;
do i=1 to 3;
t=i;
if t=2 then  t=5;
if t=3 then t=11;
```

```
\subsection*{Regression Coding}

\begin{verbatim}
title "Model 1:Unconditional Means model";
proc mixed data = research.c noclprint covtest;
class id;
model cesd =   /solution ddfm = bw;
random intercept  / subject = id type = un;
run;
title "Model 2: Unconditional Linear Growth Model";
proc mixed data = research.c covtest noclprint;
class id;
model cesd = months /solution ddfm = bw ;
random intercept months / subject = id type = un;
run;
title "Model 3: Conditional Linear Growth Model";
proc mixed data = research.c covtest noclprint;
class id;
model cesd = months unemp /solution ddfm = bw ;
random intercept months / subject = id type = un;
run;
title "Model 4: Conditional Linear Growth Model:With fixed slopes";
proc mixed data = research.c covtest noclprint;
class id;
model cesd = months unemp /solution ddfm = bw ;
random intercept  / subject = id type = un;
run;
```

Properties of the geometric Poisson distribution

Nozipho Nyathi 12045502

WST795 Research Report

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

Supervisor: Dr R. Ehlers

Department of Statistics, University of Pretoria

2 November 2016

Abstract

In this study we compare properties of the geometric Poisson and different geometric weighted Poisson distributions in order to get a greater understanding of the geometric Poisson distribution. The focus will be on analysing the dispersion (variability) of the geometric Poisson in relation to two geometric weighted Poisson distributions using the Fisher index as a measure of dispersion.

Declaration

I, *Nozipho M.Z Nyathi* 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.

_____
Nozipho M.Z Nyathi

_____
Dr R. Ehlers

_____
Date

Acknowledgments

The author would like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a postgraduate bursary.

Contents

1 Introduction **6**

2 Preliminary results **7**

3 The geometric Poisson distribution **8**

4 The geometric weighted Poisson distribution **10**
4.1 The general case of the geometric weighted Poisson distribution 11
4.2 A geometric weighted Poisson distribution with weight function$w\left(n\right)=n$ 14
4.3 A geometric weighted Poisson distribution with weight function $w(n)=\frac{1}{n+1}$ 16

5 Summary of the results **20**

6 Graphical comparison of the distributions **20**
6.1 The probability mass functions 21
6.2 The Fisher indices 23

7 Conclusion **23**

References **25**

Appendix **26**

List of Figures

1 The Probability mass functions under different parameter values 22
2 The graphical representation of the Fisher indices 23

List of Tables

1 Summary of the results. 20
2 Numerical summary of statistics for $\theta=0.5$ and λ varying 21
3 Numerical summary of statistics for $\lambda=6$ and θ varying 21

1 Introduction

The geometric Poisson distribution is a special case of a compound Poisson distribution with each term geometrically distributed. The geometric Poisson distribution is defined as

$$X = \sum_{i=1}^{N_\lambda} Yi$$

where N_λ is a Poisson random variable with parameter $\lambda > 0$ and Y_i, $i = 1, 2, 3, ...$ are i.i.d. geometric random variables, independent of N_λ. This distribution is also known as the Polya-Aeppli distribution; see Johnson et al. [8].

This distribution has been applied in several fields hence it is important to study and understand its properties. There are many real life examples for which the geometric Poisson distribution has been used. Randolph and Sahinoglu [12] illustrated the importance of the geometric Poisson distribution in controlling defects in softwares, and Robin [13] and Robin et al. [14] used the geometric Poisson distribution to model the distribution of overlapping word occurrences. Chen et al. [4] showed that the geometric Poisson distribution can be used in process control to come up with a geometric Poisson CUSUM control chart. The geometric Poisson distribution was also used by Rosychuk et al. [15] to explain DNA substitution by assuming that the substitution events were Poisson distributed whilst the number of substitutions per event were geometrically distributed.

In order to learn more about this distribution we will compare the geometric Poisson distribution (GPD) to two geometric weighted Poisson distributions (GWPDs). In this essay the weight functions $w(n) = n$ and $w(n) = \frac{1}{n+1}$ will be considered; see Minkova and Balakrishnan [9].

For the GWPD we have

$$M^w = \sum_{i=1}^{N_\lambda^w} Yi$$

where N_λ^w has a weighted Poisson distribution with parameter $\lambda > 0$ and Y_i, $i = 1, 2, 3, ...$ are i.i.d. geometric random variables, independent of N_λ^w. The probability mass function (pmf) of the weighted version of the Poisson distribution is given by

$$f^w(n) = P(N_\lambda^w = n) = \frac{w(n)f(n)}{E[w(N_\lambda)]} \qquad for\ \ n = 0, 1, 2, ...$$

where $N_\lambda \sim POI(\lambda)$, $E[w(N_\lambda)] = \sum_{n=0}^{\infty} w(n)\frac{\lambda^n e^{-\lambda}}{n!}$ and $f(n)$ is the pmf of N_λ.

To calculate and compare the dispersion of the GPD and the GWPD we will use the Fisher index of dispersion defined as $FI(X) = \frac{var(X)}{E(X)}$ which is used to measure the variability of a set of observed values compared to a standard statistical model. A distribution is over-dispersed if $FI(X) > 1$, equi-dispersed if $FI(X) = 1$ and $FI(X) < 1$; see Minkova and Balakrishnan [9].

Anwar and Ahmad [2] derived several properties of the GPD including the survival function and Ata and Ozel [3] derived the survival functions for the geometric Poisson process. Minkova and Balakrishnan [9] derived the compound weighted Poisson distribution and went on to derive the Fisher index of dispersion of the distributions for different weight functions as well as derived some properties of the weighted Poisson distributions through the analysis of the Fisher index. Özel and İnal [11] derived the explicit probability function of the GPD and used it in the computation of the probabilities. They also used the probability generating function of the GPD to calculate the moments needed to find the Fisher index. Özel and İnal [11] also looked at the application and numerical examples of the GPD using traffic accident data.

2 Preliminary results

In this section we list the definitions of distributions, its properties and important statistical results that will be used in the study.

Definition 1. Let Y be a geometric distributed random variable with parameter $0 < \theta \leq 1$ and probability mass function given by

$$P(Y = j) = p_j = \theta(1-\theta)^{j-1},\ j = 1, 2, 3, ... \tag{1}$$

where $E(Y) = \frac{1}{\theta}$ and $V(Y) = \frac{1-\theta}{\theta^2}$. Then the probability generating function of Y is given by

$$g_Y(s) = E[s^Y] = \sum_{j=1}^{\infty} s^j \theta(1-\theta)^{j-1} = \frac{\theta s}{1-(1-\theta)s}. \tag{2}$$

See Johnson et al. [8].

Theorem 2. *Suppose* Y_i, $i = 1, 2, 3, ..n$ *are i.i.d.* $GEO(\theta)$. *Then* $\sum_{i=1}^{n} Y_i \sim NB(n, \theta)$ *and*

$$P\left(\sum_{i=1}^{n} Y_i = y\right) = \left(\begin{array}{c} y-1 \\ n-1 \end{array}\right) \theta^n (1-\theta)^{y-n}. \tag{3}$$

Definition 3. Let N_λ be a Poisson distributed random variable with parameter $\lambda > 0$ and probability mass function given by

$$P(N_\lambda = n) = \frac{\lambda^n e^{-\lambda}}{n!},\ n = 0, 1, 2, ... \tag{4}$$

where $E(N_\lambda) = \lambda$ and $V(N_\lambda) = \lambda$. Then the probability generating function of N_λ is given by

$$g_{N_\lambda}(s) = E(s^{N_\lambda}) = \sum_{n=0}^{\infty} \frac{s^n e^{-\lambda} \lambda^n}{n!} = e^{-\lambda} \sum_{n=0}^{\infty} \frac{(s\lambda)^n}{n!} = \exp[(s-1)\lambda]. \tag{5}$$

See Johnson et al. [8].

Definition 4. Let N_λ^w be a weighted Poisson distributed (WPD) random variable with parameter $\lambda > 0$, weight function $w(n) > 0$, $n = 0, 1, 2, 3...$, and probability mass function given by

$$P(N_\lambda^w = n) = \frac{w(n)}{E(w(N_\lambda))} \frac{\lambda^n e^{-\lambda}}{n!},\ n = 0, 1, 2, ... \tag{6}$$

where $N_\lambda \sim POI(\lambda)$ and $E(w(N_\lambda)) = \sum_{n=0}^{\infty} w(n) \frac{\lambda^n e^{-\lambda}}{n!} < \infty$ is the normalising constant. Then the expected value and variance of N_λ^w are $E(N_\lambda^w) = \lambda \frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}$ and $V(N_\lambda^w) = E(N_\lambda^w) + \lambda^2 \left[\frac{E(w(N_\lambda+2))}{E(w(N_\lambda))} - \left(\frac{E(w(N_\lambda+1))}{E(w(N_\lambda))} \right)^2 \right]$ and the probability generating function is given by

$$\psi_{N_\lambda^w(s)} = E(s^{N_\lambda^w}) = \frac{E(w(N_{\lambda s}))}{E(w(N_\lambda))} e^{-\lambda(1-s)} \tag{7}$$

where $N_{\lambda s} \sim POI(\lambda s)$; see Minkova and Balakrishnan [9].

Definition 5. let N_λ be a Poisson random variable with parameter $\lambda > 0$ and let Y_i, $i = 1, 2, 3, ...$, be i.i.d. random variables, independent of N_λ where $E(Y_i) = \eta$ and $V(Y_i) = \sigma^2$. Then

$$X = \sum_{i=1}^{N_\lambda} Y_i \tag{8}$$

is said to have a compound Poisson distribution with $E(X) = \lambda\eta$ and $V(X) = \lambda(\eta^2 + \sigma^2)$.

Definition 6. Let X be any random variable, the Fisher index of dispersion for X is defined as $FI(X) = \frac{var(X)}{E(X)}$; see Fisher [7].

Definition 7. The survival function of a nonnegetive discrete random variable X is defined as the probability $S(x) = 1 - P(X \leq x) = P(X > x)$.

Theorem 8. *Let X and N be random variables. The conditional expected value and conditional variance are given by*

$$E(X) = E_N\left[E(X|N)\right] \text{ and } Var(X) = E\left[Var(X|N)\right] + Var\left[E(X|N)\right]. \tag{9}$$

Theorem 9. *The conditional probability of the event A given the event B is given by $P(A/B) = \frac{P(A\cap B)}{P(B)}$. If A and B are independent then*

$$P(A/B) = \frac{P(A)P(B)}{P(B)} = P(A). \tag{10}$$

3 The geometric Poisson distribution

In this section we derive the results of the geometric Poisson distribution and its Fisher index. The GPD we will look at is defined as $X = \sum_{i=1}^{N_\lambda} Y_i$ with paramters $\lambda > 0$ and $0 < \theta < 1$ where $N_\lambda \sim POI(\lambda)$ and Y_i, $i = 1, 2, 3, ...$ are i.i.d. geometric random variables independent of N_λ.

Theorem 10. *The probability mass function of X is given by*

$$P_X(X = k) = \sum_{n=1}^{k} e^{-\lambda}\frac{\lambda^n}{n!}\begin{pmatrix} k-1 \\ n-1 \end{pmatrix}\theta^n(1-\theta)^{k-n},\ k = 1, 2, 3, ...\ n = 1, 2, 3, ... \tag{11}$$

It is said that X has a geometric Poisson distribution and is denoted by $X \sim GPD(\lambda, \theta)$.

Proof. By the definition of a probability mass function and using (3), (4) and (10), the pmf of X is

$p_X(X = k)$

$= \sum_{n=0}^{\infty} P(\sum_{i=1}^{N_\lambda} Y_i = k,\ N_\lambda = n)$

$= \sum_{n=0}^{\infty} P(Y_1 + Y_2 + ... + Y_{N^\lambda} = k|N_\lambda = n)P(N_\lambda = n)$

$= \sum_{n=0}^{\infty} P(Y_1 + Y_2 + ... + Y_n = k)e^{-\lambda}\frac{\lambda^n}{n!}$

$= \sum_{n=1}^{k} e^{-\lambda}\frac{\lambda^n}{n!}\begin{pmatrix} k-1 \\ n-1 \end{pmatrix}\theta^n(1-\theta)^{k-n},\ k = 1, 2, 3, ...$

□

Theorem 11. *If* $X = \sum_{i=1}^{N_\lambda} Y_i \sim GPD(\lambda, \theta)$, *then the pgf of* X *is given by*

$$g_X(s) = \exp[(g_Y(s) - 1)\lambda] \tag{12}$$

where $g_Y(s) = \frac{\theta s}{1-(1-\theta)s}$ *from* (2) .

Proof. Let $N_\lambda \sim POI(\lambda)$ with pgf $g_{N_\lambda}(s)$ given by (5) and $Y_i \sim GEO(\theta)$, $i = 1, 2, 3, ...$ with pgf $g_Y(s)$ given by (2). Using (2) , (5) *and* (9) and the definition of a pgf, the pgf of X is given by

$g_X(s)$

$= E[s^X]$

$= E[E[s^X|N_\lambda]]$

$= E[E[s^{Y_1+Y_2+Y_3+...+Y_{N_\lambda}}|N_\lambda]]$

$= E[E[s^{Y_1}s^{Y_2}\cdots s^{Y_{N_\lambda}}|N_\lambda]]$

$= E[g_Y(s)]^{N_\lambda}$

$= g_{N_\lambda}(g_Y(s))$

$= \exp[(g_Y(s) - 1)\lambda]$

$= \exp\left[(\frac{s-1}{1-(1-\theta)s})\lambda\right]$.

□

Theorem 12. *Let* $X = \sum_{i=1}^{N_\lambda} Y_i \sim GPD(\lambda, \theta)$. *Then*

$$E(X) = \frac{\lambda}{\theta} \text{ and } Var(X) = \frac{\lambda(2-\theta)}{\theta^2}. \tag{13}$$

Proof. From (12) the first and second derivatives of the pgf of a geometric Poisson random variable are given by

$g_X^{'}(s) = g_Y^{'}(s)\lambda \exp[\lambda(g_Y(s) - 1)]$

where $g_Y^{'}(s) = \theta[1 - (1 - \theta)s]^{-2}$

and
$g_X^{''}(s) = g_Y^{''}(s)\lambda \exp[\lambda(g_Y(s) - 1)] + g_Y^{'}(s)\lambda(g_Y^{'}(s)\lambda \exp[\lambda(g_Y(s) - 1)])$

where $g_Y^{''}(s) = 2\theta(1 - \theta)[1 - (1 - \theta)s]^{-1}$.

Letting $s = 1$ we have

$g_Y(1) = 1$, $g_Y^{'}(1) = \frac{1}{\theta}$ and $g_Y^{''}(1) = \frac{2(1-\theta)}{\theta^2}$.

Then
$g_X^{'}(1) = g_Y^{'}(1)\lambda \exp[\lambda(g_Y(1) - 1)] = \frac{\lambda}{\theta}$

and
$g_X^{''}(1) = \frac{2\lambda(1-\theta)+\lambda^2}{\theta^2}$.

Since $g_X^{'}(1) = E(X)$ and $g_X^{''}(1) = E[X(X-1)] = E(X^2) - E(X)$

It follows that
$E(X) = \frac{\lambda}{\theta}$

and
$Var(X) = E(X^2) - (E(X))^2$

$= g_X^{''}(1) + g_X^{'}(1) - [g_X^{'}(1)]^2$

$= \frac{\lambda(2-\theta)}{\theta^2}$.

□

Theorem 13. *If $X \sim GPD(\lambda, \theta)$ then*

$$FI(X) = \frac{2-\theta}{\theta}. \tag{14}$$

Proof. This follows directly from the results of the pervious theorem. Using (13)

$FI(X)$

$= \frac{Var(X)}{E(X)}$

$= \frac{\frac{\lambda(2-\theta)}{\theta^2}}{\frac{\lambda}{\theta}}$

$= \frac{2-\theta}{\theta}$. □

Since $0 < \theta < 1$, for any value of θ used $FI(X) = \frac{2-\theta}{\theta} > 1$ hence the geometric Poisson distribution is always over dispersed.

4 The geometric weighted Poisson distribution

In this section we derive the results of the geometric weighted Poisson distribution for the general case and the two special cases where the weight functions are $w(n) = n$ and $w(n) = \frac{1}{n+1}$. We also derive and explain the Fisher indices.

4.1 The general case of the geometric weighted Poisson distribution

In this section we will look at the random variable $M^w = \sum_{i=1}^{N_\lambda^w} Y_i$ where $N_\lambda^w \sim WPD(\lambda)$ and Y_i, $i = 1, 2, 3, ...$ are i.i.d. $GEO(\theta)$ variables independent of N_λ^w. $E(w(N_\lambda)) = \sum_{n=0}^{\infty} w(n)\frac{\lambda^n e^{-\lambda}}{n!}$ is the normalizing constant for the weighted Poisson distribution with $N_\lambda \sim POI(\lambda)$.

Theorem 14. *The probability mass function of M^w is given by*

$$P(M^w = k) = \sum_{n=1}^{k} \frac{w(n)}{E(w(N_\lambda))} \frac{\lambda^n e^{-\lambda}}{n!} \begin{pmatrix} k-1 \\ n-1 \end{pmatrix} \theta^n (1-\theta)^{k-n} \quad k = 1, 2, 3..., \;\; n = 1, 2, 3, ..., \tag{15}$$

It is said that M^w has a geometric weighted Poisson distribution and is denoted as $M^w \sim GWPD(\lambda, \theta)$.

Proof. By the definition of a probability mass function and using (3), (6) and (10), the pmf of M^w is

$P(M^w = k)$

$= \sum_{n=0}^{\infty} P(\sum_{i=1}^{N_\lambda^w} Y_i = k,\, N_\lambda^w = n)$

$= \sum_{n=0}^{\infty} P(Y_1 + Y_2 + ... + Y_n = k | N_\lambda^w = n) P(N_\lambda^w = n)$

$= \sum_{n=0}^{\infty} P(Y_1 + Y_2 + ... + Y_n = k) \frac{w(n)}{E(w(N_\lambda))} \frac{\lambda^n e^{-\lambda}}{n!}$

$= \sum_{n=1}^{k} \frac{w(n)\lambda^n e^{-\lambda}}{n! E(w(N_\lambda))} \begin{pmatrix} k-1 \\ n-1 \end{pmatrix} \theta^n (1-\theta)^{k-n}, \quad k = 1, 2, 3, ...$

□

Theorem 15. *If $M^w = \sum_{i=1}^{N_\lambda^w} Y_i \sim GWPD(\lambda, \theta)$ then the pgf of M^w is given by*

$$g_{M^w}(s) = \frac{1}{E(w(N_\lambda))} \sum_{n=0}^{\infty} w(n) \frac{(\lambda g_Y(s))^n e^{-\lambda}}{n!} = \frac{E(w(N_{\lambda g_Y(s)}))}{E(w(N_\lambda))} \exp\left[(\frac{s-1}{1-(1-\theta)s})\lambda \right] \tag{16}$$

where $N_{\lambda g_Y(s)} \sim POI(\lambda g_Y(s))$.

Proof. Let $N_\lambda^w \sim WPD(\lambda)$ independent of $Y_i \sim GEO(\theta)$, $i = 1, 2, 3...$with pgf $g_Y(s)$ given by (2). Using (2),(6) *and* (9) and the definition of a pgf, the pgf of M^w is given by

$g_{M^w}(s)$

$= E[s^{M^w}]$

$= E[E[s^{M^w} | N_\lambda^w = n]]$

$= E[E[s^{Y_1+Y_2+Y_3+...+Y_{N_\lambda^w}} | N_\lambda^w = n]]$

$= \sum_{n=0}^{\infty} E[s^{Y_1+Y_2+Y_3+...+Y_{N_\lambda^w}} | N_\lambda^w = n] P(N_\lambda^w = n)$

$= \sum_{n=0}^{\infty} E[s^{Y_1} s^{Y_2} \cdots s^{Y_{N_\lambda^w}} | N_\lambda^w = n] P(N_\lambda^w = n)$

$= \sum_{n=0}^{\infty}[g_Y(s)]^n \frac{w(n)}{E(w(N_\lambda))} \frac{\lambda^n e^{-\lambda}}{n!}$

$= \frac{1}{E(w(N_\lambda))} \sum_{n=0}^{\infty} w(n) \frac{(\lambda g_Y(s))^n e^{-\lambda}}{n!}$

$= \frac{1}{E(w(N_\lambda))} \frac{e^{-\lambda}}{e^{-\lambda g_y(s)}} \sum_{n=0}^{\infty} w(n) \frac{(\lambda g_Y(s))^n e^{-\lambda g_y(s)}}{n!}$

$= \frac{E(w(N_{\lambda g_Y(s)}))}{E(w(N_\lambda))} \exp\left[-\lambda(1-g_Y(s))\right]$

$= \frac{E(w(N_{\lambda g_Y(s)}))}{E(w(N_\lambda))} \exp\left[(\frac{s-1}{1-(1-\theta)s})\lambda\right]$ *where* $N_{\lambda g_Y(s)} \sim POI(\lambda g_Y(s))$.

□

Theorem 16. *Let* $M^w = \sum_{i=1}^{N_\lambda^w} Y_i \sim GWPD(\lambda, \theta)$. *Then*

$$E(M^w) = \frac{\lambda}{\theta} \frac{E(w(N_\lambda + 1))}{E(w(N_\lambda))} \tag{17}$$

and

$$Var(M^w) = \frac{(2-\theta)\lambda}{\theta^2} \frac{E(w(N_\lambda+1))}{E(w(N_\lambda))} + \frac{\lambda^2}{\theta^2}\left[\frac{E(w(N_\lambda+2))}{E(w(N_\lambda))} - \left(\frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}\right)^2\right]. \tag{18}$$

Proof. The first and second derivatives of the pgf of a GWPD given in (16) are given by

$g'_{M^w}(s)$

$= \frac{\lambda e^{-\lambda}}{E(w(N_\lambda))} \sum_{n=1}^{\infty} w(n) \frac{[\lambda g_Y(s)]^{n-1}}{(n-1)!} g'_Y(s)$

$g''_{M^w}(s)$

$= \frac{\lambda e^{-\lambda}}{E(w(N_\lambda))} \sum_{n=1}^{\infty} \frac{w(n)\lambda^{(n-1)}}{(n-1)!} \left[(n-1)(g_Y(s))^{n-2}[g'_Y(s)]^2 + g''_Y(s)(g_Y(s))^{n-1}\right]$

$= \frac{\lambda^2 e^{-\lambda}}{E(w(N_\lambda))} \sum_{n=2}^{\infty} w(n) \frac{(\lambda g_Y(s))^{n-2}}{(n-2)!}[g'_Y(s)]^2 + \frac{\lambda e^{-\lambda}}{E(w(N_\lambda))} \sum_{n=1}^{\infty} w(n) \frac{(\lambda g_Y(s))^{n-1}}{(n-1)!} g''_Y(s)$

$= \frac{\lambda^2 e^{-\lambda}}{E(w(N_\lambda))} \sum_{n=0}^{\infty} w(n+2) \frac{[\lambda g_Y(s)]^n}{n!}[g'_Y(s)]^2 + \frac{\lambda e^{-\lambda}}{E(w(N_\lambda))} \sum_{n=0}^{\infty} w(n+1) \frac{[\lambda g_Y(s)]^n}{n!} g''_Y(s).$

By the definition of a pgf $E(M^w) = g'_{M^w}(1)$ and $E\left[M^w(M^w-1)\right] = g''_{M^w}(1)$.

The first two moments of M^w are then given by $E(M^w) = g'_{M^w}(1)$ and $E[M^w]^2 = g''_{M^w}(1) + g'_{M^w}(1)$.

From (2) for $Y \sim GEO(\theta)$ $g_Y(1) = 1$, $g'_Y(1) = \frac{1}{\theta}$ and $g''_Y(1) = \frac{2(1-\theta)}{\theta^2}$.

From this and the expressions for $g'_{M^w}(s)$ and $g''_{M^w}(s)$ it follows that

$E[M^w]$

$= \frac{\lambda e^{-\lambda}}{E(w(N_\lambda))} \sum_{n=1}^{\infty} w(n) \frac{[\lambda g_Y(1)]^{n-1}}{(n-1)!} g'_Y(1)$

$= \frac{\lambda e^{-\lambda}}{E(w(N_\lambda))} \sum_{n=1}^{\infty} w(n) \frac{\lambda^{n-1}}{(n-1)!} \frac{1}{\theta}$

$= \frac{\lambda}{\theta E(w(N_\lambda))} \sum_{n=0}^{\infty} w(n+1) \frac{\lambda^n e^{-\lambda}}{n!}$

$= \frac{\lambda}{\theta} \frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}.$

Also,
$E[M^w]^2$

$= g''_{M^w}(1) + g'_{M^w}(1)$

$= \frac{\lambda^2 e^{-\lambda}}{E(w(N_\lambda))} \sum_{n=0}^{\infty} w(n+2) \frac{[\lambda g_Y(1)]^n}{n!} [g'_Y(1)]^2 + \frac{\lambda e^{-\lambda}}{E(w(N_\lambda))} \sum_{n=0}^{\infty} w(n+1) \frac{[\lambda g_Y(1)]^n}{n!} g''_Y(1) + \frac{\lambda}{\theta} \frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}$

$= \frac{\lambda^2 e^{-\lambda}}{E(w(N_\lambda))} \sum_{n=0}^{\infty} w(n+2) \frac{\lambda^n}{n!} \frac{1}{\theta^2} + \frac{\lambda e^{-\lambda}}{E(w(N_\lambda))} \sum_{n=0}^{\infty} w(n+1) \frac{\lambda^n}{n!} \frac{2(1-\theta)}{\theta^2} + \frac{\lambda}{\theta} \frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}$

$= \frac{\lambda^2}{E(w(N_\lambda))} \sum_{n=0}^{\infty} w(n+2) \frac{\lambda^n e^{-\lambda}}{n!} \frac{1}{\theta^2} + \frac{\lambda}{E(w(N_\lambda))} \sum_{n=0}^{\infty} w(n+1) \frac{\lambda^n e^{-\lambda}}{n!} \frac{2(1-\theta)}{\theta^2} + \frac{\lambda}{\theta} \frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}$

$= \frac{\lambda^2}{E(w(N_\lambda))} E(w(N_\lambda+2)) \frac{1}{\theta^2} + \frac{\lambda}{E(w(N_\lambda))} E(w(N_\lambda+1)) \frac{2(1-\theta)}{\theta^2} + \frac{\lambda}{\theta} \frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}$

$Hence \quad var(M^w) = E((M^w)^2) - (E(M^w))^2$

$= \frac{\lambda^2 E(w(N_\lambda+2))}{\theta^2 E(w(N_\lambda))} + \frac{\lambda E(w(N_\lambda+1))}{E(w(N_\lambda))} \frac{(2-2\theta)}{\theta^2} + \frac{\lambda}{\theta} \frac{E(w(N_\lambda+1))}{E(w(N_\lambda))} - \left(\frac{\lambda}{\theta} \frac{E(w(N_\lambda+1))}{E(w(N_\lambda))} \right)^2$

$= \frac{(2-\theta)\lambda}{\theta^2} \frac{E(w(N_\lambda+1))}{E(w(N_\lambda))} + \frac{\lambda^2}{\theta^2} \left[\frac{E(w(N_\lambda+2))}{E(w(N_\lambda))} - \left(\frac{E(w(N_\lambda+1))}{E(w(N_\lambda))} \right)^2 \right].$

□

Theorem 17. *If* $M^w = \sum_{i=1}^{N_\lambda^w} Y_i \sim GWPD(\lambda, \theta)$ *then*

$$FI(M^w) = \frac{(2-\theta)}{\theta} + \frac{\lambda}{\theta} \left[\frac{E(w(N_\lambda+2))}{E(w(N_\lambda+1))} - \frac{1}{\lambda} E\left(N_\lambda^w\right) \right]. \tag{19}$$

Proof. This follows directly from the results of Theorem 16. That is, using (17) *and* (18)

$$FI(M^w) = \frac{var(M^w)}{E(M^w)} = \frac{\frac{(2-\theta)\lambda}{\theta^2}\frac{E(w(N_\lambda+1))}{E(w(N_\lambda))} + \frac{\lambda^2}{\theta^2}\left[\frac{E(w(N_\lambda+2))}{E(w(N_\lambda))} - \left(\frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}\right)^2\right]}{\frac{\lambda}{\theta}\frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}}$$

$$= \frac{(2-\theta)}{\theta} + \frac{\lambda}{\theta}\left[\frac{E(w(N_\lambda+2))}{E(w(N_\lambda+1))} - \left(\frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}\right)\right].$$

From Definition 4

$$E\left(N_\lambda^w\right) = \lambda\frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}$$

and it follows that

$$FI(M^w) = \frac{(2-\theta)}{\theta} + \frac{\lambda}{\theta}\left[\frac{E(w(N_\lambda+2))}{E(w(N_\lambda+1))} - \frac{1}{\lambda}E\left(N_\lambda^w\right)\right].$$

□

From the above expression for the Fisher index we see that the dispersion of a GWPD is dependent upon the weight function of the underlying weighted Poisson random variable.

4.2 A geometric weighted Poisson distribution with weight function $w(n) = n$

In Section 4.1 we derived the results of a GWPD with the weight function unspecified. In this section we apply the results obtained to the specific case where $w(n) = n$. We consider $M^w = \sum_{i=1}^{N_\lambda^w} Y_i \sim GWPD(\lambda, \theta)$ with parameters $\lambda > 0$ *and* $0 < \theta < 1$, $N_\lambda^w \sim WPD(\lambda)$ with Y_i, $i = 1, 2, 3, ...$ i.i.d. geometric random variables independent of N_λ^w. With weight function $w(n) = n$ and $E(w(N_\lambda)) = E(N_\lambda) = \lambda$.

Theorem 18. *Let $M^w = \sum_{i=1}^{N_\lambda^w} Y_i \sim GWPD(\lambda, \theta)$ with weight function $w(n) = n$. Then the pmf of M^w is given by*

$$P(M^w = k) = \sum_{n=1}^{k} \frac{\lambda^{n-1}e^{-\lambda}}{(n-1)!}\begin{pmatrix} k-1 \\ n-1 \end{pmatrix}\theta^n(1-\theta)^{k-n} \quad k = 1, 2, 3..., \;\; n = 1, 2, ..., \tag{20}$$

Proof. Using (15) and the fact that $E(w(N_\lambda)) = \lambda$ it follows

$$P(M^w = k)$$
$$= \sum_{n=1}^{k} \frac{n}{\lambda}\frac{\lambda^n e^{-\lambda}}{n!}\begin{pmatrix} k-1 \\ n-1 \end{pmatrix}\theta^n(1-\theta)^{k-n}$$
$$= \sum_{n=1}^{k} \frac{\lambda^{n-1}e^{-\lambda}}{(n-1)!}\begin{pmatrix} k-1 \\ n-1 \end{pmatrix}\theta^n(1-\theta)^{k-n}.$$

□

Theorem 19. *If* $M^w = \sum_{i=1}^{N_\lambda^w} Y_i \sim GWPD(\lambda, \theta)$ *with weight function* $w(n) = n$ *then the pgf of* M^w *is given by*

$$g_{M^w}(s) = g_Y(s) \exp\left[(\frac{s-1}{1-(1-\theta)s})\lambda\right] \tag{21}$$

where $g_Y(s)$ *is given by* (2).

Proof. From (16) and the fact that $N_{\lambda g_Y(s)} \sim POI(\lambda g_Y(s))$ it follows

$g_{M^w}(s)$

$= \frac{E(w(N_{\lambda g_Y(s)}))}{E(w(N_\lambda))} \exp\left[(\frac{s-1}{1-(1-\theta)s})\lambda\right]$

$= \frac{E(N_{\lambda g_Y(s)})}{E(N_\lambda)} \exp\left[(\frac{s-1}{1-(1-\theta)s})\lambda\right]$

$= \frac{\lambda g_Y(s)}{\lambda} \exp\left[(\frac{s-1}{1-(1-\theta)s})\lambda\right]$

$= g_Y(s) \exp\left[(\frac{s-1}{1-(1-\theta)s})\lambda\right]$.

□

Theorem 20. *Let* $M^w = \sum_{i=1}^{N_\lambda^w} Y_i \sim GWPD(\lambda, \theta)$ *with weight function* $w(n) = n$. *Then*

$$E(M^w) = \frac{\lambda + 1}{\theta} \tag{22}$$

and

$$Var(M^w) = \frac{(1-\theta) + (2-\theta)\lambda}{\theta^2}. \tag{23}$$

Proof. From (17) and (18)

$E(M^w)$

$= \frac{\lambda}{\theta} \frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}$

$= \frac{\lambda}{\theta} \frac{E(N_\lambda+1)}{E(N_\lambda)}$

$= \frac{\lambda+1}{\theta}$.

$Var(M^w)$

$= \frac{(2-\theta)\lambda}{\theta^2} \frac{E(w(N_\lambda+1))}{E(w(N_\lambda))} + \frac{\lambda^2}{\theta^2}\left[\frac{E(w(N_\lambda+2))}{E(w(N_\lambda))} - \left(\frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}\right)^2\right]$

$= \frac{(2-\theta)\lambda}{\theta^2} \frac{\lambda+1}{\lambda} + \frac{\lambda^2}{\theta^2}\left[\frac{\lambda+2}{\lambda} - \left(\frac{\lambda+1}{\lambda}\right)^2\right]$

$= \frac{(1-\theta)+(2-\theta)\lambda}{\theta^2}$.

□

Theorem 21. *The Fisher index for a GWPD with weight function $w(n) = n$ is given by*

$$FI(M^w) = \frac{2-\theta}{\theta} - \frac{1}{\theta(1+\lambda)} < \frac{2-\theta}{\theta}. \tag{24}$$

Proof. Using (22) and (23), the Fisher index of dispersion is

$$FI(M^w) = \frac{var(M^w)}{E(M^w)}$$

$$= \frac{\frac{(1-\theta)+(2-\theta)\lambda}{\theta^2}}{\frac{\lambda+1}{\theta}} = \frac{(1-\theta)+(2-\theta)\lambda}{\theta(\lambda+1)}$$

$$= \frac{2-\theta}{\theta}\frac{\lambda}{\lambda+1} + \frac{1-\theta}{\theta(1+\lambda)} + \frac{2-\theta}{\theta(1+\lambda)} - \frac{2-\theta}{\theta(1+\lambda)}$$

$$= \frac{2-\theta}{\theta} - \frac{1}{\theta(1+\lambda)} < \frac{2-\theta}{\theta}.$$

□

From the above expression for the Fisher index we see that the dispersion of a GWPD with weight function $w(n) = n$ is always lower than that of the GPD for all values of $\lambda > 0$ and $0 < \theta < 1$.

4.3 A geometric weighted Poisson distribution with weight function $w(n) = \frac{1}{n+1}$

In this section we look at the other special case of the GWPD with weight function $w(n) = \frac{1}{n+1}$. We consider $M^w = \sum_{i=1}^{N_\lambda^w} Y_i \sim GWPD(\lambda, \theta)$ with parameters $\lambda > 0$ $0 < \theta < 1$, $N_\lambda^w \sim WPD(\lambda)$ and Y_i, $i = 1, 2, 3, ...$ i.i.d. geometric random variables independent of N_λ^w.

Theorem 22. *Let $M^w = \sum_{i=1}^{N_\lambda^w} Y_i \sim GWPD(\lambda, \theta)$ with weight function $w(n) = \frac{1}{n+1}$. Then the pmf is given by*

$$P(M^w = k) = \sum_{n=1}^{k} \frac{e^{-\lambda}}{(1-e^{-\lambda})} \frac{\lambda^{n+1}}{(n+1)!} \begin{pmatrix} k-1 \\ n-1 \end{pmatrix} \theta^n (1-\theta)^{k-n} \quad k = 1, 2, 3..., \;\; n = 1, 2, , ..., \tag{25}$$

Proof. Using (15),

$P(M^w = k)$

$= \sum_{n=1}^{k} \frac{1}{E\left(\frac{1}{n+1}\right)(n+1)} \frac{\lambda^n e^{-\lambda}}{n!} \begin{pmatrix} k-1 \\ n-1 \end{pmatrix} \theta^n (1-\theta)^{k-n}$

From the exponential function power series $\left(\sum_{k=0}^{\infty} \frac{z^k}{k!} = e^z\right)$ it follows that

$E\left(\frac{1}{N_\lambda+1}\right)$

$= \sum_{n=0}^{\infty} \frac{1}{n+1} \frac{\lambda^n e^{-\lambda}}{n!}$

$= \frac{e^{-\lambda}}{\lambda} \sum_{n=0}^{\infty} \frac{\lambda^{n+1}}{(n+1)!}$

$= \frac{e^{-\lambda}}{\lambda} \left(\sum_{n=0}^{\infty} \frac{\lambda^n}{n!} - 1\right)$

$= \frac{e^{-\lambda}}{\lambda} \left(e^{\lambda} - 1\right)$

$= \frac{1}{\lambda}(1 - e^{-\lambda})$.

Hence

$P(M^w = k)$

$= \sum_{n=1}^{k} \frac{1}{\frac{1}{\lambda}(1-e^{-\lambda})} \frac{\lambda^n e^{-\lambda}}{(n+1)!} \begin{pmatrix} k-1 \\ n-1 \end{pmatrix} \theta^n (1-\theta)^{k-n}$

$= \sum_{n=1}^{k} \frac{e^{-\lambda}}{(1-e^{-\lambda})} \frac{\lambda^{n+1}}{(n+1)!} \begin{pmatrix} k-1 \\ n-1 \end{pmatrix} \theta^n (1-\theta)^{k-n}$.

□

Theorem 23. *If $M^w = \sum_{i=1}^{N_\lambda^w} Y_i \sim GWPD(\lambda, \theta)$ with weight function $w(n) = \frac{1}{n+1}$. Then the pgf of M^w is given by*

$$g_{M^w}(s) = \frac{e^{-\lambda}(\exp(g_Y(s)\lambda) - 1)}{[1 - e^{-\lambda}]g_Y(s)} \qquad 0 < \theta < 1,\ \lambda > 0 \tag{26}$$

where $g_Y(s)$ is the pgf of a $GEO(\theta)$ random variable.

Proof. From (16) and the fact that $E(w(N_\lambda)) = \frac{1}{\lambda}(1 - e^{-\lambda})$ for $w(n) = \frac{1}{n+1}$ and $N_\lambda \sim \mathrm{POI}(\lambda)$ it follows that

$g_{M^w}(s)$

$= \frac{E(w(N_{\lambda g_Y(s)}))}{E(w(N_\lambda))} \exp\left[(\frac{s-1}{1-(1-\theta)s})\lambda\right]$

$= \frac{\sum_{n=0}^{\infty} \frac{1}{(n+1)n!}(\lambda g_Y(s))^n e^{-\lambda g_Y(s)}}{\frac{1}{\lambda}(1-e^{-\lambda})} \exp\left[(\frac{s-1}{1-(1-\theta)s})\lambda\right]$

$= \frac{e^{-\lambda g_Y(s)}}{(1-e^{-\lambda})g_Y(s)} \sum_{n=0}^{\infty} \frac{(\lambda g_Y(s))^{n+1}}{(n+1)!} \exp\left[(\frac{s-1}{1-(1-\theta)s})\lambda\right]$

$= \frac{e^{-\lambda g_Y(s)}}{(1-e^{-\lambda})g_Y(s)} (e^{\lambda g_Y(s)} - 1) \exp\left[(g_Y(s) - 1)\lambda\right]$

$= \frac{e^{-\lambda}(e^{g_Y(s)\lambda} - 1)}{(1-e^{-\lambda})g_Y(s)}$.

□

Theorem 24. *Let* $M^w = \sum_{i=1}^{N_\lambda^w} Y_i \sim GWPD(\lambda, \theta)$ *with weight function* $w(n) = \frac{1}{n+1}$. *Then*

$$E(M^w) = \frac{1}{(1-e^{-\lambda})\theta}(\lambda - 1 + e^{-\lambda}) \tag{27}$$

and

$$Var(M^w) = \frac{(2-\theta)(1-e^{-\lambda})(\lambda - 1 + e^{-\lambda}) + (1-e^{-\lambda})^2 - \lambda^2 e^{-\lambda}}{\theta^2(1-e^{-\lambda})^2}. \tag{28}$$

Proof. From (17) and (18)

$E(M^w)$

$= \frac{\lambda}{\theta}\frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}$

$= \frac{\lambda}{\theta}\frac{\sum_{n=0}^{\infty}\frac{1}{n+2}\frac{\lambda^n e^{-\lambda}}{n!}}{\frac{1}{\lambda}(1-e^{-\lambda})}$

$= \frac{1}{(1-e^{-\lambda})\theta}\sum_{n=0}^{\infty}\frac{\lambda^{n+2}(n+1)e^{-\lambda}}{(n+2)!}$

setting $y = n+2$

$= \frac{1}{(1-e^{-\lambda})\theta}\sum_{y=2}^{\infty}\frac{\lambda^y e^{-\lambda}}{y!}(y-1)$

$= \frac{1}{(1-e^{-\lambda})\theta}\left(\sum_{y=0}^{\infty}\frac{\lambda^y e^{-\lambda}}{y!}(y-1) - \frac{\lambda^0 e^{-\lambda}}{0!}(-1) - \frac{\lambda e^{-\lambda}}{1}(0)\right)$

$= \frac{1}{(1-e^{-\lambda})\theta}(\lambda - 1 + e^{-\lambda})$

$Var(M^w)$

$= \frac{(2-\theta)\lambda}{\theta^2}\frac{E(w(N_\lambda+1))}{E(w(N_\lambda))} + \frac{\lambda^2}{\theta^2}\left[\frac{E(w(N_\lambda+2))}{E(w(N_\lambda))} - \left(\frac{E(w(N_\lambda+1))}{E(w(N_\lambda))}\right)^2\right]$

$= \frac{(2-\theta)}{\theta^2}\frac{(\lambda-1+e^{-\lambda})}{[1-e^{-\lambda}]} + \frac{\lambda^2}{\theta^2}\left[\frac{E(w(N_\lambda+2))}{\frac{1}{\lambda}[1-e^{-\lambda}]} - \left(\frac{(\lambda-1+e^{-\lambda})}{\lambda[1-e^{-\lambda}]}\right)^2\right]$

Since $E(w(N_\lambda+2))$

$= \sum_{n=0}^{\infty}\frac{1}{n+3}\frac{\lambda^n e^{-\lambda}}{n!}$

$= \frac{1}{\lambda^3}\sum_{n=0}^{\infty}\frac{\lambda^{n+3}e^{-\lambda}}{(n+3)!}(n+1)(n+2)$

Letting $z = n+3$ it follows

$E(w(N_\lambda+2))$

$= \frac{1}{\lambda^3}\sum_{z=3}^{\infty}\frac{\lambda^z e^{-\lambda}}{z!}(z-1)(z-2)$

$= \frac{1}{\lambda^3}\left(\sum_{z=0}^{\infty}\frac{\lambda^z e^{-\lambda}}{z!}(z-1)(z-2) - \frac{\lambda^0 e^{-\lambda}}{0!}(0-2)(0-1) - \frac{\lambda^1 e^{-\lambda}}{1!}(1-1)(1-2) - \frac{\lambda^2 e^{-\lambda}}{2!}(2-1)(2-2)\right)$

$= \frac{1}{\lambda^3}\left(\sum_{z=0}^{\infty}\frac{\lambda^z e^{-\lambda}}{z!}(z-1)(z-2) - 2e^{-\lambda}\right)$

$= \frac{1}{\lambda^3}\left(\lambda^2 - 2\lambda + 2 - 2e^{-\lambda}\right)$.

Therefore $Var(M^w)$

$= \frac{(2-\theta)(1-e^{-\lambda})(\lambda-1+e^{-\lambda})+(1-e^{-\lambda})^2-\lambda^2 e^{-\lambda}}{\theta^2(1-e^{-\lambda})^2}$.

□

Theorem 25. *The Fisher index for the geometric weighted Poisson distribution with weight function* $w(n) = \frac{1}{n+1}$ *is*

$$FI(M^w) = \frac{(2-\theta)}{\theta} + \frac{(1-e^{-\lambda})^2 - \lambda^2 e^{-\lambda}}{\theta(1-e^{-\lambda})(\lambda - 1 + e^{-\lambda})}. \tag{29}$$

Proof. Using (27) and (28) the Fisher index of dispersion is given by

$$FI(M^w)$$

$$= \frac{var(M^w)}{E(M^w)}$$

$$= \frac{\frac{(2-\theta)(1-e^{-\lambda})(\lambda-1+e^{-\lambda})+(1-e^{-\lambda})^2-\lambda^2 e^{-\lambda}}{\theta^2(1-e^{-\lambda})^2}}{\frac{1}{(1-e^{-\lambda})\theta}(\lambda - 1 + e^{-\lambda})}$$

$$= \frac{(2-\theta)(1-e^{-\lambda})(\lambda-1+e^{-\lambda}) + (1-e^{-\lambda})^2 - \lambda^2 e^{-\lambda}}{\theta(1-e^{-\lambda})(\lambda-1+e^{-\lambda})}$$

$$= \frac{(2-\theta)}{\theta} + \frac{(1-e^{-\lambda})^2 - \lambda^2 e^{-\lambda}}{\theta(1-e^{-\lambda})(\lambda-1+e^{-\lambda})}.$$

□

5 Summary of the results

Table 1 summaries the results for the three distributions that were looked at previously. For any value of $0 < \theta < 1$ the Fisher index of the GPD is always greater than 1 hence the GPD is always over-dispersed. The Fisher index for GWPD: $w(n) = n$ is always smaller than the Fisher index for the GPD and is therefore under-dispersed relative to the GPD. On the other hand, the GWPD: $w(n) = \frac{1}{n+1}$ is over-dispersed relative to the GPD since the term $\frac{(1-e^{-\lambda})^2-\lambda^2 e^{-\lambda}}{\theta(1-e^{-\lambda})(\lambda-1+e^{-\lambda})}$ is always positive.

Distribution	Pmf	Pgf , $g(s)$
GPD	$P_X(X=k) = \sum_{n=1}^{k} e^{-\lambda}\frac{\lambda^n}{n!}\begin{pmatrix} k-1 \\ n-1 \end{pmatrix}\theta^n(1-\theta)^{k-n}$	$\exp[(g_Y(s)-1)\lambda]$
GWPD: $w(n)=n$	$P(M^w=k) = \sum_{n=1}^{k} \frac{\lambda^{n-1}e^{-\lambda}}{(n-1)!}\begin{pmatrix} k-1 \\ n-1 \end{pmatrix}\theta^n(1-\theta)^{k-n}$	$g_Y(s)\exp\left[(\frac{s-1}{1-(1-\theta)s})\lambda\right]$
GWPD: $w(n)=\frac{1}{n+1}$	$P(M^w=k) = \sum_{n=1}^{k} \frac{e^{-\lambda}}{(1-e^{-\lambda})}\frac{\lambda^{n+1}}{(n+1)!}\begin{pmatrix} k-1 \\ n-1 \end{pmatrix}\theta^n(1-\theta)^{k-n}$	$\frac{e^{-\lambda}(\exp(g_Y(s)\lambda)-1)}{[1-e^{-\lambda}]g_Y(s)}$

(a) The probability mass functions and probability generating functions of the distributions.

Distribution	Expected value	Variance	Fisher index
GPD	$\frac{\lambda}{\theta}$	$\frac{\lambda(2-\theta)}{\theta^2}$	$\frac{2-\theta}{\theta}$
GWPD: $w(n)=n$	$\frac{\lambda+1}{\theta}$	$\frac{(1-\theta)+(2-\theta)\lambda}{\theta^2}$	$\frac{2-\theta}{\theta} - \frac{1}{\theta(1+\lambda)}$
GWPD: $w(n)=\frac{1}{n+1}$	$\frac{(\lambda-1+e^{-\lambda})}{(1-e^{-\lambda})\theta}$	$\frac{(2-\theta)(1-e^{-\lambda})(\lambda-1+e^{-\lambda})+(1-e^{-\lambda})^2-\lambda^2e^{-\lambda}}{\theta^2(1-e^{-\lambda})^2}$	$\frac{(2-\theta)}{\theta} + \frac{(1-e^{-\lambda})^2-\lambda^2e^{-\lambda}}{\theta(1-e^{-\lambda})(\lambda-1+e^{-\lambda})}$

(b) The moments and Fisher index of the distributions.

Table 1: Summary of the results.

6 Graphical comparison of the distributions

In this subsection we study the pmfs and the Fisher indices of the three distributions for different parameter values, first by keeping θ constant and varying λ and then by keeping λ constant and varying θ. For illustration purposes, we will use continuous lines for the pmf graphs even though the distributions are discrete. The data analysis for this subsection was performed using SAS software [1].

6.1 The probability mass functions

For θ fixed and λ varying.

Table 2 gives the summary of statistics calculated for the three distributions when $\theta = 0.5$ and λ increases from 3 to 6 to 20. Figures 1(a), 1(c) and 1(e) give the graphical displays of these distributions for these values. From Table 2 it can be seen that as λ increases the mean and variance of each of the distributions increases. For given values of θ and λ the central location of the GWPD:$w(n) = n$ is always greater than that of the GPD whilst the central location of the GWPD:$w(n) = \frac{1}{n+1}$ is always smaller than that of the GPD. The same order is also observed for the variances with the GWPD:$w(n) = n$ having the largest variance and the GWPD:$w(n) = \frac{1}{n+1}$ having the smallest variance of the three distributions for given θ and λ.

	λ	Expected value	Variance	Fisher index
GPD	3	6	18	3
	6	12	36	3
	20	40	120	3
GWPD: $w(n) = n$	3	8	20	2.5
	6	14	38	2.714
	20	42	122	2.905
GWPD: $w(n) = \frac{1}{n+1}$	3	4.314	14.958	3.467
	6	10.030	33.731	3.363
	20	38	118	3.105

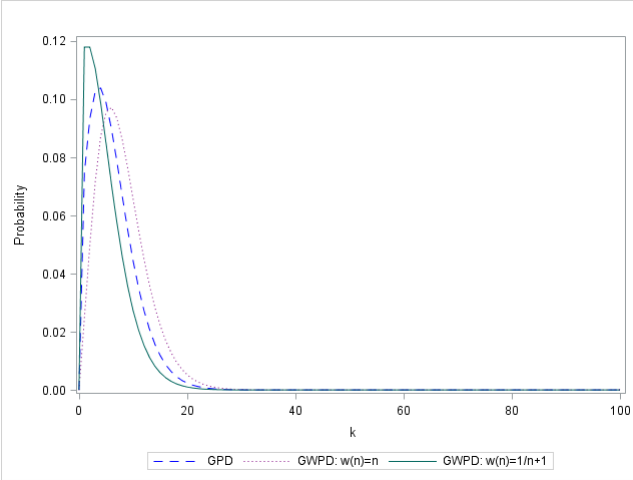
Table 2: Numerical summary of statistics for $\theta = 0.5$ and λ varying

For λ fixed and θ varying.

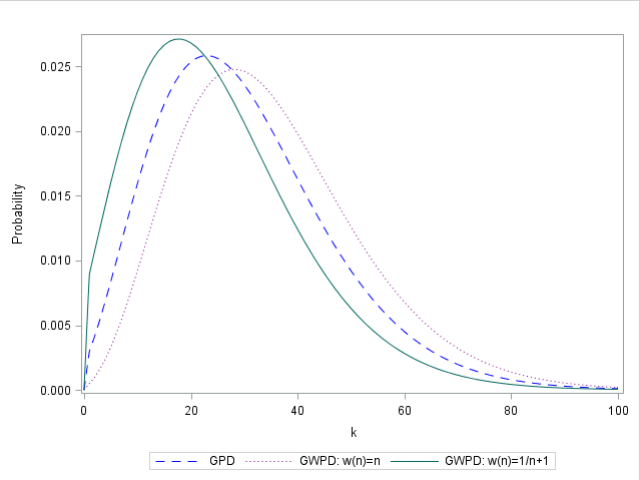
Table 3 shows as θ increases the variance and mean of each of the distributions decreases. This is also illustrated in Figures 1(b), 1(d) and 1(f) where its seen that as θ increases the distributions become less spread out. It is also observed that the central location and variance of the GWPD:$w(n) = n$ is always greater than that of the GPD whilst the central location and variance of the GWPD:$w(n) = \frac{1}{n+1}$ is always smaller than that of the GPD.

	θ	Expected value	Variance	Fisher index
GPD	0.2	30	270	9
	0.5	12	36	3
	0.7	8.571	15.918	1.857
$GWPD : w(n) = n$	0.2	35	290	8.286
	0.5	14	38	2.715
	0.7	10	16.531	1.653
$GWPD : w(n) = \frac{1}{n+1}$	0.2	25.075	248.429	9.908
	0.5	10.030	33.731	3.363
	0.7	7.164	15.163	2.116

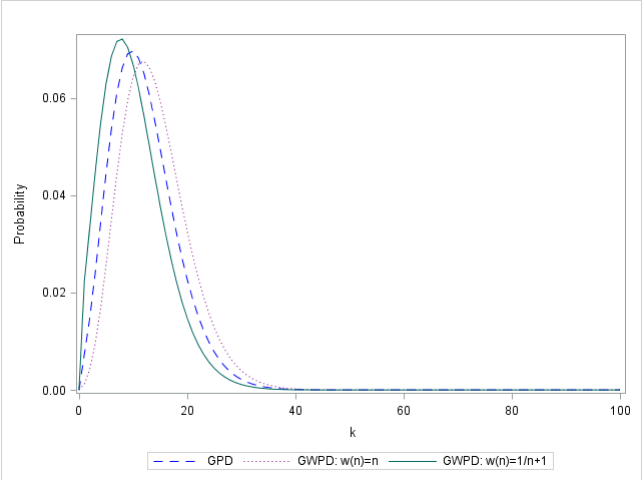
Table 3: Numerical summary of statistics for $\lambda = 6$ and θ varying



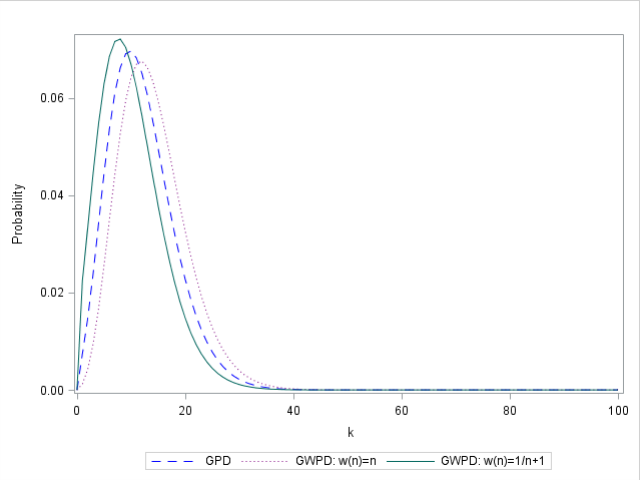
(a) For $\theta = 0.5$ and $\lambda = 3$



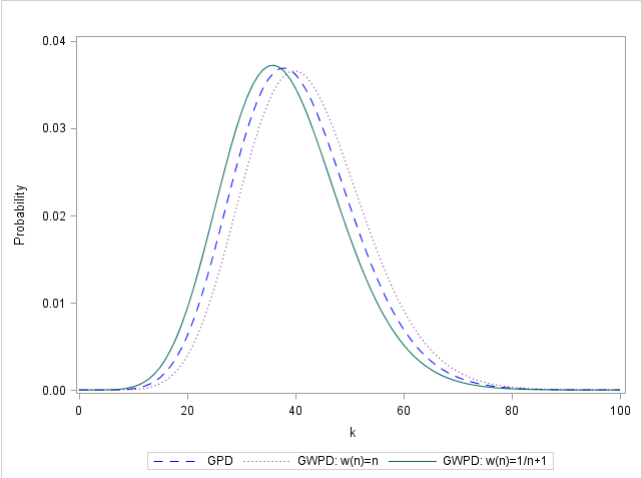
(b) For $\theta = 0.2$ and $\lambda = 6$



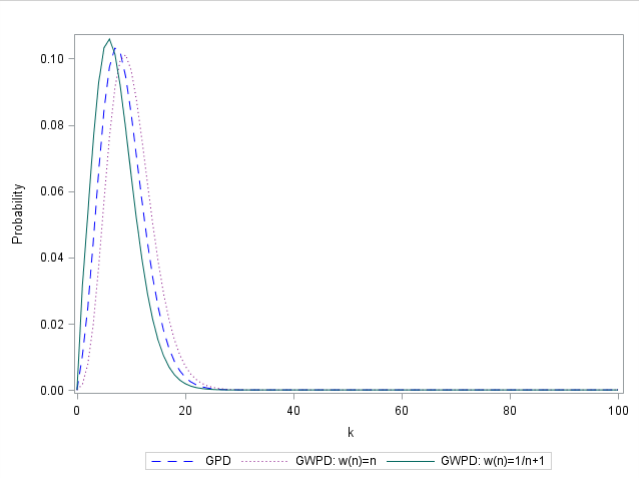
(c) For $\theta = 0.5$ and $\lambda = 6$



(d) For $\theta = 0.5$ and $\lambda = 6$



(e) For $\theta = 0.5$ and $\lambda = 20$



(f) For $\theta = 0.7$ and $\lambda = 6$

Figure 1: The Probability mass functions under different parameter values

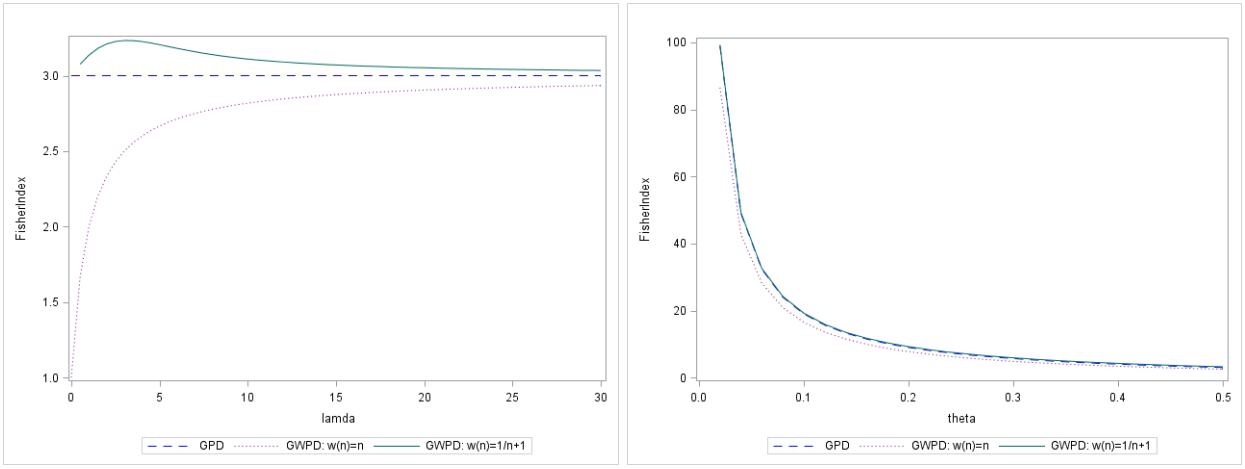
6.2 The Fisher indices

Keeping θ constant and letting λ vary

We obtain Figure 2(a) which shows how the Fisher indices for each distribution changes as λ increases. In Figure 2(a) we see that when $\theta = 0.5$ the Fisher index of GPD remains constant as λ increases. However the Fisher index of the GWPD: $w(n) = n$ is seen to increase steepily at first and then approaches a limit. The Fisher index of the GWPD: $\frac{1}{n+1}$ increases to a certain level, then steadily decreases and approaches a limit. The limit approached by both the GWPDs is the Fisher index of the GPD. From Figure 2(a) it can be seen that the GWPD: $\frac{1}{n+1}$ is over-dispersed with respect to the GPD and the GWPD: $w(n) = n$ is under-dispersed with respect to the GPD.

Keeping λ constant and letting θ vary

Figure 1(b) shows how the Fisher indices for the distributions change when λ remains constant and θ increases. The Fisher indices decrease as θ increases but the difference between the three distributions are very small. It can still be seen that the GWPD: $\frac{1}{n+1}$ is over-dispersed with respect to the GPD and the GWPD: $w(n) = n$ is under-dispersed with respect to the GPD.



(a) Fisher indices for $\theta = 0.5$ and λ varying

(b) Fisher indices for $\lambda = 3$ and θ varying

Figure 2: The graphical representation of the Fisher indices

7 Conclusion

In this essay the distributions, moments and Fisher indices of the geometric Poisson distribution and two geometric weighted Poisson distributions with weight functions $w(n) = n$ and $w(n) = \frac{1}{n+1}$ are derived and compared. The main aim was to compare the dispersion of the GPD to the two variations of the GPD (the GWPDs). The dispersion was measured using the Fisher index. A distribution is over-dispersed if $FI(X) > 1$, equi-dispersed if $FI(X) = 1$ and under-dispersed if $FI(X) < 1$. The GPD is always over-dispersed, that is the Fisher index is always greater than one. Relative to the GPD the GWPD:$w(n) = n$ and the GWPD:$w(n) = \frac{1}{n+1}$ are respectively under and over-dispersed. This difference is more pronounced

for fixed values of θ and varying λ. The three distributions were also studied and compared graphically to see how the probability mass function and Fisher index change for different parameter values.
We found, the GPD can be easily modified to give a set of flexible distributions (the GWPDs) that can be fitted to data allowing for different dispersions hence increasing its application to different types of data. Since the GPD is overdispersed it is suited for clumped, concentrated data, the level of concentration then determines which GWPD to modify the GPD to.

References

[1] *SAS Software, Version 9.4 of SAS System for Windows, Copyright 2016, SAS Institute Inc. Cary, NC, USA.*

[2] Masood Anwar and Munir Ahmad. On some properties of geometric Poisson distribution. *Journal of Statistical Computation and Simulation*, 30(2):233–244, 2014.

[3] Nihal Ata and Gamze Özel. Survival functions for the frailty models based on the discrete compound Poisson process. *Journal of Statistical Computation and Simulation*, 83(11):2105–2116, 2013.

[4] Ching-Wen Chen, Paul H Randolph, and Tian-Shy Liou. Using CUSUM control schemes for monitoring quality levels in compound Poisson production environment: the geometric Poisson process. *Quality Engineering*, 17(2):207–217, 2005.

[5] Joan Del Castillo and Marta Pérez-Casany. Weighted Poisson distributions for overdispersion and underdispersion situations. *Annals of the Institute of Statistical Mathematics*, 50(3):567–585, 1998.

[6] JB Douglas. Pólya–aeppli distribution. *Encyclopedia of Statistical Sciences*, 1986.

[7] RA Fisher. The effect of methods of ascertainment upon the estimation of frequencies. *Annals of eugenics*, 6(1):13–25, 1934.

[8] Norman L Johnson, Adrienne W Kemp, and Samuel Kotz. *Univariate Discrete Distributions*, volume 444. John Wiley & sons, 2005.

[9] Leda D Minkova and N Balakrishnan. Compound weighted Poisson distributions. *Metrika*, 76(4):543–558, 2013.

[10] Gregory Nuel. Cumulative distribution function of a geometric Poisson distribution. *Journal of Statistical Computation and Simulation*, 78(3):385–394, 2008.

[11] Gamze Özel and Ceyhan Inal. The probability function of a geometric Poisson distribution. *Journal of Statistical Computation and Simulation*, 80(5):479–487, 2010.

[12] Paul Randolph and Mehmet Sahinoglu. A stopping rule for a compound Poisson random variable. *Applied Stochastic Models and Data Analysis*, 11(2):135–143, 1995.

[13] Stéphane Robin. A compound Poisson model for word occurrences in DNA sequences. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 51(4):437–451, 2002.

[14] Stephane Robin, Sophie Schbath, and Vincent Vandewalle. Statistical tests to compare motif count exceptionalities. *BMC Bioinformatics*, 8(1):84, 2007.

[15] Rhonda J Rosychuk, Carolyn Huston, and Narasimha GN Prasad. Spatial event cluster detection using a compound Poisson distribution. *Biometrics*, 62(2):465–470, 2006.

Appendix

The data analysis for this essay was performed using SAS software [1], the sas codes that were used are given below.

SAS code for the probability mass functions.

```
/*Code to commute values of the Probability mass functions and the graphical representation.*/
    /*For k>=n*/
    data probmass;
    theta = 0.5;
    Lamda=3;
    do k=0 to 100;
    Probability=0;
    sumgwpd1=0;
    sumgwpd2=0;
    do n=1 to k;
    GPD=exp((-1)*lamda)*((lamda**n)/(fact(n)))*(comb(k-1,n-1))*(theta**n)*((1-theta)**(k-n));
    GWPD1=((lamda**(n-1))*(exp((-1)*lamda))/(fact(n-1)))*(comb(k-1,n-1))*(theta**n)*((1-theta)**(k-n));
    GWPD2=((exp((-1)*lamda)*(lamda**(n+1)))/(((1-exp((-1)*lamda))*(fact(n+1)))))*(comb(k-1,n-1))*
    (theta**n)*((1-theta)**(k-n));
    Probability=Probability + GPD;
    sumgwpd1=sumgwpd1 + GWPD1;
    sumgwpd2=sumgwpd2 + GWPD2;
    END;
    OUTPUT;
    end;
    run;
    proc print data=probmass;
    run;
    PROC SGPLOT DATA = probmass;
    SERIES X = k Y = Probability / lineattrs=(color=blue pattern=dash) LEGENDLABEL = 'GPD';
    SERIES X = k Y = sumgwpd1 /lineattrs=(color=PURPLE pattern=DOT) LEGENDLABEL = 'GWPD:
    w(n)=n';
    SERIES X = k Y = sumgwpd2 / LEGENDLABEL = 'GWPD: w(n)=1/n+1';
    TITLE ' ';
    RUN;
```

SAS code for the Fisher Indices.

```
/*Code to commute the Fisher indices and the graphical representation when λ is increasing.*/
    data fisher;
    theta=0.5;
    gp=0;
    gwp1=0;
    GWP2=0;
    do lamda= 0 to 30 by 0.5;
    FisherIndex=(2-theta)/theta;
    gwp1=((1-theta)+(2-theta)*lamda)/(theta*(lamda+1));
    GWP2=((2-theta)/theta )+(((1-exp(-lamda))**2)-(lamda**2)*exp(-lamda))/((1-exp(-lamda))*
    (lamda-1+exp(-lamda)));
    OUTPUT;
    end;
```

```
run;
proc print data=fisher;
run;
PROC SGPLOT DATA = fisher;
SERIES X = lamda Y = FisherIndex/ lineattrs=(color=blue pattern=dash) LEGENDLABEL = 'GPD';
SERIES X = lamda Y = gwp1 / lineattrs=(color=PURPLE pattern=DOT) LEGENDLABEL = 'GWPD:
w(n)=n';
SERIES X = lamda Y = gwp2 / LEGENDLABEL = 'GWPD: w(n)=1/n+1'; TITLE ' ';
RUN;


/*Code to commute the Fisher indices and the graphical representation. For θ increasing*/
fisher;
LAMDA=3;
gp=0;
gwp1=0;
GWP2=0;
do theta= 0 to 0.5 by 0.02;
FisherIndex=(2-theta)/theta;
gwp1=((1-theta)+(2-theta)*lamda)/(theta*(lamda+1));
GWP2=((2-theta)/theta )+(((1-exp(-lamda))**2)-(lamda**2)*exp(-lamda))/((1-exp(-lamda))*
(lamda-1+exp(-lamda)));
OUTPUT;
end;
run;
proc print data=fisher;
run;
PROC SGPLOT DATA = fisher;
SERIES X = theta Y = FisherIndex/ lineattrs=(color=blue pattern=dash) LEGENDLABEL = 'GPD';
SERIES X = theta Y = gwp1 / lineattrs=(color=PURPLE pattern=DOT) LEGENDLABEL = 'GWPD:
w(n)=n';
SERIES X = theta Y = gwp2 / LEGENDLABEL = 'GWPD: w(n)=1/n+1';
TITLE ' ';
RUN;
```

Gated recurrent neural networks for language modeling

Cedric Oeldorf 13413636

STK795 Research Report

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

Supervisor: Dr Alta de Waal

Department of Statistics, University of Pretoria

2 November, 2016

Abstract

Language models are pivotal in the development of many applications such as speech recognition and translation. Whilst the n-gram model has been the leading algorithm, the following research proposes the use of gated recurrent neural networks for the training of a German language model. The theory behind language models and neural networks will be covered, as well as an empirical evaluation of the long short-term memory architecture and the gated recurrent unit architecture.
We found that the one-layer LSTM performed better than the three other tested models. There was a noticable gap between the performance of one-layer and two-layer architectures.

Declaration

I, *Cedric Rolf Oeldorf,* 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.

_____
Cedric Rolf Oeldorf

_____
Dr Alta de Waal

_____
Date

Acknowledgments

This research is supported by the Centre for Artificial Intelligence Research (CAIR). CAIR is managed by the Meraka Institute, CSIR. This research is additionally supported by STATOMET, which is managed by the Department of Statistics at the University of Pretoria.
The author is grateful to Ottokar Tilk and Botty Dimanov for their extensive guidance and advice in the construction and training of the neural networks applied to language modeling in Python.

Contents

1 Introduction **7**

2 Literature Review **8**

3 Background Theory **9**
3.1 Language Models . . . 9
3.2 Evaluation Techniques . . . 9

4 Artificial Neural Networks **10**
4.1 Activation Functions . . . 10
4.1.1 Logistic Sigmoid Function . . . 10
4.1.2 Hyperbolic Tangent Function . . . 10
4.1.3 Softmax Function . . . 10
4.2 Architecture . . . 11
4.2.1 Recurrent Neural Networks . . . 11
4.2.1.1 Long Short-Term Memory . . . 11
4.2.1.2 Gated Recurrent Unit . . . 12

5 Experimental Design **13**
5.1 Data . . . 13
5.2 Python Library . . . 14
5.3 Models . . . 14
5.3.1 Layers . . . 14
5.3.1.1 Embedding Layer . . . 14
5.3.1.2 GRU and LSTM layers . . . 15
5.3.1.3 Dropout Layer . . . 15
5.3.1.4 Dense Layer . . . 15
5.3.2 One-Layer Architecture . . . 15
5.3.3 Two-Layer Architecture . . . 15

6 Evaluation **16**
6.1 Cross-Entropy Loss . . . 17
6.1.1 One Layer Model . . . 17
6.1.2 Two-Layer Model . . . 18
6.2 Perplexity . . . 18

7 Conclusion **19**

Appendix **21**

List of Figures

1 Recurrent Neural Network . . . 11
2 Long Short-Term Memory Cell . . . 12
3 Gated Recurrent Unit . . . 13
4 Preprocessed Data . . . 14
5 Words in a vector space . . . 15
6 One-Layer architecture . . . 15
7 Two-Layer architecture . . . 16
8 One-Layer architecture evaluation . . . 17
9 Two-Layer architecture evaluation . . . 18

List of Tables

1 Perplexity scores . . . 19

1 Introduction

Already as far back as the 1950s, researchers aspired to allow computers to exhibit intelligence in the way a human does. Although the problem has not been solved, the past half a century gave rise to several algorithms that represent the stepping stones towards artificially intelligent machines. We have seen machines recognize and caption complex images [18] and even drive cars on public roads [6]. Whilst these are astonishing accomplishments, there is one task for which a solution has been sought-after for decades, that is the ability of a machine to successfully communicate with a human. Alan Turing devised a test in 1950 that serves to examine the capacity of a machine to exhibit intelligent human behavior. In this test, a machine and a human communicate by asking and answering questions. The machine has to fool the human into thinking that it is also a human. Alan Turing called this test the *Imitation Game* [12].
In order to build a machine that can communicate in natural language to the extent of passing Turing's test, we would have to start by giving it a fundamental understanding of the basic patterns and connections within a human language. A solution to such a representation of a language is the so called *language model.*
Language models have swiftly gained importance since technologies in fields such as speech recognition, machine translation and text-to-speech systems have become integral parts of daily life. Examples of such technologies include Apple's *Siri* and *Google Translate*, both of which rely on powerful language models in order to approximate the words that make up your recorded speech or text.
We can describe a language model as being a distribution function of the next word in a sentence, given a previous word or phrase [2]. The model should compute a score, $P(word)$, which can represent the probability of a sequence of words to be part of a given language. This probability can be interpreted as a score on how fluent the input was in the given language, detect grammatical inconsistencies and even derive logic and world knowledge.
Historically, the n-gram model has been the flagship when estimating language models. It has become the dominant model of use for various reasons such as its simplicity and performance [3]. The undoing of this seemingly superior language modeling technique lies in the use of grammatically complex languages such as Czech, Arabic or German [13]. N-gram language models face major difficulties when applied to languages where word order is challenging and verbs are highly irregular.
We will propose the estimation of language models with an approach that attempts to mimic the human brain through computer simulation called an *artificial neural network* (ANN). ANNs simulate a network of biological neurons in form of interconnected nodes that work in unison in order to process information [11]. The downfall of ANNs was that computers were not powerful enough to process the large amounts of data needed by the neural nets, a situation which has changed. The past few years have shown considerable progress to the extent that we now have what is called *deep learning.* This is essentially a set of techniques for training an ANN with more than one hidden layer of neurons, giving the neural net a deep architecture and thus bringing us a step closer to true artificial intelligence. This research will propose the use of two deep learning architectures based on recurrent neural networks (RNN) for training a language model specifically for the German language. The difference between RNNs and regular feed-forward neural networks is that RNNs allow signals to travel backwards. Output from earlier computations can be fed back into the network thus making them exceptionally powerful sequence models. This power comes with a price, they are notoriously difficult to train due to the exploding and vanishing gradients problem [15]. The issue of vanishing gradients was successfully circumvented with the development of the *Long Short-Term Memory* (LSTM) architecture [17], a mechanism that controls the amount of memory the neural network retains. Recently, another recurrent unit was developed called the *Gated Recurrent Unit* (GRU), which is said to be more efficient in terms of processing and data than the LSTM [10]. What makes these two architectures attractive for language modeling is that they allow the recurrent neural network to retain a longer memory than usual. This is important when considering in language, how many previous words of a sentence need to be given thought on when placing the next word. We will empirically evaluate both of these architectures in order to conclude which is more suitable for language modeling.
The hypothesis in this case is whether both the LSTM RNN and the GRU RNN are good algorithms when being utilized for the training of a German language model on small data. We will be using a collection of German works by Franz Kafka from the Gutenberg Project to train our model. In order to evaluate and compare our models, we will consider two methods, namely perplexity and the rate at which the cross-entropy

loss is minimized. The structure of this paper is as follows:
In Section 2 we review the literature on what has been done in the field of artificial neural networks and language modeling. Section 3 will introduce language models and describe the evaluation techniques that we will utilize in order to measure the performance of our language models.
Section 4 is dedicated to the conceptualization of artificial neural networks and the distinct architectures that will be implemented for language modeling. The section will start with a general background and description of an artificial neural network and then move on to recurrent neural networks followed by the LSTM and GRU networks.
Thereafter, Section 5 covers both the methodology and the experiments. This includes a description of the Python libraries, data used, our experimental design and the results. Two recurrent neural network architectures and an n-gram model will be trained and applied in order to objectively compare the performance of each.
Section 6 evaluates our experimental results and Section 7 concludes if using the two RNN architectures are feasible in a limited data environment.

2 Literature Review

This section serves to review the research that has already been done in this field and on this topic. The listed articles lay the foundation for the theory in this reseach.

N-gram

Class-based n-gram models of natural language by P. Brown et al. [2]
The article by Brown et al. proposes the use of a class-based n-gram model for the prediction of a word given previous words. The theory behind language models is discussed and the basics behind n-gram models are conceptualized before the article goes into too much depth for purposes of our research.

Artificial Neural Networks

Machine Learning: A Probabilistic Perspective by K.P. Murphy [14]
Chapter 28 of Murphy's book introduces deep neural networks and describes the different architectures that these artificial neural networks can take on. Importantly, he also describes several applications of deep networks.

Artificial neural networks: A tutorial by A.K. Jain and J. Mao [9]
The fundamentals of artificial neural networks are covered in this article. The basic concepts concerning the structure of ANN's are depicted both in theory and graphically. The article serves to give a reader with little knowledge about the topic a good idea of what ANN's are.

Learning Deep Architectures for AI by Y. Bengio [1]
This paper, by one of the world's leading deep learning researchers, Yoshua Bengio, illustrates the principle behind a *deep architecture* and also relates the theoretical advantages of these deep architectures.

Recurrent Neural Networks

Statistical Language Models Based on Neural Networks by T. Mikolov [13]
This paper claims that a statistical language model based on a simple recurrent neural network outperforms other state-of-the-art technologies. This will be the focus of our research. Mikolov mathematically defines language modeling, describes the architecture of recurrent neural networks and empirically evaluates the results of applying the RNN as a language model. These are the core sections of our research and will serve as a fundamental paper to our dissertation.

LSTM Neural Networks for Language Modeling by M. Sundermeyer et al. [17]

This paper describes a special kind of RNN which we will be using in our paper. It is the long short-term memory network, which is capable of learning long-term dependencies in the data. Whilst RNN's do preserve previous information, they cannot use a previous result and connect it to the present assignment. LSTM's have a slightly different structure which allows for exactly this to happen. The paper goes into the theory behind LSTM's, which we will use to describe the architecture that we want to apply.

An Empirical Exploration of Recurrent Network Architectures by R. Jozefowicz et al. [10]
This paper empirically evaluates different RNN architectures. A theoretical insight into both LSTM's and GRU's is given which lays the foundation of the mathematical theory in this paper.

3 Background Theory

3.1 Language Models

As mentioned in the introduction, language models have been around for a couple of decades solving problems in the fields such as speech recognition, machine translation and automatic spelling correction. The principle of a language model is based on, given a sequence of words denoted as \emph{X}, the computation of the probability of the sequence. The most frequently used technique for language modeling is the n-gram model, in which the chain rule is used to compute the probability of a sequence of words.

$$\begin{aligned} P(X) &= P(x_1, x_2, x_3...., x_n) && (1) \\ &= P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1, x_3)...P(x_n \mid x_1, .., x_{n-1}) && (2) \end{aligned}$$

where $P(X)$ is the joint probability of the individual words and x_n are individual words. Once we are able to compute this probability, we can start predicting words given previous words using conditional probability theory.
Google consider the same approach. When we type a query into the search bar, it predicts what the most likely query is, given what has been entered. With these formulas in mind, one of the core issues of language modeling becomes apparent. That is the problem of data sparsity. There is no data set that encompasses every combination of every sentence that will ever be constructed in a language. Thus we require ways to allow the estimation of probabilities concerning words combinations that were not seen in training, which notably complicates the task.

3.2 Evaluation Techniques

The core concept used when measuring the usefulness of a language model is called entropy. It can represent the amount of information a language model holds and how well it performs at a certain task [7]. In our paper this was calculated as cross-entropy loss where p represents our true label and q is the predicted value. Thus the loss is calculated by this mathematical expression:

$$H(p, q) = -\sum p_i \log q_i \tag{3}$$

This represents a measure of distance between the predicted and true values. Thus, the smaller our cross-entropy loss, the more accurate our model. This brings us to our next measure, perplexity, which is nothing more than 2 to the power of the cross-entropy loss, H:

$$PP = 2^H \tag{4}$$

Perplexity has been the standard for measuring the performance of language models. It can be described as the average number of decisions a random variable is forced to undergo [7]. We will report both cross-entropy loss and perplexity for our evaluation.
Although we will not be utilizing this next measure for the purposes of this paper, it is important to mention

the word error rate as an evaluation technique. Usually there is a stable relationship between perplexity and word error rate defined by T. Hain as:

$$WER \approx k \times \sqrt{PP} \tag{5}$$

where k is a constant and PP represents the perplexity.

4 Artificial Neural Networks

An *Artificial Neural Network* (ANN) is an algorithm inspired by the way the humans central nervous system works. Just as biological neurons are connected in a network, ANN's contain layers of nodes that are interconnected by weighted connection lines [9]. When training the model, these weights are adjusted to the point at which the network can successfully perform tasks such as classification, clustering and pattern recognition.

4.1 Activation Functions

Before we dive into the different neural network architectures relevant to this paper, we will look the functions lying at the hearts of our neurons. Each neuron has the purpose of computing a single output from several inputs using a linear combination of input weights [19]. Mathematically, this process can be depicted as:

$$y = f(\sum_{i=1}^{n} w_i x_i + b) \tag{6}$$

where w_i is the vector of weights, x_i is the vector of inputs and b represents the bias. These variables are a function of f, a non-linear transformation function. The following subsections will describe a few of the non-linear transformation functions utilized in this paper.

4.1.1 Logistic Sigmoid Function

The logistic sigmoid function is a special case of the logistic function which introduces non-linearity into our model. It is mathematically defined as follows:

$$f(x) = \frac{1}{(1 + e^{-x})} \tag{7}$$

The function follows an "S" shaped curve and transforms numeric input into a value between 0 and 1 which can be interpreted as probabilities. This is especially important during model evaluation.

4.1.2 Hyperbolic Tangent Function

The hyperbolic tangent function *(tanh)* also serves to relate input and output in a neural network as follows:

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{8}$$

$$= \tanh(x) \tag{9}$$

it outputs a value between -1 and 1, which is in essence a re-scaling of the logistic sigmoid function.

4.1.3 Softmax Function

The softmax function, also called the normalized exponential, is similar to the logistic function in that it takes a vector of real values and "squashes" them into same-sized vector with values between 0 and 1. This function is commonly used as the output layer function in neural networks and is mathematically defined as:

$$f(x_i) = \frac{e^{x_i}}{\sum_i e^{x_i}} \tag{10}$$

4.2 Architecture

Based on the way the nodes in an ANN are connected, we can identify different categories of ANN's. The basic structure of an ANN is a feed-forward neural network in which connections only flow forward. Neurons are organized in layers where the bottom layer receives the input and the upper layers receive the preceding layers output [9]. Feed-forward networks do not hold memory, thus their preceding network states do not influence future computations of input [9]. Due to the fact that we need to consider all the preceding words in our model, this type of neural network would not work as it does not send feedback. The solution to this problem lies in the recurrent neural network architecture.

4.2.1 Recurrent Neural Networks

A recurrent neural network connects each classifier to the input at each time step just as the feed-forward network. The difference lies in what is called a recurrent connection, which connects your model to the past at each time step. Thus, unlike in the feed-forward neural network, we do not assume the input and outputs to be independent, which makes it optimal for processing sequential data [8].
Figure 1 depicts a basic RNN. On the left side we see the input x running into the node. This node does not only output data to "o", but also into a node we call a *recurrent unit*. When this is unfolded we can see how the recurrent unit is updated at each time step based on all previous time steps. This unrolled format has similarities with sequences or lists.

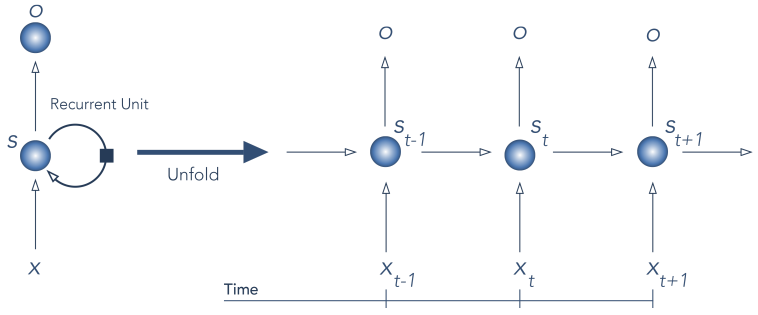


Figure 1: Recurrent Neural Network

In order to compute the parameter updates, we need to backpropagate derivatives through time as far back as we can computationally afford. With so many correlated updates for individual weights, we find ourselves with two notorious issues native to RNN's, namely the *exploding gradient* and *vanishing gradient* problems [17]. Either your gradients grow exponentially to infinity or they reduce to zero, which results in your model not learning anything from the information passed through it. Vanishing gradients leave the model with only a short memory, thus RNNs are only effective for a small amount of time steps. Both exploding and vanishing gradients can be circumvented by limiting and controlling the amount of memory the neural network retains. This is achieved through the nature of the gated architectures used in this research. These architectures are in essence RNNs, with the difference lying in the recurrent unit, which is replaced by a cell such us the two explained in the following two subsections.

4.2.1.1 Long Short-Term Memory LSTM's are a type of RNN with the capability of learning long-term dependencies without encountering the vanishing or exploding gradient problems. The basic architecture of an LSTM is the same as an RNN, the difference lying in the recurrent unit.

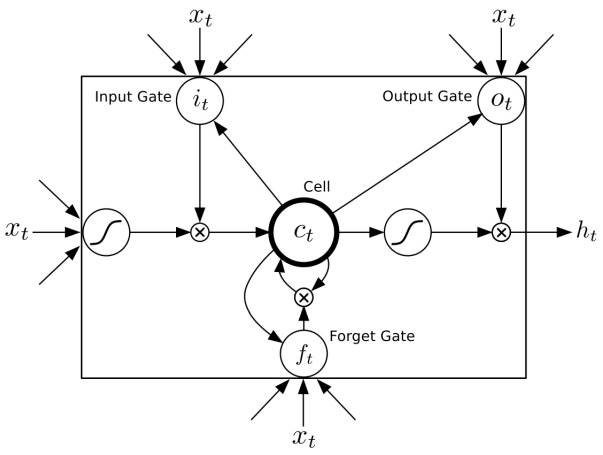


Figure 2: Long Short-Term Memory Cell

Figure 2[1] illustrates such an LSTM cell. Mathematically, its implementation can is described as follows:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) && (11)\\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) && (12)\\
c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) && (13)\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) && (14)\\
h_t &= o_t \tanh(c_t) && (15)
\end{aligned}
$$

σ is the sigmoid function as described in section 3.1.1. i is the input gate, f the forget gate, o the output gate and c the cell activation vectors [5]. This comes together to calculate the hidden state h_t. The input, forget and output functions are technically identical besides using different parameters. The output of their respective sigmoid functions lies between 0 and 1, which, when multiplied with another vector, will determine how much of that vector is passed through. This is why these three components are called *gates*. The first gate is the input gate, it stipulates how much of the networks most recent state enters the cell. The forget gate then decides how much of the preceding state is retained and the output gate determines how much of this state within the memory cell will be carried on to the next time step.
c_t represents our actual memory cell. It multiplies the forget gate with the previous memory $c_{t-1}$and adds to that what the input gate lets in. This in combination with the output gate is used to compute the new hidden state h_t.

4.2.1.2 Gated Recurrent Unit The Gated Recurrent Unit (GRU) was introduced by Cho et al. in 2014. It is conceptually very similar to the LSTM and several papers have found the GRU to outperform the LSTM on certain engagements [10]. The equations of the GRU will not look too foreign after the section on the LSTM.

$$
\begin{aligned}
z &= \sigma(x_t U^z + s_{t-1}W^z) && (16)\\
r &= \sigma(x_t U^r + s_{t-1}W^r) && (17)\\
h &= tanh(x_t U^h + (s_{t-1} \odot r)W^h) && (18)\\
s_t &= (1-z) \odot h + z \odot s_{t-1} && (19)
\end{aligned}
$$

[1]Image courtesy of: A. Graves et al. (2013)

Instead of three gates the GRU only has two, as seen in Figure 3[2], the reset gate r and the update gate z. The reset gate computes the way the previous memory and the new input is merged together whereas the update gate determines how much of that previous memory will be retained [10].

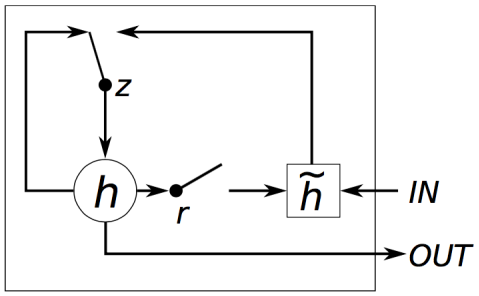


Figure 3: Gated Recurrent Unit

5 Experimental Design

5.1 Data

As training data, several e-books were extracted from the Gutenberg Project. This website has a large collection of free e-books in German which can be downloaded in text format. Two works of Franz Kafka were combined to construct the corpus.
Neural networks take numeric vectors as input, which is why some data preprocessing had to be done in order to embed the words of the corpus into a vector space. Due to memory constraints and exceptionally long training times, we limited our sentences to those that are under 50 words in length. An extract of our corpus can be seen below. It depicts the first sentence in our corpus before any preprocessing was done:

```
schreibweise und interpunktion des originaltextes
wurden uebernommen; lediglich offensichtliche
druckfehler wurden korrigiert.
```

The sentences were then tokenized, which results in a list of sentences, each with a sub-list of words:

```
[u'schreibweise', u'und', u'interpunktion', u'des',
u'originaltextes', u'wurden', u'uebernommen', u';',
u'lediglich', u'offensichtliche', u'druckfehler',
u'wurden', u'korrigiert']
```

In order to improve the accuracy of the model, the tokens "<Start>" and "<End>" were added to the sentences to demarcate their respective beginning and endings.

```
[u'<Start>', u'schreibweise', u'und', u'interpunktion',
u'des', u'originaltextes', u'wurden', u'uebernommen',
u';', u'lediglich', u'offensichtliche', u'druckfehler',
u'wurden', u'korrigiert', u'<End>']
```

[2]Image courtesy of Denny Britz at http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/

Thereafter, the tokens were indexed in a list and one-hot encoded [3] to create matrices in which each sentence is represented by a row of 1's and 0's. The matrices X and y were then padded with 0's so that the rows are all of equal length, this is a requirement when using the Keras package. In Figure 4 we visualize what our data will look like after preprocessing:

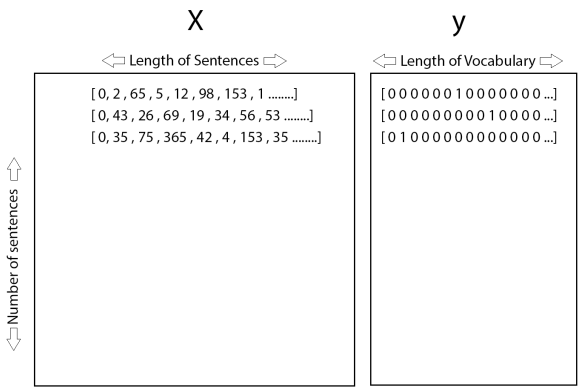


Figure 4: Preprocessed Data

Each row in the X matrix is a vectorized sequence of words where each word is represented by an index. The y matrix represents the next word in the sequence as a one-hot encoded table of our vocabulary. Our testing data is compromised of 20% of the full data set.

5.2 Python Library

As we are running all our experiments in Python using the Keras library [4] to construct our neural networks. It has the capability of running on top of Theano [4], which allowed us to run our models on a Graphics Processing Unit (GPU), drastically decreasing computation time.

5.3 Models

We keep our models fairly simple in order to make it easier to understand the inner working of the models during training.

5.3.1 Layers

5.3.1.1 Embedding Layer The embedding layer is that at which our words get mapped into a continuous vector space. This is the same as what the *Word2Vec* algorithm does. After the data runs through this layer, the word indexes in the X matrix will be turned into dense vectors. eg. [[3],[53]] will be turned into [[-0.56,2],[-9,5]], thus leaving us with words represented in a space similar to Figure 5[5]:

[3] See http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing. OneHotEncoder.html
[4] See http://deeplearning.net/software/theano/
[5] Image courtesy of Benjamin Bolte at http://benjaminbolte.com/blog/2016/keras-language-modeling.html

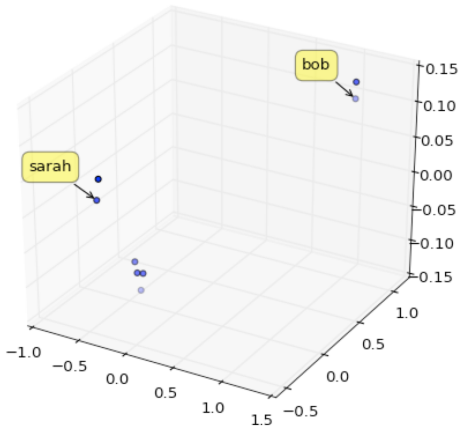


Figure 5: Words in a vector space

5.3.1.2 GRU and LSTM layers These are as described in the theory section. They take the output of the embedding layer to learn patterns in the data.

5.3.1.3 Dropout Layer RNN's are notorious for easily over-fitting the data. The dropout layer is a regularization technique that sets a certain percentage of the nodes to 0 at each batch iteration [16]. This is a method that was first suggested by Geoffrey Hinton in his coursera course.

5.3.1.4 Dense Layer This layer simply serves to connect each neuron to each neuron in the next layer.

5.3.2 One-Layer Architecture

Our one-layer neural network architecture can be visualized as as in Figure 6:

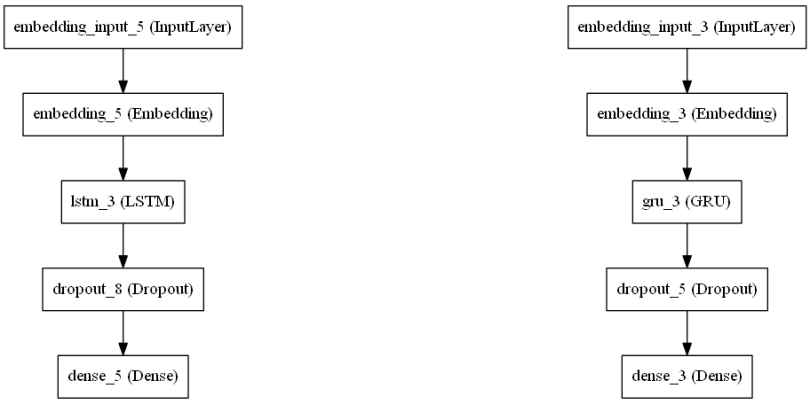


Figure 6: One-Layer architecture

It is apparent that these are very basic models with only one GRU/LSTM layer. This layer will have a total of 128 hidden nodes and was trained using a batch size of 32.

5.3.3 Two-Layer Architecture

In this architecture each of the two LSTM/GRU layers have 64 hidden nodes, which leaves us with a total of 128 hidden nodes. In all other aspects the parameters of this architecture are the same as the more basic models. Figure 7 depicts our two-layer neural network.

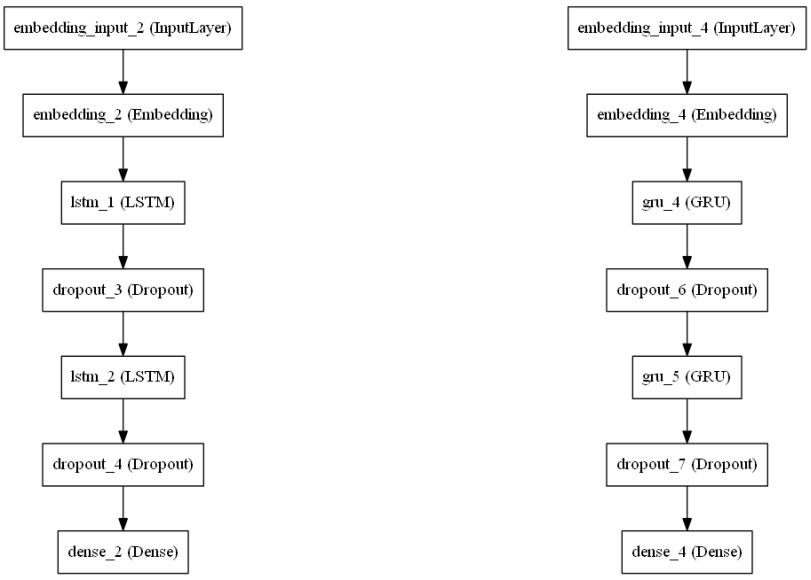


Figure 7: Two-Layer architecture

6 Evaluation

In the following graphics we plotted the minimization of our loss function a number of iterations for both our LSTM and our GRU architectures. The red line represents our validation loss, this is the loss that was computed on our test data set. The blue line represents the loss computed during training.

6.1 Cross-Entropy Loss

6.1.1 One Layer Model

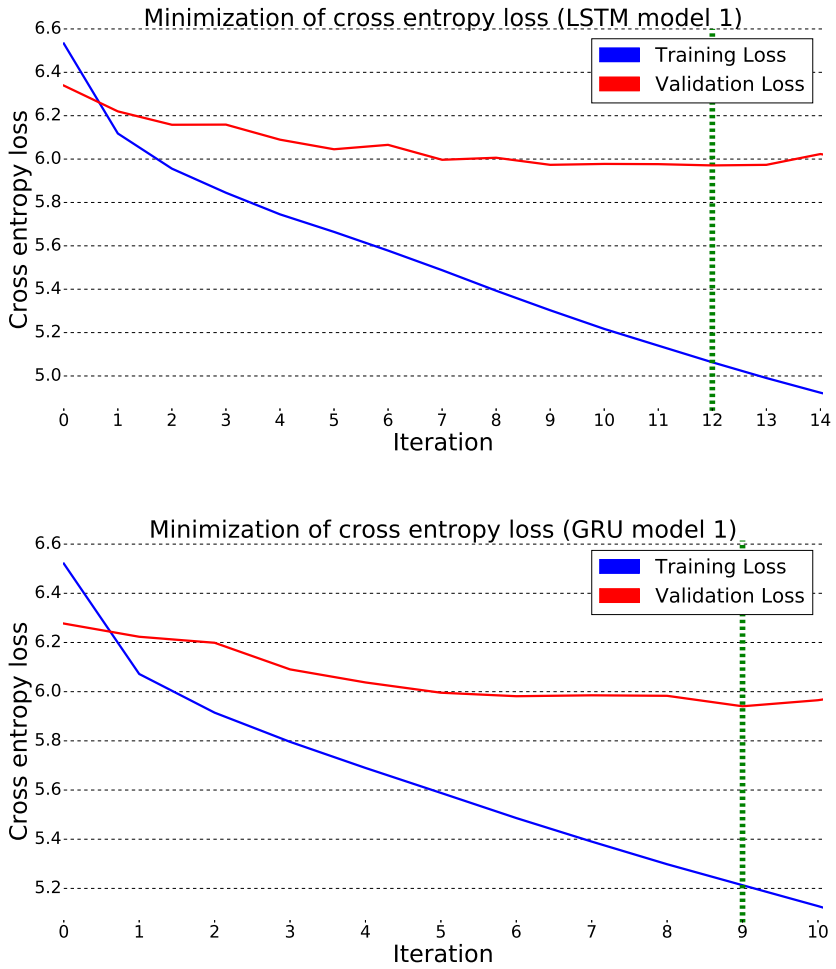


Figure 8: One-Layer architecture evaluation

By Figure 8, it is apparent that the models start over-fit the data starting from approximately the twelfth and ninth iteration respectively. Over-fitting happens when the model starts modeling the noise in the training data [16]. We defined this the moment as the point at which our loss on our validation set starts increasing indefinitely. The green dotted line marks the iteration at which our validation loss was at its minimum. It seems that both models behaved in a similar fashion, although the GRU over-fitted the data a little quicker than the LSTM.

6.1.2 Two-Layer Model

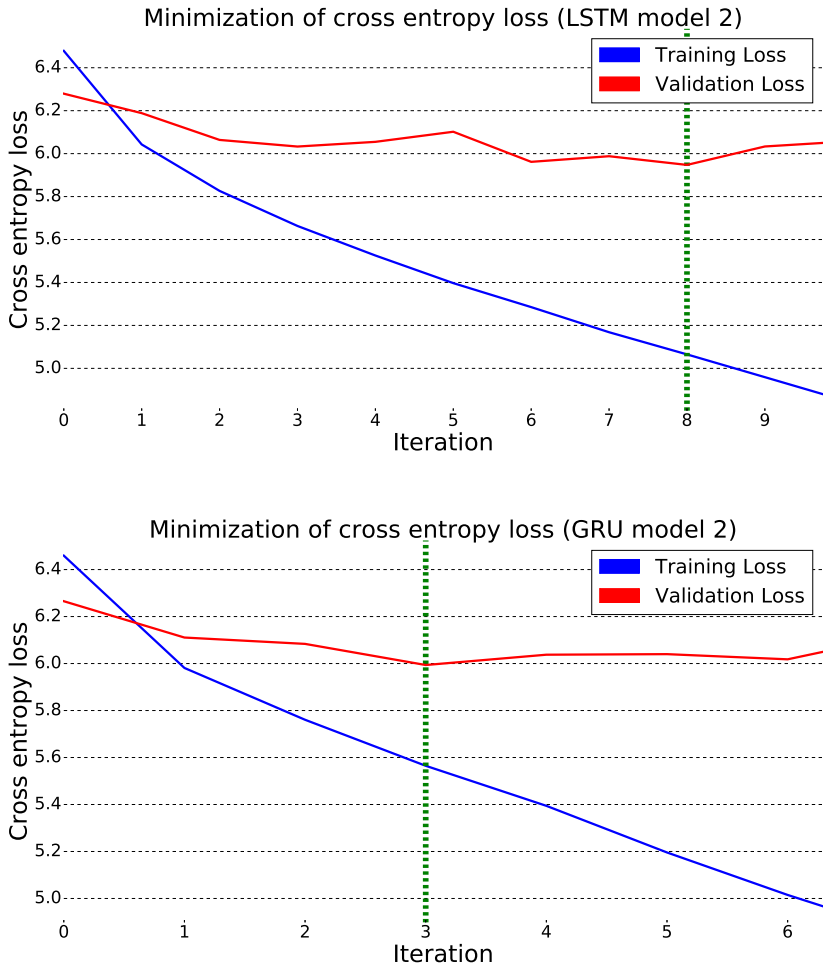


Figure 9: Two-Layer architecture evaluation

Figure 9 shows that, in terms of over-fitting the data in comparison to the one layer models, two-layer models perform poorly. Although the LSTM reaches a lower minimum in terms of validation loss, it did not outperform the GRU model by much.

6.2 Perplexity

The perplexity is reported on all four models in Table 1. To put these into perspective, using an LSTM, Josefowicz et al (2015) achieved a perplexity of 79.83 training on the Penn Tree-Bank corpus, which has around one million words in it. This is considerably larger than our corpus. We trained our models using the number of iterations at which each architecture converged in the previous subsection. Our perplexity scores were calculated using the values resulting from 10-fold cross-validation, after which we tested whether the scores are significantly different using several two-sample t-tests. We tested our hypothesis at a 95% level of significance. The null hypothesis of equal means was rejected for each combination of mean perplexity scores.

	Mean	Standard Deviation
One-Layer GRU	79.23	0.069
One-Layer LSTM	77.91	0.044
Two-Layer GRU	85.97	0.187
Two-Layer LSTM	81.84	0.012

Table 1: Perplexity scores

The best performance, although not by far, was achieved by the one-layer LSTM model, with the one-layer GRU coming in at a close second. The two layer architectures performed relatively poorly, for which over-fitting is to blame.

7 Conclusion

We have evaluated two recurrent neural network architectures, the LSTM and the GRU, when applied to modeling a morphologically complex language using a limited text corpus. Our perplexity scores support our hypothesis, which leads us to the conclusion that, when your vocabulary is limited, these two architectures still perform fairly well when generalizing on to unseen text. The one-layer LSTM architecture performed the best, which is in line with the results achieved by Jozefowicz et al. (2015). The next step to this research would be the application of these models to resource constrained languages, such as those of the *Nguni* family, of which isiXhosa and Zulu are members.

A main point of criticism to this research is that the models were not tested on a wider variety of text corpora. It is safe to assume that, by testing our models only on Kafka's work, we have allowed them to perform better than they would on totally unrelated text. This is due to the fact that the language of this one author does not deviate as drastically across his works as, for example, a collection of different works spanning across the century. Additionally, in future work, optimization of the computational design could allow a more efficient hyper-parameter search through reduced training times. For example, using Theano to directly write the neural networks would leave you with more control over your models. Also, tuning parameters such as the dropout layer to adjust for over-fitting could improve the results.

References

[1] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.

[2] P. F. Brown, P.V. Desouza, R. L. Mercer, V.J. D. Pietra, and J.C Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

[3] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393, 1999.

[4] F. Chollet. Keras. urlhttps://github.com/fchollet/keras, 2015.

[5] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

[6] Erico Guizzo. How googles self driving car works. *IEEE Spectrum Online, October*, 18, 2011.

[7] T. Hain. Language modelling and search. In *HLT Winter School*, 2016.

[8] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. Book in preparation for MIT Press, 2016.

[9] A.K. Jain, J. Mao, and K.M. Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 3:31–44, 1996.

[10] R. Jozefowicz, W. Zaremba, and I. Sutskever. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2342–2350, 2015.

[11] S Maind and P. Wankar. Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(1):96–100, 2014.

[12] M.L. Mauldin. Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. In *AAAI*, volume 94, pages 16–21, 1994.

[13] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[14] K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[15] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063*, 2012.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[17] M. Sundermeyer, R. Schlüter, and H. Ney. Lstm neural networks for language modeling. In *Interspeech*, pages 194–197, 2012.

[18] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[19] M.R. Zadeh, S. Amin, D. Khalili, and V.P. Singh. Daily outflow prediction by multi layer perceptron with logistic sigmoid and tangent sigmoid activation functions. *Water Resources Management*, 24(11):2673–2688, 2010.

Appendix

Below is the Python code used for the experiments in this research paper.

```
# -*- coding: utf-8 -*-
"""
Created on Fri Jul 08 22:50:00 2016

@author: Cedric Oeldorf
"""
from __future__ import print_function

MAX_CHARACTERS_FROM_TEXT = 360000

SYMBOLS = '{}()[].,:;+-*/&|<>=~$'
ENCODING = 'UTF-8-SIG'
MAX_VOCABULARY_SIZE = 15000
HIDDEN_SIZE = 512
MAX_SEQ_LEN = 50 # sentences with more tokens than MAX_SEQ_LEN are filtered
BATCH_SIZE = 52

from keras.models import Sequential
from keras.layers import Embedding
from keras.layers.core import Dense, Activation, Dropout, TimeDistributedDense
from keras.layers.recurrent import GRU, LSTM
from keras.preprocessing import sequence
from keras.utils.np_utils import to_categorical
import nltk
import numpy as np
import sys
import codecs
from nltk.tokenize import sent_tokenize, word_tokenize
import pandas as pd
# taken from: https://github.com/fchollet/keras/blob/master/keras/datasets/imdb.py
# and modified
def convert_sequences(sequences, max_nb_words=None,
                      maxlen=None, seed=113):

    start_char=1
    end_char=2
    oov_char=3

    if maxlen:
        new_sequences = []
        for s in sequences:
            if len(s) + 2 < maxlen:
                new_sequences.append(s)
        sequences = new_sequences
    if not sequences:
        raise Exception('After filtering for sequences shorter than maxlen=' +
                        str(maxlen) + ', no sequence was kept. '
                        'Increase maxlen.')
```

```
    print(len(sequences), "sentences remanied after filtering by MAX_SEQ_LEN")

    # Count the word frequencies
    word_freq = nltk.FreqDist([w for s in sequences for w in s])
    uniq_words = len(word_freq.items())
    print("Found %d unique tokens." % uniq_words)

    max_nb_words = min(max_nb_words, uniq_words+4)
    print("Using vocabulary size %d." % max_nb_words)

    # Get the most common words and build index_to_word and word_to_index vectors
    vocabulary = [u"<NULL>", u"<START>", u"<END>", u"<UNK>"] +
    [w for w, c in word_freq.most_common(max_nb_words-4)] # keep 4 slots for:
    padding=0, start=1, end=2 and oov=3
    word_indices = dict((w, i) for i, w in enumerate(vocabulary))
    indices_word = dict((i, w) for i, w in enumerate(vocabulary))

    # Convert words to indices and pad with start-end
    X = [[start_char] + [word_indices.get(w, oov_char) for w in s] +
        [end_char] for s in sequences]
    # create subsequences
    X = [x[:i] for x in X for i in range(1,len(x)+1)]

    np.random.seed(seed)
    np.random.shuffle(X)

    y = [x[-1] for x in X]
    X = [x[:-1] for x in X]

    X = sequence.pad_sequences(X, maxlen=maxlen)
    y = to_categorical(y, len(word_indices))

    return X, y, word_indices, indices_word


# 1. Import text and tokenize into sentences
path = "C:/Users/Cedric Oeldorf/Desktop/University/Research/Data/
Gutenberg/kafka.txt"
with codecs.open(path, 'r', ENCODING) as f:
    if MAX_CHARACTERS_FROM_TEXT:
        text = f.read()[:MAX_CHARACTERS_FROM_TEXT].lower()
    else:
        text = f.read().lower()
sent_tokenize_list = sent_tokenize(text)
print('Number of characters:', len(text))
print('Number of sentences:', len(sent_tokenize_list))
print('First sentence:', sent_tokenize_list[0].encode(ENCODING))
del text

# 2. Clean sentences of surrounding symbols and tokenize into lists of tokens
tokens = [word_tokenize(sentence.strip(SYMBOLS)) for sentence in sent_tokenize_list]
print("First sentence tokens:", tokens[0])
```

```
# 3. Convert to inputs X and labels Y, and pad with start-end tokens,
#and replace rare words with UNK
print('Converting data...')
X, y, word_indices, indices_word = convert_sequences(tokens, max_nb_words=MAX_VOCABULARY_S
maxlen=MAX_SEQ_LEN)
vocab_size = len(word_indices)
print('Got %d sequences' % len(X))
"""
print('Build model...')
model = Sequential()
model.add(Embedding(vocab_size, HIDDEN_SIZE, input_length=MAX_SEQ_LEN, mask_zero=True))
model.add(GRU(HIDDEN_SIZE/2,return_sequences=True))
model.add(Dropout(0.2))
model.add(GRU(HIDDEN_SIZE/2,return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(vocab_size, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='rmsprop')
"""
print('Build model...')
model = Sequential()
model.add(Embedding(vocab_size, HIDDEN_SIZE, input_length=MAX_SEQ_LEN, mask_zero=True))
model.add(LSTM(HIDDEN_SIZE))
model.add(Dropout(0.2))
model.add(Dense(vocab_size, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='rmsprop')

def sample(a, temperature=1.0):
    # helper function to sample an index from a probability array
    a = np.log(a) / temperature
    a = np.exp(a) / np.sum(np.exp(a))
    return np.random.choice(len(a), p=a)
    #return np.argmax(np.random.multinomial(1, a, 1))

model.load_weights('C:/Users/Cedric Oeldorf/Desktop/University/Research/Code/
                    MODELS/GRU_final_final_final.h5')

# train the model, output generated text after each iteration
from keras.callbacks import History
hist = History()
h = []
for iteration in range(1, 50):
    print()
    print('-' * 50)
    print('Iteration', iteration)

    model.fit(X, y, batch_size=BATCH_SIZE, nb_epoch=1, callbacks=[hist],
              validation_split=0.1, show_accuracy=True)
    model.save_weights('C:/Users/Cedric Oeldorf/Desktop/University/Research/
                        Code/MODELS/GRU_24June_bigmod.h5',overwrite=True)
    h.append(hist.history)
    for diversity in [1.0, 1.2]:
```

```
        print()
        print('----- diversity:', diversity)

        generated = [word_indices["<NULL>"]] * (MAX_SEQ_LEN - 1) +
        [word_indices["<START>"]]

        for i in range(MAX_SEQ_LEN):

            preds = model.predict(np.array([generated]), verbose=0)[0]
            next_index = sample(preds, diversity)
            next_word = indices_word[next_index]

            generated.append(next_index)
            generated = generated[-MAX_SEQ_LEN:]

            sys.stdout.write(next_word + " ")
            sys.stdout.flush()

            if next_word == "<END>":
                break
    print()

def save_loss(filename):
    list0 = []
    list0 = [f['loss'] for f in h]
    list2 = []
    list2 = [l[0] for l in list0]
    list0 = []
    list0 = [f['val_loss'] for f in h]
    list3 = []
    list3 = [l[0] for l in list0]
    import csv

    with open("C://Users//Cedric Oeldorf//Desktop//University//Research//RESULTS//
GRU//300k//loss_model2.csv", 'wb') as myfile:
        wr = csv.writer(myfile)
        wr.writerow(list2)
    with open("C://Users//Cedric Oeldorf//Desktop//University//Research//
RESULTS//GRU//300k//loss_val_model2.csv", 'wb') as myfile:
        wr = csv.writer(myfile)
        wr.writerow(list3)
save_loss("C://Users//Cedric Oeldorf//Desktop//University//sResearch//
RESULTS//GRU//300k//loss_withval_512bigbatch.csv")


path_loss = "C://Users//Cedric Oeldorf//Desktop//University//Research//
                RESULTS//LSTM//300k//L1.csv"
path_val_loss = "C://Users//Cedric Oeldorf//Desktop//University//
                Research//RESULTS//LSTM//300k//L1_val.csv"

import pandas as pd
loss = pd.DataFrame()
import csv
```

```
with open(path_loss, 'rb') as f:
    reader = csv.reader(f)
    your_list = list(reader)

lstm = [item for sublist in your_list for item in sublist]
loss["loss"] = lstm
with open(path_val_loss, 'rb') as f:
    reader = csv.reader(f)
    your_list = list(reader)
val = [item for sublist in your_list for item in sublist]

loss["val"] = val

#model.save_weights('C:/Users/Cedric Oeldorf/Desktop/University/Research/
                                    Code/MODELS/GRU_19June.h5')
#from keras.utils.visualize_util import plot
#plot(model, to_file='C:/Users/Cedric Oeldorf/Desktop/University/
                Research/Results/LSTM/kafka/LSTMmodel1.png')

#*****************************************************************************

# PLOTTING

#*****************************************************************************

import matplotlib.pyplot as plt
import numpy as np

fig, ax = plt.subplots(1, 1, figsize=(12, 14))

ax.spines['top'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.get_xaxis().tick_bottom()
ax.get_yaxis().tick_left()
k = len(loss)
#range for whole, np arrange for decimal
plt.xticks(range(0, k, 1), fontsize=14)
plt.yticks(np.arange(0, 7, 0.2))

for y in np.arange(0, 7, 0.2):
    plt.plot(range(0,k), [y] * len(range(0,k)), '--',
             lw=0.5, color='black', alpha=0.3)
plt.tick_params(axis='both', which='both', bottom='on', top='off',
                labelbottom='on', left='off', right='off', labelleft='on',labelsize=30)

plt.plot(loss['loss'],color='b',linewidth=4)
plt.axvline(x=12,color='g',ls='dashed',linewidth=10)

plt.plot(loss['val'],color='r',linewidth=4)
#plt.text(1, 0.8, 'LSTM', fontsize=14, color='b')
#plt.text(1, 0.8, 'GRU', fontsize=14, color='b')
```

```
import matplotlib.patches as mpatches

blue_patch = mpatches.Patch(color='b', label='Training Loss')
red_patch = mpatches.Patch(color='r', label='Validation Loss')
plt.legend(bbox_to_anchor=(1, 1),handles=[blue_patch,red_patch],prop={'size':35})
#plt.title("Minimization of cross entropy loss (GRU model 2)")
#plt.xlabel("Iteration")
#plt.ylabel("Cross entropy loss")
ax.set_title("Minimization of cross entropy loss (LSTM model 1)",fontsize=42)
ax.set_xlabel("Iteration",fontsize=42)
ax.set_ylabel("Cross entropy loss",fontsize=42)
#*********************************************************************

#Calculate perplexity

#*********************************************************************
loss = loss.convert_objects(convert_numeric=True)

loss["perp_lstm"] = 2**(loss["val"])
loss["perp_gru"] = 2**(loss["gru"])

"""
T-test
"""

stats.ttest_ind_from_stats(85.97135, 0.187188, 10, 77.910128,0.0442985,
                           10, equal_var=True)
```

Using hazard functions for modeling arc lengths

Unathi Oliphant 12031578

WST795 Research Report

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

Supervisor: Mr MT Loots, Co-supervisor: Miss S Makgai

Department of Statistics, University of Pretoria

02 November 2016 (Final)

Abstract

The arc length per unit decreases along the density curve of a random variable, and then increase again after reaching the mode. This is typically the behaviour of a hazard function. This report focusses on finding out if hazard functions can be used to model arc lengths.

Declaration

I, *Unathi Oliphant,* 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.

_____
Unathi Oliphant

_____
Mr MT Loots

_____
Miss S Makgai

_____
Date

Acknowledgements

I would like to acknowledge my supervisors Mr MT Loots and Miss S Makgai for their guidance and support during the compilation of this research report.

I would also like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a post graduate bursary

Contents

1 Introduction **6**

2 Background Theory **6**
2.1 The Weibull distribution 7
2.2 The Exponential distribution 8

3 The Cox PH model **8**
3.1 Background Theory 8
3.2 Application 9
3.2.1 Probability density function, $f(t)$ 9
3.2.2 Model fit 9

4 Accelerated Failure Time model **10**
4.1 Background Theory 10
4.2 Application 11
4.2.1 Model fit 11
4.2.2 Hazard function, $h(t)$ 11

5 Probability model **12**
5.1 Background Theory 12

6 Application **12**
6.1 Cox PH model 13
6.2 Accelerated Failure Time Model 13

7 Conclusion **14**

Appendix **16**

List of Figures

1 PDF and histogram of survival times 9
2 Hazard function for heart failure 11
3 Arc lengths of standard normal PDF 13
4 Survival function of arc lengths of standard normal PDF 14

List of Tables

1 Cox PH Model fit statistics 10
2 AFT Model fit statistics 11
3 Cox PH model fit statistics 13
4 AFT model fit statistics 13

1 Introduction

In simple terms, an arc length is the distance along a curved line making up an arc. The determination of an arc length is called rectification of a curve. A curve can be approximated by connecting a number of points on the curve using line segments to create a path along the curve. The total length is then approximated by summing the lengths of each of the linear segments [13]. Rectifiable curves i.e. curves with finite length, are defined to have a smallest value L, that acts as an upper bound on the length of any polygonal approximation. The number L is defined as the arc length and is calculated using the formula:

$$\int_a^b \sqrt{1 + [f'(x)]^2} \tag{1}$$

for any finite interval$[a, b]$. This report aims at modeling an arc length using hazard functions since these functions display a similar behaviour. A hazard function, $h(t)$ is defined as the ratio between the probability density function and survival function, $f(t)$ and $S(t)$ respectively. The function $h(t)$ is defined is defined by the equation:

$$h(t) = \frac{f(t)}{S(t)} \tag{2}$$

where $S(t) = 1 - F(t)$, and $F(t)$ represents the cumulative distribution function [4]. The hazard function is also known as the rate of failure since it is the probability of failure within an almost instantaneous period of time given the survival of a subject until time t [5]. Therefore a hazard function can be taken as a measure of risk; the higher the hazard function in a specific time period, the greater the failure in that time period. This report is structured as follows: Section 2 gives background theory and derivations of hazard functions. Section 3 presents theory and application of the Cox PH, Section 4 and 5 presents an overview of the Accelerated Failure Time model and Probability model respectively. This will be followed by a practical example comparing the Cox PH model and Accelerated Failure Time model and concluding remarks in Sections 6 and 7.

2 Background Theory

Consider the definition as given in [4], let T be the continuous survival time, and $f(t)$ the probability density function of T. The CDF of T is therefore given by:

$$F(t) = P(T \leq t) = \int_0^t f(s)ds,$$

Thus, $F(t)$ denotes the probability of failure by time t.
The survival function is defined as:

$$S(t) = P(T > t) = 1 - F(t).$$

Therefore $S(t)$ is the probability of survival beyond time t.
The hazard function is defined as:

$$h(t) = \frac{f(t)}{S(t)}.$$

$h(t)$ can be thought of as the probability of failure in a very small time period i.e. between t and $t + \triangle t$ given that the individual has survived until time t.
By definition using first principle:

$$f(t) = \lim_{\triangle t \to 0} \frac{F(t + \triangle t) - F(t)}{\triangle t}.$$

Therefore from [9], the hazard function can be expressed as:

$$\begin{aligned} h(t) &= \lim_{\triangle t \to 0} \frac{F(t+\triangle t) - F(t)}{\triangle t \cdot S(t)} \\ &= \lim_{\triangle t \to 0} \frac{P(t < T \leq t+\triangle t)}{\triangle t \cdot S(t)} \\ &= \lim_{\triangle t \to 0} \frac{F(t < T \leq t+\triangle t | T > t)}{\triangle t}. \end{aligned}$$

To determine the relationship between the hazard and survivor functions [7]. We have:

$$\begin{aligned} h(t) &= \frac{f(t)}{S(t)} \\ &= \frac{f(t)}{1-F(t)} \\ &= -\frac{d}{dt} log[1-F(t)] \\ &= -\frac{d}{dt} log[S(t)]. \end{aligned}$$

Therefore,

$$S(t) = exp[-H(t)],$$

where

$$H(t) = \int_0^t h(s)ds.$$

$H(t)$ is defined as the cumulative hazard function. Similar to the hazard function, $H(t)$ is a risk measure. Large values of $H(t)$ correspond to an increased risk of failure by time t.

Consider the case when T is a discrete random variable,[11] the probability mass function is given by $P(T = t_i) = f(t_i)$, $i = 1, 2, ...$ The survival function is then given by:

$$\begin{aligned} S(t) &= \sum_{j|t_j \geq t} f(t_j) \\ &= \sum_{j|t_j \geq t} f(t_j) I_{(t_j \geq t),} \end{aligned}$$

and the indicator function is defined as:

$$I_{(t_j \geq t)} = \begin{cases} 0 & if \quad t_j < t \\ 1 & if \quad t_j \geq t \end{cases}.$$

Here the hazard function is defined as the conditional probability of failure at time t_j conditioned on the fact that the individuals survived up to time t_j,

$h_j = h(t_j) = P(T = t_j | T \geq t_j) = \frac{f(t_j)}{S(t_j)} = \frac{S(t_j) - S(t_{j+1})}{S(t_j)} = 1 - \frac{S(t_{j+1})}{S(t_j)}$ [7].

2.1 The Weibull distribution

The density function of a Weibull with parameters λ and p, is

$$f(t) = p\lambda^p t^{p-1} \exp[-(\lambda t)^p].$$

for $\lambda > 0$, $p > 0$ and $t \geq 0$. If the survival times follow a Weibull distribution with these parameters, the survival function is given by

$$S(t) = \exp[-(\lambda t)^p].$$

Therefore, using equation (2) the hazard function as in [12]will be:

$$h(t) = \frac{f(t)}{S(t)}$$

$$= p\lambda^p t^{p-1},$$

with $\lambda, p > 0$. The Weibull hazard function can be increasing or decreasing depending on whether the value of p is less than or greater than 1. Values of $p > 1$ result in an increase of the hazard function while $p < 1$ results in a decrease. As specified above the value $p = 1$ reduces to the special case of the exponential distribution since the hazard function remains constant [11].

2.2 The Exponential distribution

An Exponential distribution with the parameter λ has density function:

$$f(t) = \lambda \exp(-\lambda t)$$

with $\lambda > 0$. This is the special case of the Weibull hazard function where $p = 1$. The survival function is given as

$$S(t) = 1 - F(t) = \exp(-\lambda t).$$

Therefore, using equation (2) the hazard function will be:

$$h(t) = \frac{f(t)}{S(t)}$$

$$= \lambda.$$

The hazard function for the exponential distribution is a constant with respect to time [11]. This makes sense since the exponential distribution has the memory-less property. This also means that the probability of failure at any given time interval does not depend on what has happened before time t.

3 The Cox PH model

3.1 Background Theory

The general formula for the proportional hazard function is given by:

$$h(t, x) = h_0(t)g(x, \beta),$$

where $g(x, \beta)$ is a function of the vector of covariates x and the unknown parameter β [1]. The function $h_0(t)$ is called the baseline hazard and is dependent on t. The baseline function should be estimated when applying the Cox PH model. An assumption of the model is that the covariates act multiplicatively on the hazard rate, it also assumed as a consequence that the hazard rates of various individuals should be proportional to each other. The Cox proportional hazard model (Cox PH model) proposed by Sir David Cox is given by the formula:

$$h(t, x) = h_0(t) \exp(x, \beta),$$

where the function $h_0(t)$ is a totally unspecified baseline function [1]. The Cox PH model is a multivariate regression semi-parametric model. The model supports the modeling of continuous covariates, and includes the assumption of proportional hazards amoungst different groups. Arc length data will be fit to the Cox PH model to see whether this model can successfully predict the behaviour of an arc length.

3.2 Application

To demonstrate the standard application of hazard functions we are going to consider an example from the Worcester Heart Attack Study [3]. The study examines factors that can have an effect on the time that an individual survives after a heart attack. The same data set used in this Section will also be used8 in Section 4.

3.2.1 Probability density function, $f(t)$

Let T be a random variable that records the time of survival. The likelihood of observing T at a time t relative to all other times of survival is the function $f(t)$. To get the survival time within an interval we integrate the PDF over the range of survival times. Figure 1 is the PDF of the survival times for the data set from the Worcester Heart Attack Study [3].

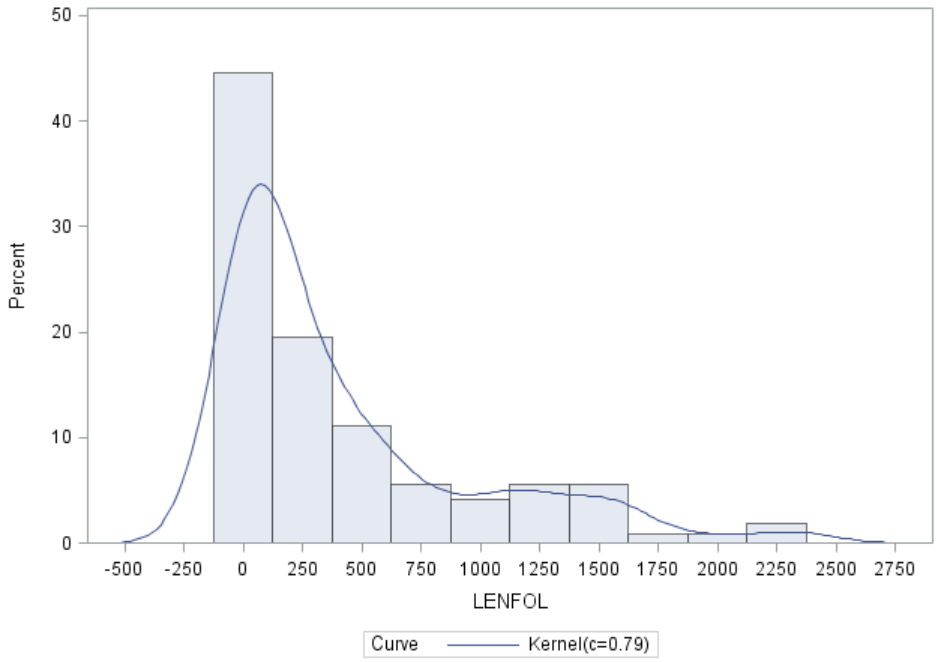


Figure 1: PDF and histogram of survival times

In both the histogram and PDF we see that for this data, shorter survival times are more likely observed, which means that there is a higher risk of heart attack initially but it decreases rapidly as time goes on. (Note that since there is no time that is less than zero, hence on the left of LENFOL=0 there should be no graph)

3.2.2 Model fit

A simple Cox regression model is fit using PROC PHREG in SAS. We will use the Log-likelihood criterion (-2LogL), Akaike Information Criterion (AIC) and the Schwarz Bayesian Criterion (SBC)/BIC in Table 1 to determine if this model fits data better compared to the AFT model (Table 2) .

Criterion	Without Covariates	With Covariates
-2 LOG L	2455.158	2313.140
AIC	2455.158	2317.140
SBC	2455.158	2323.882

Table 1: Cox PH Model fit statistics

4 Accelerated Failure Time model

4.1 Background Theory

The Accelerated Failure Time model, also known as the AFT model is a parametric model that is mostly used as a substitute for proportional hazards models (PH models). The difference between an AFT model and a PH model is that the PH model makes the assumption that the covariates have a multiplicative effect on the hazard rate and the AFT model makes the assumption that the covariates have the effect to either decelerate or to accelerate the hazard rate [2].
The general formula for the AFT model is given by:

$$h(t, \theta) = \theta h_0(\theta t),$$

where θ is the joint effect of the covariates, typically expressed as:

$$\theta = \exp(-[\beta_1 X_1 + \cdots + \beta_p X_p]).$$

It should be noted that the negative sign in the expression indicates that the survival time is increased by high values of the covariates, hence if we omit the negative sign, the increase will be on the hazard. The condition is satisfied, if the conditional PDF of this event is:

$$f(t|\theta) = \theta f_0(\theta t),$$

thus it follows that the survival function is:

$$S(t|\theta) = S_0(\theta t).$$

The AFT model that is commonly used is the log-logistic distribution [6]. This model can model a hazard function that is non-monotonic i.e. functions that increase and then decrease as time goes on. It has a shape that is similar to that of the log-normal distribution but has a simple closed form CDF. This is essential when fitting censoring data. We need the survival function for observations that are censored, which is given in section 1 as:

$$S(t|\theta) = 1 - F(t|\theta).$$

The only distributions that possess the property of being parameterized as either an AFT or PH model are the Weibull distribution and the Exponential distribution. We can therefore use either of the models to interpret the results of fitting the Weibull model. But, the practical application of the Weibull model may be restricted since it has a monotonic hazard function. There are also other distributions that are suitable for the AFT model like the gamma, inverse Gaussian and log-normal distributions, even though these distributions are not as popular as the log-logistic since they do not have closed form CDFs. Lastly, the generalized gamma distribution has three parameters and the gamma, log-normal and Weibull distributions are special cases [12].

4.2 Application

4.2.1 Model fit

The model fit statistics for the AFT model are given in Tabel 2. These results show that the AFT model fits the data better than the Cox PH model since the -2LogL, AIC and SBC/BIC are all lower for the AFT model.

-2 Log Likelihood	1348.033
AIC (smaller is better)	1356.033
AICC (smaller is better)	1356.114
BIC (smaller is better)	1372.891

Table 2: AFT Model fit statistics

4.2.2 Hazard function, $h(t)$

In Section 1, the hazard function was given by equation (2) as:

$$h(t) = \frac{f(t)}{S(t)}.$$

The Sas PROC LIFETEST procedure used to fit the model also gives a plot of the hazard function which is shown in Figure 2.

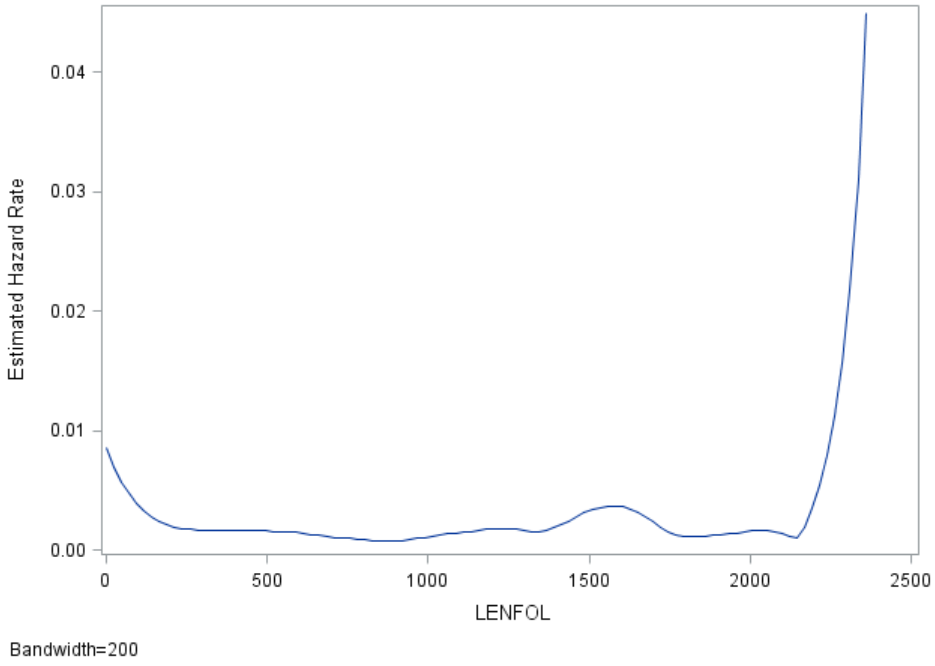


Figure 2: Hazard function for heart failure

The hazard function is higher at the beginning then it declines until reaches a point where it levels off. Thus at the start of the study we would be expecting approximately 0.008 failures a day and approximately 0.002 failure a day for the individuals that survived. Bear in mind that in this report we are more interested in the shape of the hazard function than the interoretation of the graph. The shape of this hazard function is similar to that of that the "bathtub" hazard function [8].

5 Probability model

5.1 Background Theory

Consider the modeling of an individual's risk of failure, let the path followed by the individual p be fixed, apart from its end point. An individual would then follow a potential path until censoring or failure, whichever happens first. This is corresponding to the assumption that is usually made in PH regression which states that the vector of covariates for every individual is a fixed time dependent function. The probability of survival of an individual is hence conditional on the potential path p. Now consider the arc length for each path,l. The most general model would let the integrated hazard log-survival function, say G'', be a function of p and l. Therefore

$$\log(P\{surviving\,to\,l|p\}) = -G''(l,p),$$

this allows for the definition of time scales (a,b) that can be collapsed. Suppose that the function $G'(a,b)$ exists such that, for all observations of (a,b) and all the paths p that are passing along the points(a,b). Then

$$G''(l,p) = G'(a,b),$$

where l is the arc length of p at the point (a,b). The survival probability for a specific point is hence dependent on where that point is and not dependent on the path taken to reach that point. In this sense if the we have collapsible time scales, then G' or any increasing function of G' can be viewed as univariate measures of time [10]. Although the theory of this model is presented in this paper, no further application will be considered since no implementations of the model have been presented in SAS/R thus far. However this could be an area of interest for future research.

6 Application

This section aims to model the arc length as calculated using equation (1), of a standard normal probability density function using hazard functions. This process is done as follows:

1. Calculate the quantiles of the standard normal cumulative density function in the interval $[0.001, 0.999)$. This will give us a vector Q of length 998.
2. Calculate the arc lengths for consecutive points in Q using (1). This results in a vector L of 998 arc length values.
3. Fit the data generated from 1 and 2 to the different hazard functions

Figure 3 is the plot of the arc lengths L of each interval in Q. The plot has a shape that is similar to that of a "bathtub" hazard function [8].

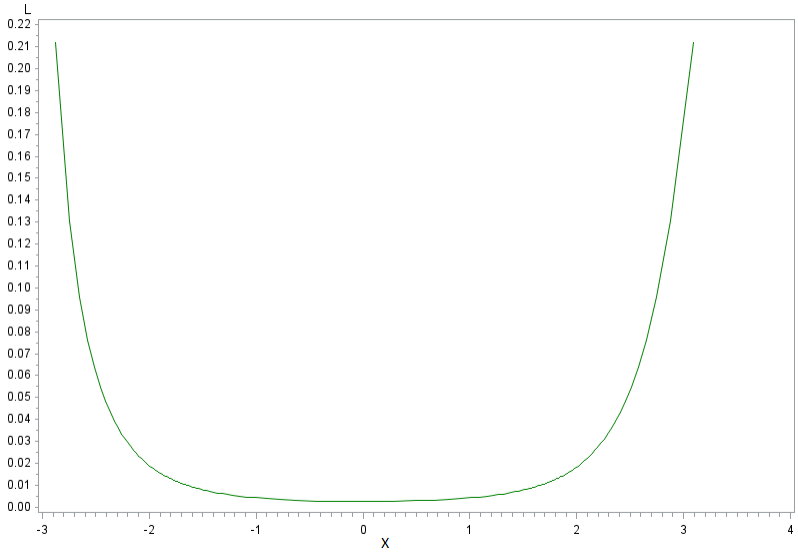


Figure 3: Arc lengths of standard normal PDF

6.1 Cox PH model

The SAS procedure PROC PHREG is used to fit the data to the Cox PH model. Table 3 shows the model fit statistics for the Cox PH model which will be compared to that of the AFT model (Table 4) to determine which model fits the data better using the -2LogL, AIC and SBC/BIC.

Criterion	Without Covariates	With Covariates
-2 LOG L	11796.627	11796.608
AIC	11796.627	11798.608
SBC	11796.627	11803.513

Table 3: Cox PH model fit statistics

6.2 Accelerated Failure Time Model

The SAS procedure PROC LIFEREG is used to fit the data to the AFT model and the output is given in the table in Table 4. The results in Table 3 and Table 4 were compared and it was found that the AFT model is fits data better than the Cox PH model since the -2LogL, AIC and SBC/BIC are much lower for the AFT model.

-2 Log Likelihood	2813.811
AIC (smaller is better)	2819.811
AICC (smaller is better)	2819.835
BIC (smaller is better)	2834.528

Table 4: AFT model fit statistics

The plot of the hazard function in Figure 5 has a similar shape to that of the data generated and plotted in Figure 4, this shows that this is a viable method of modeling arc length. Note that in this report we are interested in the shape of the function.

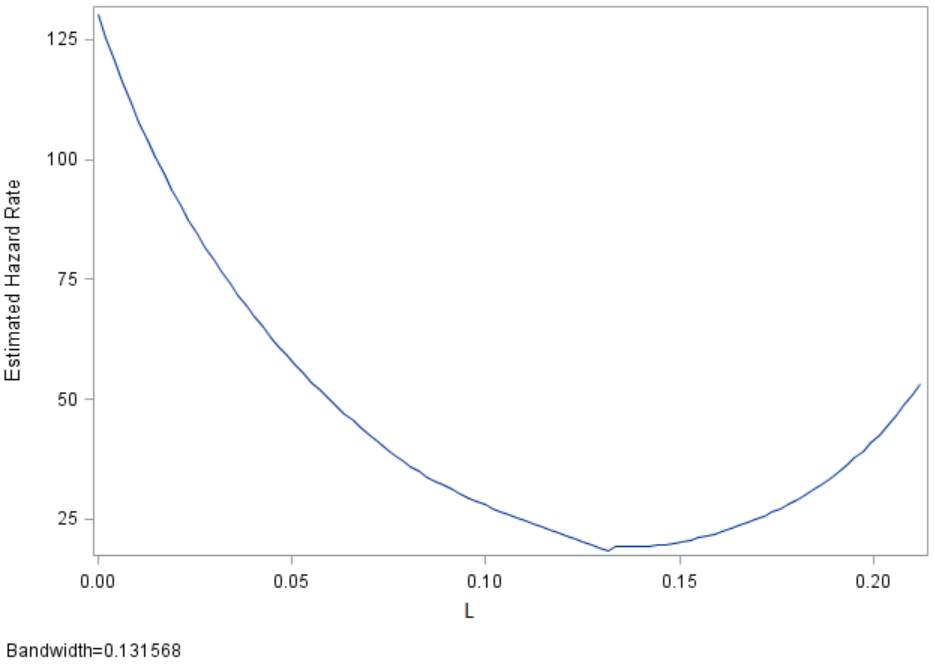


Figure 4: Survival function of arc lengths of standard normal PDF

7 Conclusion

In this report we discussed the modeling of arc lengths using hazard functions. The Cox PH model and the AFT model were used to model the arc length from the PDF of the standard normal distribution. Other distributions and other forms of arc lengths can be considered, however in this report only the standard normal distribution was considered. The data was fit to the Cox PH model and the model fit statistics were obtained. The data was also fit to the AFT model to obtain the model fit statistics and the plot of the hazard function. The plot of the hazard function in Figure 5 has a similar shape to that of the data generated and plotted in Figure 3. The results in Table 3 and Table 4 were compared and it was found that the AFT model is fits data better than the Cox PH model since the -2LogL, AIC and SBC/BIC are much lower for the AFT model. This was also true for the data used in the Worcester Heart Attack Study [3]. Even though the AFT model is better than the Cox PH model, the model fit is not satisfactory therefore further research is still required to improve the model fit. We also discussed a model called the Probability model which has not been used in application yet. This model could be used in future for modeling arc length.

References

[1] JA Anderson and A Senthilselvan. Smooth estimates for the hazard function. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 322–327, 1980.

[2] Mike J Bradburn, Taane G Clark, SB Love, and DG Altman. Survival analysis part ii: multivariate data analysis–an introduction to concepts and methods. *British Journal of Cancer*, 89(3):431–436, 2003.

[3] J Bruin. newtest: command to compute new test, February 2011. [ONLINE] http://www.ats.ucla.edu/stat/stata/ado/analysis/. Accessed: 19-05-2016.

[4] John Fox. Cox proportional-hazards regression for survival data. *An R and S-PLUS Companion to Applied Regression*, pages 1–18, 2002.

[5] MC Gacula and JJ Kubala. Statistical models for shelf life failures. *Journal of Food Science*, 40(2):404–409, 1975.

[6] Ian James. Accelerated failure-time models. *Encyclopedia of Biostatistics*, 1998.

[7] John P Klein. Survival distributions and their characteristics. *Wiley StatsRef: Statistics Reference Online*, 2005.

[8] Georgia-Ann Klutke, Peter C Kiessler, and MA Wortman. A critical look at the bathtub curve. *IEEE Transactions on Reliability*, 52(1):125–129, 2003.

[9] M McAleer, Wei-Chen Chen, and CL Chang. Survival analysis of very low birth weight infant mortality in taiwan. *Journal of Health and Medical Economics*, 2014.

[10] David Oakes. Multiple time scales in survival analysis. *Lifetime Data Analysis*, 1(1):7–18, 1995.

[11] Jiezhi Qi. Comparison of proportional hazards and accelerated failure time models. *Thesis, University of Saskatchewan, Saskatoon, Saskatchewan, Canada.*, 2009.

[12] German Rodriguez. Parametric survival models. *Lectures Notes, Princeton University*, 2005.

[13] JH Walsh, EN Nilson, and JL Walsh. The theory of splines and their applications. *Academic Press, New York*, 38(51), 1967.

Appendix

```
proc iml;
x_ =do (0.002,0.999,0.001);
Q=quantile('normal', x_);
start MyFunc(x);
f=pdf('normal', x);
f2=-x*f;
return(sqrt(1+f2##2));
finish;
free rr;
do i = 0.001 to 0.998 by 0.001;
a=quantile('normal', i);
b=quantile('normal', (i+0.001));
call quad(R, "MyFunc", a||b);
rr=rr//r;
end;
dat1=Q'||rr;
print dat1;
create plot1 from dat1[colname={'x' 'L'}];
append from dat1;
symbol1 value=none color=green i=join;
proc gplot data=plot1;
plot L*x;
run;
proc phreg data=plot1 plots=s;
model L=x;
run;
```

Machine learning ensemble: random decision forest

Philip Owen Randall 12005721

STK795 Research Report

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

Supervisor: Dr A De Waal, Co-supervisor: J Mazarura

Department of Statistics, University of Pretoria

2 November 2016

Abstract

The introduction of ensemble learning algorithms in predictive statistics remains relatively undocumented when compared to more established methods. Consideration of the linear regression modeling approach, ordinary least squares and by extension, weighted least squares, comprises a large part of the discussion.

This paper employs multiple regression models which are affected by heteroscedasticity. This data is used with the aim of drawing a comparison between the standard linear regression approach, ordinary least squares and weighted least squares, as well as an ensemble learning algorithm, the random forest regressor. This comparison deals with each methods predictive ability, the R^2value and their respective ability to detect heteroscedasticity.

Lastly, the random forest regressor makes use of certain tuning parameters. The effect of one of these parameters, the number of estimators, on the predictive ability and generalized error is studied with the aid of out-of-bag error rates.

Declaration

I, *Philip Owen Randall,* 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.

_____
Philip Owen Randall

_____
Dr Alta De Waal

_____
Date

Contents

1 Introduction **5**
- 1.1 Objectives . . . 5
- 1.2 Literature Review . . . 5
  - 1.2.1 Random Decision Forests . . . 5
    - 1.2.1.1 Decision Trees . . . 5
    - 1.2.1.2 Randomization . . . 6
  - 1.2.2 Machine Learning: A Probabilistic Approach . . . 6
  - 1.2.3 Machine Learning Benchmarks and Random Forest Regression . . . 6
  - 1.2.4 An Introduction to Statistical Learning . . . 6
  - 1.2.5 Understanding the Impact of Heteroscedasticity on the Predictive Ability of the Modern Regression Methods . . . 7
- 1.3 Terminology . . . 7
- 1.4 Research Structure . . . 7

2 Background Theory **8**
- 2.1 Ensemble Techniques - Bootstrap Aggregating . . . 8
- 2.2 Decision Trees . . . 8
- 2.3 Random Forest Regressor . . . 9
- 2.4 Heteroscedasticity . . . 11
  - 2.4.1 Basic Overview of Heteroscedasticity . . . 11

3 Application **12**
- 3.1 Introducing the Data . . . 12
- 3.2 Experimental Design . . . 13
  - 3.2.1 Comparative Ability of the Random Forest Regressor . . . 13
  - 3.2.2 Adjusting the Number of Estimators . . . 15

4 Final Remarks **16**
- 4.1 Conclusions . . . 16
- 4.2 Future Studies . . . 16

Appendix **18**

List of Figures

1 Recursive partitioning of an space . . . 8
2 Homoscedastic vs heteroscedastic residuals . . . 12
3 Comparison of residual plots for OLS,WLS, RFR. . . . 14
4 Out-of-bag errors for RFR over the three models . . . 15

List of Tables

1 Coefficient of determination (R^2) comparison between OLS, WLS and RFR . . . 13
2 Spearman's rank coefficient correlation test comparison between OLS, WLS, RFR . . . 13
3 Effect of the number of estimates in RFR. . . . 15

1 Introduction

Ensemble learning methods employ multiple learning algorithms in an attempt to obtain improved predictive performance that exists in any one given algorithm. Machine learning ensembles only make use of a finite set of alternative models unlike their counterpart, statistical ensembles, which are typically an infinite set [10]. However, the use of a finite set of alternative models allows machine learning ensemble techniques a certain flexibility in their structure.

Due to the ease of fitting linearly regression models, as opposed to models which are non-linearly related, the linear regression model was the first type of regression analysis to be both studied punctiliously and used exhaustively in practice[12]. With the aim to determine the ability of modern machine learning techniques as suitable methods in statistical computation, it is an obvious choice to first make use of linear regression to act as the comparison.

1.1 Objectives

This research serves to better understand the performance of the random forest regressor in the presence of heteroscedasticity and to determine the possible effects changes to data as well as to specific tuning elements which the model exerts on the overall credibility of the model's predictive ability.
Ultimately the objective of the research is to:

- Comprehensively understand the foundation of the random forest model, taking special note of the models predictive ability with regard to the addition of additional explanatory variables and adjustment of the number of estimators (trees) the model may make use of in the forest.
- Compare the predictive ability of the ordinary least squares, weighted least squares and random forest models.
- Determine the random forest models capability in identifying heteroscedasticity in comparison to the ordinary least squares and weighted least squares models.

1.2 Literature Review

1.2.1 Random Decision Forests

Author Tin Kam Ho[5], the creator of the random decision forest using the random subspace method provides insight into the method and fully explains the basic principles that the method is founded on. His unique view of decision trees, taking into account various tree growing methods, the optimization of those trees and the creation of multiple trees allows for an elementary introduction to the topic.

The paper can be portioned into two primary sections:

1.2.1.1 Decision Trees Binary decision trees make use of a single feature at each non-terminal (decision) node. Oblique decision trees are similar to the binary decision trees, but differ on the use of hyper planes within the feature space. Oblique decision trees have been studied extensively over the course of the past two decades; various decision tree classifiers have been used, primarily due to their ease of use and fast classification, such as the Hidden Markov model (HMM) and multi-layer perceptions[5]. These oblique decision trees exist in both regression and classification. Furthermore, two distinct methods of tree growing and by extension pruning, are discussed with the aim of highlighting their relative strengths and uses. The central axis projection aims at separating at least two classes at each non-terminal node. Perception training differs by using a fixed-increment perception training algorithm to choose the hyperplane at each of the non-terminal nodes[5].

The pitfalls of decision trees, namely the bias of a single classifier, are addressed by making use of multiple decision trees. Multiple trees are just another stepping stone to reaching a random forest. It is the first time the reader starts to get a glimpse of the end results through the creation of the forest.

1.2.1.2 Randomization Randomization is a method used in statistics to great affect and is certainly a powerful tool for introducing differences in multiple classifiers[5]. Previously it has been used to initialize training algorithms with different configurations which would in time yield different classifiers. Though the use of randomization in selecting components acts primarily as a convenient way to explore possibilities.

Further insight will be added to this process within random forests once the idea of bootstrap aggregating, an ensemble technique is fully introduced[5].

1.2.2 Machine Learning: A Probabilistic Approach

Where Tin Kam Ho[5] in Random Decision Forests gives an insight to the basic practice of decision trees, bagging and randomization when creating a random forest, Kevin P. Murphy [9] goes into more detail of similar principles that surround the topic of random forests. This helps create more confidence in the random forest approach through understanding the subtle differences in the components of the model.

More specifically, the concepts behind growing a tree and pruning it are explained for single trees as well as multiple trees. Growing trees makes use of a rather greedy procedure which causes overfitting, a typical drawback to single decision trees when not pruned, is avoided to some extent by averaging the regressors when creating multiple trees[9].

Lastly, some light is shed on the effect these compounding methods have on more general descriptive statistics, the notable takeaway being the reduction in variance, point what is confirmed later by Mark R. Segal in Machine Learning Benchmarks and Random Forest Regression[11].

1.2.3 Machine Learning Benchmarks and Random Forest Regression

The use of Machine Learning Benchmarks and Random Forest Regression[11] serves primarily as a mathematical description of the random forest model. Definitions are given for all the relevant equations which are used, given that certain assumptions are met, to draw base conclusions about bias, variance and correlation between the predictors. Knowing the results in theoretical terms allows for a strategy to be employed to achieve those results empirically. These results are what typically distinguish forests from black-box predictors (e.g. neural nets)[11].

Furthermore, confirmation is gained about previous accumulations of the non-factor of bias, shifting the emphasis to lowering the variance for the regressor. This shift inflates the importance of prediction error results by means of the out-of-bag estimates[11].

1.2.4 An Introduction to Statistical Learning

An introduction to basic ensemble techniques is critical and is explained in great detail in An Introduction to Statistical Learning[6]. The idea of bootstrap aggregating, or bagging, which is often referred to in other literature, forms the basis for many of the results expected to be observed by the random forest classifier. This is a part truth as the random forest classifier does not use bagging by the general definition, but makes slight adjustments to meet its own end.

Bootstrapping takes a standard data set and generates a number of new data sets by means of random resampling. The new inputs then fit the models by averaging themselves. Bagging essentially builds a number of decision trees based on the bootstrapped sample, making use of all of the predictor variables available. In doing so, the meta-algorithm leads to increased accuracy and stability.

Random forests use much the same technique, but do not use all of the predictors in the creation of the decision trees, rather it limits the number of predictors to be used. Further discussion in Section 2 will better describe the reasoning for such a tactic and the outcome thereof.

1.2.5 Understanding the Impact of Heteroscedasticity on the Predictive Ability of the Modern Regression Methods

Heteroscadisticity is the violation of the assumption of homoscedasticity - a constant variance in response which is an explicit assumption when using linear regression and thus an implicit assumption with other predictive tools[4]. The effect is widely documented across a variety of statistical techniques, such as ordinary least squares[4] and must be taken seriously when applied to random forests.

While modern regression methods are growing in sophistication, there seems to be a lack in the developed literature describing the effects of heteroscadasticity on these methods [4]. This research deals primarily with identifying heteroscadisticity, while quantifying and measuring the effect on predictive tools allowing for conclusions to be made about the robustness of a model. While many regression tools are identified in this paper the primary concern will be on random forests with some comparisons being drawn to regression trees and boosted regression tools.

1.3 Terminology

We define the following terms:

- Decision trees are a predictive model which maps observations of an item to their conclusion. It makes use of leaves, representing class labels, as well as branches and conjunctions of features that lead to class labels.
- Random forest, or random decision forest, refers to an ensemble learning method for regression and classification. Random forests are constructed with a multitude of decision trees that make use of randomization to correct the over-fitting of decision trees.
- Robust statistics models are those that provide good performance for data drawn from a variety of probability distributions, especially non-normal distributions.
- Bagging, primarily used for the reduction of bias and variance in supervised learning, is a machine learning ensemble meta-algorithm.
- Heteroscedasticity occurs in the absence of homoscedasticity and is characterized by the variability of certain sub-populations in comparison to others.

1.4 Research Structure

- Part I serves primarily as a reference to the reader; with the intention to serve as an introduction to the key concepts that will be discussed in the paper.
- Part II serves primarily to solidify the key theoretical concepts of the random decision forest ensemble techniques, as well as other techniques used by the method.
- Part III is the core of the research, which involves the evaluation of the random forest method on three generated data sets. Comparisons will be drawn between the random forest model and the linear regression models, OLS and WLS, in the accuracy of the predictions and their capability in identifying heteroscedasticity in the population.
- Part IV will discuss the results obtained in Part III, drawing conclusions about the study and proposing possible future study.

2 Background Theory

2.1 Ensemble Techniques - Bootstrap Aggregating

Bootstrap aggregating, or bagging, is a machine learning ensemble based on the sampling technique, bootstrapping. Bagging makes use of a standard data set and generates a number of new data sets by uniformly sampling with replacement from the standard sample. The new data sets are later combined by averaging results of the regression. Bagging is used as it leads to "improvements for unstable procedures" [8]. Typical improvements by using this method are a reduction in variance and assisting in avoiding over-fitting of the model. Both the reduction in variance and the safeguard against over-fitting are results achieved when the n samples are averaged[8].

2.2 Decision Trees

A classification and regression tree (CART), not to be confused with decision theory, is created through recursively partitioning the input space, then defining a local model at each division within the input space[9]. The outcome of this parallel shift is the splitting of a two dimension space into M regions, each with a mean response and therefore a piece-wise constant surface. Figure 1 shows the creation of these spaces with their mean responses[9].

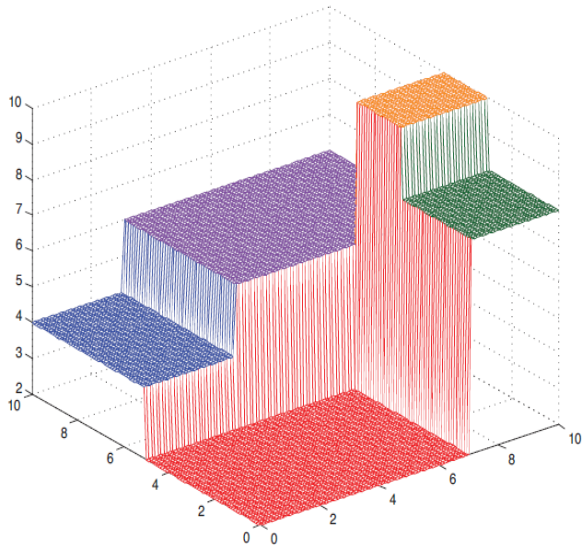


Figure 1: Recursive partitioning of the input space and defining a local model at each space. Each surface indicates one of M regions with an individual mean response[9].

The model can be expressed mathematically as:

$$f(\mathbf{x}) = E[\mathrm{y}|\mathbf{x}] = \sum_{\mathrm{m}=1}^{\mathrm{M}} \mathrm{w_m} \mathbb{I}(\mathbf{x} \epsilon \mathrm{R_m}) \tag{1}$$

Where $R_m$is the m^{th}region and $\mathrm{w_m}$is the mean response in this region.

$$\Longrightarrow \sum_{m=1}^{M} \mathrm{w_m} \phi(\mathbf{x};\mathbf{v_m}) \tag{2}$$

$\mathbf{v}_{\mathrm{m}}$ encodes the choice of the variable that will create the split and the threshold value on the path from the root to the m^{th} leaf. CART is an adaptive basis function with the basis function defining the region and the weights defining the response variable in each region[11].

In computational complexity theory a decision problem is considered NP-complete, where NP refers to nondeterministic polynomial time, when it is both NP and NP-hard[2]. Essentially this entails that the time required to solve a known algorithm increases rapidly as the size of the problem grows. A way to circumvent an NP-complete problem, which is often difficult to identify, is to make use of heuristic methods or approximation algorithms[2]. When growing a tree the optimal partitioning is NP-complete, therefore there is need a to use a split function, which chooses the best function and the best value for the feature[11]. The cost of using the split, however, is the greed of the function. This greed will become more relevant at a later stage.

The prevention of over-fitting the model is another concern when using decision trees. To avoid over-fitting, the tree growth needs to be halted at some stage. Naturally, tree growth is ceased if the decrease in error is not justified by the addition of another subtree. This is a rather myopic approach because if each feature has little predictive power, individually no splits would occur[9]. A better approach would be to grow a full tree and prune it. The layman's thinking conclude that if pruning the branch decreases the error then proceed. Determining just how far back to prune is established by evaluating the cross-validation error on each specific subtree[9].

The discussion of the CART models allows for certain conclusions to be drawn about their use. While they are considered easy to interpret, have an automatic variable selection and are robust to outliers in the data, there are also very concerning disadvantages. CART models tend to be inaccurate in prediction, when compared to alternatives, due to their greedy nature[9]. Furthermore, trees are high variance estimators as they are unstable to small changes to the input data. These implications have some effect in random forests, where CARTS are used[9].

2.3 Random Forest Regressor

Binary decision trees use a feature at each non-terminal (decision) node[5]. Geometrically, this can be visualized by assigning a point to one side of a hyperplane, which is parallel to one axis of the feature space. Oblique decision trees follow a similar principle. The only difference being that the hyperplanes are not necessarily parallel to an axis of the feature space[5]. The result is smaller decision trees that fully split the data into leaves containing a single class[5].

There exist two primary methods to growing trees which use the above concept of oblique decision trees, central axis projection and perception training[5]. Neither will be discussed further but the results that both yield are of importance. Both methods are capable of growing complex trees that fully classify the training data. Due to the bias in which hyperplanes are chosen, both methods tend to have poor generalization accuracy[9].

To extenuate the bias in which hyperplanes are chosen, multiple classifiers can be used[5]. This makes use of multiple trees, a *forest*. This method is only successful when the trees are generalized independently. Randomization will be used to achieve this effect. To inject randomness into the trees this paper will focus on one form of bootstrap aggregating (bagging), a form of random training set sampling[5]. The method which random forests apply is an improvement over standard bagging. In standard bagging, a number of decision trees are built on bootstrapped training data but in random forests, there are changes to each split. If there are p total predictors then m predictors are chosen at random from p, typically the number of predictors chosen is a function of the total predictors, $m \approx \sqrt{p}$[9]. This may seem illogical because each split is not permitted to consider half of the available predictors. The rationale is actually quite sound in that when using standard bagging, where all the predictors are considered, each split will look similar if a particularly strong predictor exists as it will always be chosen to be used at the top of the split. This causes the trees to

be highly correlated[9]. Averaging a large number of highly correlated predictors does not cause a significant decrease in variance. Making use of a limited number of predictors at each split gives other predictors a chance of being selected, resulting in decorrelated trees and a lower variance.

Machine Learning Benchmarks and Random Forest Regression[11], mathematically help the reader understand the calculation of trees, as well as determining what strategies to employ to achieve desirable results.

Random Forests are comprised of multiple trees as seen where there are M different trees on different subsets of the data, which are chosen randomly to compute the ensemble:

$$f(x) = \sum_{m=1}^{M} \frac{1}{M} f_m(x)$$

where f_m is the m^{th} tree.

As discussed in Machine Learning Benchmarks and Random Forest Regression[9], a random forest is a collection of individual trees. For the sake of this example we make use of the empirical calculations[9].The random forest will be defined as a collection of tree predictors.

$$h(x; \theta_p) \qquad p = 1, \ldots, P$$

x represents the observed input vector of length P. X and θ_k are independent and identically distributed random vectors. The random forest prediction is the (unweighted) average over the collection, such that:

$$\bar{h}(x) = \left(\frac{1}{P}\right) \sum_{k=1}^{P} h(x; \theta_k)$$

It is important to note that as $k \to \infty$ the Law of Large Numbers transforms the equation slightly and designates a prediction error (generalization) PE_f^*. The convergence that takes place implies that random forests do not over fit. As such we now have:

$$PE_t^* = E_\theta E_{X,Y}(Y - h(x; \theta_k))^2 \tag{3}$$

where PE_t^* is the average prediction error for the individual trees.

If we were to assume that all individual trees are unbiased i.e. $EY = E_x h(x; \theta)$ and we define $\bar{p}$ as the weighted correlation between the residuals $Y - h(x; \theta)$ and independent $\theta^{'}$,Thus we are left with:

$$PE_f^* \leq \bar{p} PE_t^* \tag{4}$$

Equation 4 details that for accurate random forest regression two requirements must be met. The first, is a low correlation between residuals of differing tree members of the forest. The randomness injected strives for low correlation. The second, is the low prediction error for individual trees. The expectation is that the random forest will decrease the individual tree error by a factor $\bar{p}$.

To meet the ends detailed above:

- Trees must be grown to their maximum depth to minimize individual error.
- Grow each tree on a bootstrap sample from the training data.
- $m \ll p$ - select m covariates and pick the best split at each node of every tree.

Following the strategy outlined will control bias but not variance[11]. Only by averaging many estimates can the variance be reduced[7]. This is the unfortunate nature of greed by which the tree construction algorithm works. Variance can also only be reduced to a limited extent due to the highly correlated predictors[3]. There

is a partial solution to the problem of controlling variance, or more particularly prediction variance. The expected generalization error of an ensemble corresponds to:

$$var(x) = \rho(x).\sigma^2_{\zeta,\theta}(x) + \left(\frac{1-\rho(x)}{M}.\sigma^2_{\zeta,\theta}(x)\right) \tag{5}$$

$\rho(x)$ is Pearson's correlation coefficient between the predictions of two independent, randomized models trained on the same data. If the number of decision trees (M) in the random forest model increases, the variance of the ensemble decreases when $\rho(x) < 1$. Strictly speaking, the variance of the ensemble is smaller than the variance of the model. Thus increasing the number of individual randomized models (increasing the number of trees) in an ensemble will never increase the generalization error. This essentially means that the random forest model shouldn't over fit if more trees are used in each ensemble. However, this would significantly lower training error but result in a bad prediction error.

2.4 Heteroscedasticity

Homoscedasticity is defined as the constant variance in the response and is an explicit assumption made when using linear regression, thus an implicit one for random forests, as well as other predictive tools [4]. Heteroscadasticity occurs when the assumption of homoscedasticity is violated, or the variance of the response is non-constant.

2.4.1 Basic Overview of Heteroscedasticity

Simple linear regression allows for the study of relationships between p explanatory variables, $X_1, X_2, \ldots, X_p$, and a continuous response variable Y. The model

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

is the matrix form of a simple linear regression where $\boldsymbol{X}$ is an $n \times (p+1)$ matrix of explanatory variables, $\boldsymbol{Y}$ is an $n \times 1$ vector of responses, $\boldsymbol{\beta}$ is a $(p+1) \times 1$ vector of unknown regression coefficients and that $\boldsymbol{\varepsilon}$ is an $n \times 1$ vector of unobserved errors within the regression that have a normal distribution with parameters $N(0, \sigma^2\boldsymbol{I})$[4]. The method of ordinary least squares (OLS) allows for the estimation of the unknown regression coefficients as $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}\text{`}\boldsymbol{X})^{-1}\boldsymbol{X}\text{`}\boldsymbol{Y}$. Under heteroscedasticity these estimators are still unbiased but become inefficient and thus lead to incorrect inferences about the data [4].

Behavior diagnosis of the variance in a data set can make use of the residuals, calculated as $e = \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}$ when considering a simple linear regression. Typically residuals (e_i on the y-axis) are plotted against the predicted responses ($\hat{y}_i$ on the x-axis). Residuals that are plotted randomly and uniformly around the horizontal line (at 0) represent the presence of homoscedasticity[4], whereas if the residuals create a fan shape homoscedasticity is not satisfied, as seen in Figure 2.

The standard residual plot in Figure 2, while the simplest, is not necessarily the best. It may be difficult to interpret, especially if the positive and negative residuals do not exhibit the same general pattern. A proposed change is the square of the residuals, but has the risk of scaling problems when residuals that are large in magnitude are considered. It seems the best method would be the absolute value of the residuals which eliminates the scaling problems[4]. However, due to the nature of the data generated for this paper, this method is not necessary for the identification of heteroscedasticity.

While the above definition and attributed results hold for primarily linear regressive models, the same underlying effects are present in other predictive models such as Random Forests[4]. Subsequent discussion will provide more detail on the effects of heteroscedasticity on the variance of the Random Forest model as well as bias.

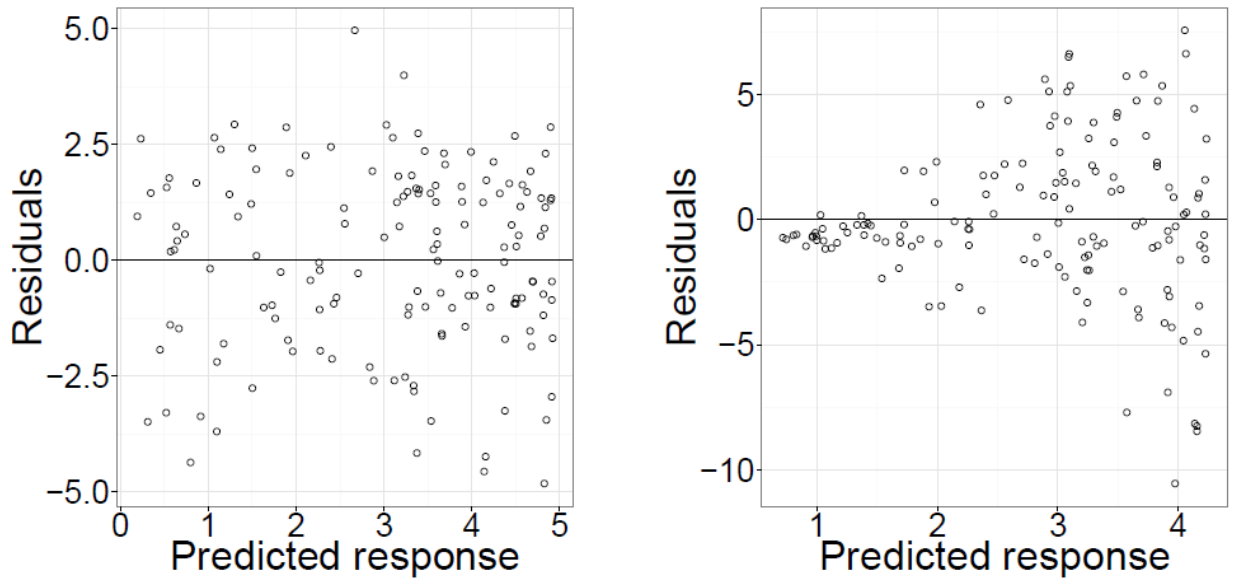


Figure 2: Homoscedastic errors (left) where residuals seem to be randomly scattered around 0. Heteroscedastic errors (right) seem to follow a pattern whereby the residuals start to fan out[4].

3 Application

3.1 Introducing the Data

For the purpose of this research three randomly generated data set's have been created. Each of the three models uses arbitrary coefficients for their relevant explanatory variables and all make use of the same error added to the observed values. The only difference between the models is the number of explanatory variables used by each. The models are comprised of one, five and ten explanatory variables. The purpose behind the addition of explanatory variables is to determine the effect their addition may have on the accuracy of each of the three methods used.

The three models used are listed below:

$$y_i = 100 + 10x_{1i} + \varepsilon_i$$

$$y_i = 100 + 10x_{1i} + 6x_{2i} - 10x_{3i} + 0.5x_{4i} - 2x_{5i} + \varepsilon_i$$

$$y_i = 100 + 10x_{1i} + 6x_{2i} - 10x_{3i} + 0.5x_{4i} - 2x_{5i} + 0.33x_{6i} + 2x_{7i} - 1.375x_{8i} - 6.67x_{9i} - x_{10i} + \varepsilon_i$$

$$\varepsilon_i = (2 \times j)^{1.5} \times \pi(s) \qquad j = 1 \ldots n$$

Where $\pi(s)$ is a randomly generated value from a uniform distribution with a seed value of s.

As mentioned above, three methods will be used. They are, namely: ordinary least squares (OLS), weighted least squares (WLS) and the random forest regressor (RFR) and will be tested on each of the three models.

	Number of Explanatory Variables		
	One	**Five**	**Ten**
OLS	0.002	0.003	0.037
WLS	0.023	0.015	0.022
RFR (n = 100)	0.764702	0.813499	0.832402

Table 1: Coefficient of determination (R^2) comparison between OLS, WLS and RFR

3.2 Experimental Design

3.2.1 Comparative Ability of the Random Forest Regressor

Table 1 compares the relative coefficient of determination (R^2) between the different models and the methods used in each. Expectedly, the ordinary least squares method did not achieve a high score for any of the models, further showcasing its inability to make accurate predictions in the presence of extreme heteroscedasticity, as showcased in Figure 3. As initially proposed by Alexander Aitken[1], a best linear unbiased estimator is achieved when the weighted sum of squared residuals is minimized. This is done, ideally, by the weight being set as the reciprocal of the variance of the measurement. The use of such a weight, though not entirely accurate, yields a higher coefficient of determination value for the one variable case of the weighted least squares method in comparison to the ordinary least squares score obtained for the same model. However, the weighted function incorrectly calculated which resulted in worsted scores for WLS in both the five and ten variable cases.

In contrast, the random forest regressor seems to present considerably better results. Furthermore, the scores seem to increase with an increase in the number of explanatory variables. This effect corresponds with the expectations laid down previously in the theory for the model.

	Number of Explanatory Variables					
	One		**Five**		**Ten**	
	Correlation	**P-Value**	**Correlation**	**P-Value**	**Correlation**	**P-Value**
OLS	0.997737	<0.0001	0.989871	<0.0001	0.946558	<0.0001
WLS	0.996143	<0.0001	0.996378	<0.0001	0.992996	<0.0001
RFR (n = 100)	0.676174	<0.0001	0.715486	<0.0001	0.918688	<0.0001

Table 2: Spearman's rank coefficient correlation test comparison between OLS, WLS, RFR

Table 2 serves to confirm the presence of heteroscedasticity in a non-visual way. This is achieved by the use of Spearman's rank sign test. With the null hypothesis (H_o) of heteroscedasticity, determining the correlation coefficient between the residuals and x_1, though the correlation coefficient can be calculated between the residuals and any of the explanatory variables in this case, due to the overall model heteroscedasticity. Examination of the p-values provided in Table 2 shows that the p-values are far below any significance level set ($\alpha = 0.01/0.05/0.1$). There is, therefore, enough evidence to reject the null hypothesis of homoscedasticity for all the cases. These results are expected for both the OLS and WLS methods. It was unknown whether the random forest regressor would be accurate in detecting heteroscedasticity in this fashion, but it seems very accurate, or at least as accurate at the other two models.

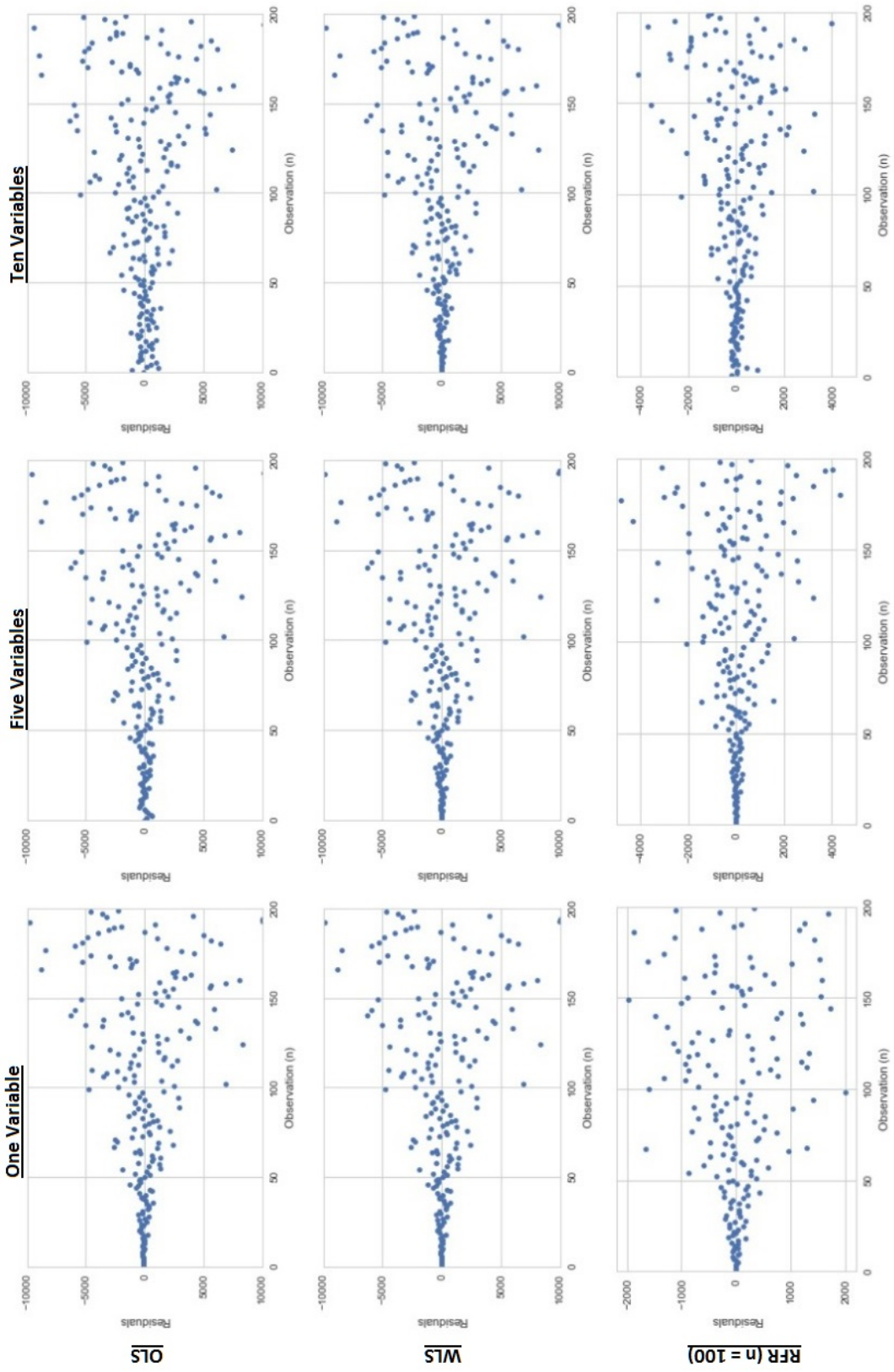


Figure 3: Residual plots of the three methods for each of the three data sets.

The use of the artificial error used by the models outlined above, creates an extreme case of model wide heteroscedasticity. As discussed previously, while there are better means of observing heteroscedasticity, for a case as extreme as this, heteroscedasticity is rather clear. Figure 3 clearly depicts the increasing difference of the residuals over the $n = 200$ observations. Plotting the residuals against the number of observations, as opposed to individual predictor variables, shows the overarching heteroscedasticity across the entire model.

3.2.2 Adjusting the Number of Estimators

The random forest regressor has a host of tuning parameters to ensure that the method is fully taken advantage of. Many of these parameters can boost the results quite significantly, though there are consequences that must be acknowledged.

The n estimators tuning parameters is one such parameter with significant effects on the results of the model, though inappropriate usage of this parameter can not only decrease the accuracy of results, but more importantly, the faith in the results obtained in their entirely. Figure 4 illustrates the out-of-bag (OOB) error rates for the random forest regressors used previously in this paper. The OOB error rates in these graphs depict the prediction error of the random forest method as the number of estimators increases. Typically the more trees allowed in a forest, the better the prediction results obtained. Due to how estimators are averaged, it is also important to not add more trees than necessary. Figure 4 allows the user to better identify the number of trees that should be chosen to gain the best results. Table 3 better shows the effect on prediction gained by the increase of the number estimators. As can be seen, the results in Table 3, an increase in the number of trees available to the random forest method has a significant effect on prediction results.

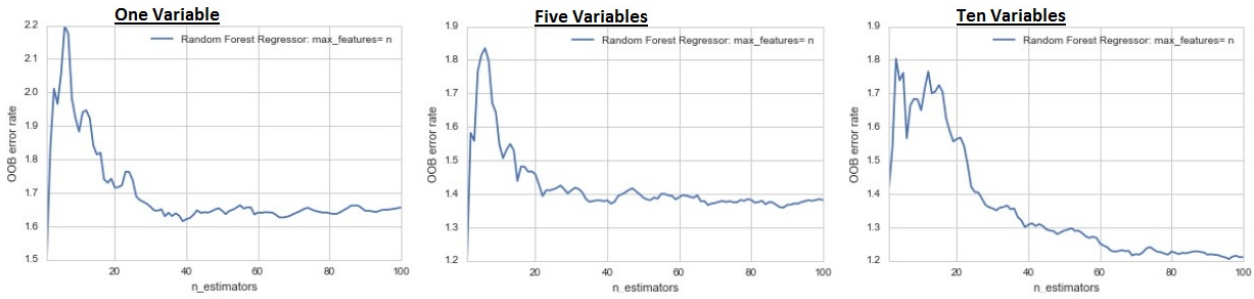


Figure 4: Out-of-bag errors for the random forest regressor over the three models

Prediction score (R^2) change through the increase of the number estimators			
Number of Estimators	1 Variable	5 Variables	10 Variables
1	0.561272	0.211825	0.071152
5	0.619072	0.597685	0.575556
10	0.708317	0.742880	0.798325
20	0.747322	0.796158	0.801613
100	0.764702	0.813499	0.832402

Table 3: Increased prediction of the random forest model when the number of estimators is increased.

4 Final Remarks

4.1 Conclusions

Based on the results obtained from the R^2 values, tests for heteroscedasticity, as well as the out-of-bag error rates there are a number of positive conclusions that can be drawn with respect to the use of the random forest regressor as a method of estimation for linear regression models. Firstly, while the prediction scores obtained through the R^2 values are certainly inflated due to both training and testing the model on the same set of data, the scores are significant improvements from those in the OLS and WLS models, which suffer from the same inflated scores. Secondly, the random forest regressor is as capable in identifying heteroscedasticity with the use of Spearman's rank correlation coefficient. These two events, in conjunction with the results obtained from the OOB error rates, show the random forest regressor is capable of being a method of estimation in linear regression, particularly in the presence of heterosecdasticity. It also generates an internal unbiased estimate of the generalization error as the forest building progresses, a property not always present when making use of OLS.

However, there are a number of factors to consider when attempting to make use of the random forest regressor, aside from the fact that there is significantly less documentation for the RFR in contrast to that of OLS and by extension WLS. Furthermore, OLS has seen extensive practical usage, cementing a certain degree of trust for the method. The OLS method also provides a large number of descriptive statistics which are more easily obtained through its use. Most statistical packages or programs, such as: SAS, the statsmodel package in python, and a variety of packages in R display most of the necessary descriptive statistics as a default when dealing OLS or WLS. A level of support and intuitive output that is not present with the random forest ensemble method.

While the RFR is capable of multiple input variables without variable deletion, the internal safeguards from its derivation may not be efficient in protecting against overfitting when variables are particularly noisy. Due to this the random forest regressor must be handled with the utmost care. There are a number of assumptions that should be met to ensure the validity of the results obtained, these are:

1. A bootstrap sample must be used, not the entire dataset, when growing trees.
2. The data set is split into a training and testing set. Without considering optimization of the training set, a split of the sets would allow the random forest regressor to accurately learn from the testing set. This learning would not, however, be biased in that the method is tuned to specific parameters. This tuning no longer makes the testing set independent, thus lowering the credibility of its results.

4.2 Future Studies

Machine learning algorithms are continuously being developed and adjusted to meet the demands for data analysis. Many of these methods may be incorrectly used to deliver what are perceived desirable results. These results are not necessarily reliable and could be further affected by other statistical phenomena, such as multicollinearity. Further research could aim to expand on the capabilities of the random forest method by:

1. Determining what action can be taken by the random forest method in the event of heteroscedastic data.
2. Estimation of more descriptive statistics which could contribute to various forms of hypothesis testing.
3. Find ways to more easily interpret the results of a random forest model, and how to make use of those results.

References

[1] Alexander C. Aitken. *On Least Squares and Linear Combination of Observations.* Proceedings of the Royal Society of Edinburgh, 1936.

[2] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[3] 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(23):81–227, 2012.

[4] Sharla Jaclyn Gelfand. Understanding the impact of heteroscedasticity on the predictive ability of modern regression methods. Master's thesis, University of Calgary, 2013.

[5] Tin Kam Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition - Volume 1*, ICDAR '95, pages 278–, Washington, DC, USA, 1995. IEEE Computer Society.

[6] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R.* Springer Publishing Company, Incorporated, 2014.

[7] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[8] Thomas M. Mitchell. *Machine Learning.* McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

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

[10] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1):1–39, 2010.

[11] Mark R. Segal. Machine learning benchmarks and random forest regression. *Division of Biostatistics, University of California, San Francisco*, 2004.

[12] Xin Yan and Xiao Gang Su. *Linear Regression Analysis: Theory and Computing.* World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2009.

Appendix

Code

Random Data Generation

The data sets used in the paper were all randomly generated using Statistical Analysis System (SAS) and make use of arbitrary coefficients and seed values.

```
proc iml; n=200;
Seedx = j(n,1,567567);
x1 = rannor(seedx)*10+40;
call sort(x1);
s = 2#((1:n)`)##1.5;
seedu = j(n,1,345345);
u = rannor(seedu)#s;

x2 = rannor(Seedx)*1-35;
x3 = rannor(Seedx)*2+17;
x4 = rannor(Seedx)*5+11;
x5 = rannor(Seedx)*6-0.5;
x6 = rannor(seedx)*2.5+87;
x7 = rannor(seedx)*7+20;
x8 = rannor(seedx)*11+5;
x9 = rannor(seedx)*3-4;
x10 = rannor(seedx)*1+0;

y=100+(10*x1)+(6*x2)-(10*x3)+(0.5*x4)-(2*x5)+(0.33*x6)
+(2*x7)+(1.375*x8)-(6.67*x9)+(1*x10)+u;
y = 100 + 10*x1 +6*x2 -10*x3 +0.5*x4 -2*x5 +u;*/ y = 100 + 10*x1 +u;
FullData=y||x1||x2||x3||x4||x5||x6||x7||x8||x9||x10||s;
FullData=y||x1||x2||x3||x4||x5||s;
FullData=y||x1||s;
column_names={"Y" "X1" "X2" "X3" "X4" "X5" "X6" "X7" "X8" "X9" "X10" "sig"};
column_names={"Y" "X1" "X2" "X3" "X4" "X5" "sig"};
column_names={"Y" "X1" "sig"};

create output from FullData[colname=column_names];
append from FullData;
quit;

proc export data=output
outfile='C:\Users\Philip\Anaconda3\Inputs\Research_Honours\TenVariable.csv'
outfile='C:\Users\Philip\Anaconda3\Inputs\Research_Honours\FiveVariable.csv'
outfile='C:\Users\Philip\Anaconda3\Inputs\Research_Honours\OneVariable.csv'
dbms=csv     replace;
run;
```

Model Estimation

All model estimation used in this paper, and as a result all graphical output, unless referenced otherwise, made exclusive use of Python and Python packages.

```
# Statsmodels
from __future__ import print_function
import statsmodels.api as sm i
mport statsmodels.formula.api as smf
import statsmodels.stats.api as sms
import scipy as sp from statsmodels.compat
import lzip from statsmodels.sandbox.regression.predstd
import wls_prediction_std from statsmodels.graphics.regressionplots
import plot_regress_exog from scipy
import stats from scipy
import array, linalg, dot from scipy.linalg
import toeplitz from statsmodels.formula.api
```

```
13 import ols, wls, gls from statsmodels.sandbox.regression.predstd
14 import wls_prediction_std from statsmodels.iolib.table import (SimpleTable, default_txt_fmt)
15 import random
16
17 # pandas import pandas as pd
18 from pandas import Series,DataFrame
19
20 # numpy, matplotlib
21 import numpy as np
22 import matplotlib.pyplot as plt
23
24 # machine learning
25 from sklearn.linear_model import LogisticRegression as logreg
26 from sklearn.ensemble import RandomForestRegressor
27 from collections import OrderedDict
28 from sklearn.datasets import make_classification
29 from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
30 from sklearn.tree import DecisionTreeRegressor
31 from sklearn.metrics import roc_curve, auc, confusion_matrix,
32 \ classification_report, accuracy_score, recall_score, precision_score,
33 \ f1_score, roc_auc_score
34 from math import log10
35
36 Dataset = pd.read_csv("C://OneVariable.csv")
37 #Dataset = pd.read_csv("C://FiveVariable.csv")
38 #Dataset = pd.read_csv("C://TenVariable.csv")
39
40 #ONE VARIABLE CASE#
41 Y = Dataset['Y']
42 X1 = Dataset['X1']
43 S = Dataset['sig']
44 X = sm.add_constant(X1)
45 #END OF ONE VARIABLE CASE#
46
47 #FIVE VARIABLE CASE#
48 Y = Dataset['Y']
49 X1 = Dataset['X1']
50 X1 = np.row_stack(X1)
51 X2 = Dataset['X2']
52 X2 = np.row_stack(X2)
53 X3 = Dataset['X3']
54 X3 = np.row_stack(X3)
55 X4 = Dataset['X4']
56 X4 = np.row_stack(X4)
57 X5 = Dataset['X5'] X5 = np.row_stack(X5)
58 S = Dataset['sig']
59 Xc = np.column_stack((X1,X2,X3,X4,X5))
60 X = sm.add_constant(Xc)
61 #END OF FIVE VARIABLE CASE$
62
63 #TEN VARIABLE CASE#
64 Y = Dataset['Y']
65 X1 = Dataset['X1']
66 X1 = np.row_stack(X1)
67 X2 = Dataset['X2']
68 X2 = np.row_stack(X2)
69 X3 = Dataset['X3']
70 X3 = np.row_stack(X3)
71 X4 = Dataset['X4']
72 X4 = np.row_stack(X4)
73 X5 = Dataset['X5']
74 X5 = np.row_stack(X5)
75 X6 = Dataset['X6']
76 X6 = np.row_stack(X6)
77 X7 = Dataset['X7']
78 X7 = np.row_stack(X7)
79 X8 = Dataset['X8']
```

```
X8 = np.row_stack(X8)
X9 = Dataset['X9']
X9 = np.row_stack(X9)
X10 = Dataset['X10']
X10 = np.row_stack(X10)
S = Dataset['sig']
Xc = np.column_stack((X1,X2,X3,X4,X5,X6,X7,X8,X9,X10))
X = sm.add_constant(Xc)
#END OF TEN VARIABLE CASE#

#OLS MODEL
OLSfit = sm.OLS(Y,X).fit()
print(OLSfit.summary())

#SPEARMANRANK CORRELATION COEFFICIENT (OLS)
E_ols = OLSfit.resid
E_ols = np.row_stack(E_ols)
Y_t = np.row_stack(Y)
Spear = sp.stats.spearmanr(Y_t,E_ols)
Spear

#RESIDUAL PLOT vs OBS (OLS)
plt.plot(E_ols,'#4C72B0', marker = ".",markersize = 10,linestyle = "None")
plt.xlabel('Observation (n)')
plt.ylabel('Residuals')

#OTHER RESIDUAL PLOTS vs Xi (WLS)
#SUBSTITUTE "X1" FOR PREDICTOR OF INTEREST
fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(OLSfit, "X1", fig=fig)

#WLS MODEL
mod_wls = sm.WLS(Y, X, weights=1./S)
WLSfit = mod_wls.fit()
print(WLSfit.summary())

#SPEARMANRANK CORRELATION COEFFICIENT (WLS)
E_wls = WLSfit.resid
E_wls = np.row_stack(E_wls)
Y_t = np.row_stack(Y)
Spear = sp.stats.spearmanr(Y_t,E_wls)
Spear

#RESIDUAL PLOT vs OBS (WLS)
plt.plot(E_wls,'#4C72B0', marker = ".",markersize = 10,linestyle = "None")
plt.xlabel('Observation (n)')
plt.ylabel('Residuals')

#OTHER RESIDUAL PLOTS vs Xi (WLS)
#SUBSTITUTE "X1" FOR PREDICTOR OF INTEREST
colour1 = '#cae8dc'
fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(WLSfit, "X1", fig=fig)

#RANDOM FOREST REGRESSOR
model = RandomForestRegressor(n_estimators=100, oob_score = 'True',
warm_start= 'True',max_features = None)
model.fit(X, Y)
model.score(X,Y)

#SPEARMANRANK CORRELATION COEFFICIENT (RFR)
predicted = model.predict(X)
predicted = np.row_stack(predicted)
Y = np.row_stack(Y)
Error = Y - predicted
Y_t = np.row_stack(Y)
Spear = sp.stats.spearmanr(Y_t,Error)
```

```
Spear

#RESIDUAL PLOT vs OBS (RFR)
plt.plot(Error,'#4C72B0', marker = ".",markersize = 10,linestyle = "None")
plt.xlabel('Observation (n)')
plt.ylabel('Residuals')

#OTHER RESIDUAL PLOTS vs Xi (RFR)
#SUBSTITUTE "X1" FOR PREDICTOR OF INTEREST
X1 = np.row_stack(X1)
plt.plot(X1,Error,'#4C72B0', marker = ".",markersize = 10,linestyle = "None")
plt.xlabel('X1')
plt.ylabel('Residuals')

#OUT-OF-BAG (OOB) ERRORS - RFR

from collections import OrderedDict
ensemble_clfs = [("Random Forest Regressor: max_features= n",
RandomForestRegressor(warm_start=True, max_features="log2", oob_score=True,))]
error_rate = OrderedDict((label, []) for label, _ in ensemble_clfs)
min_estimators = 1
max_estimators = 100
for label, clf in ensemble_clfs: for i in range(min_estimators, max_estimators + 1):
clf.set_params(n_estimators=i)
clf.fit(X, Y)
oob_error = 1 - clf.oob_score_ error_rate[label].append((i, oob_error))

for label, clf_err in error_rate.items(): xs, ys = zip(*clf_err)
plt.plot(xs, ys, label=label)
plt.xlim(min_estimators, max_estimators)
plt.xlabel("n_estimators")
plt.ylabel("OOB error rate")
plt.legend(loc="upper right")
plt.show()
```

Identifying the best maximum likelihood estimator for the Gutenberg-Richter $b-$value

Hannaline Roux 13040911

WST795 Research Report

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

Supervisor: Mr MT Loots, Co-supervisors: Prof A Kijko, Ms A Smit

Department of Statistics, University of Pretoria

2 November 2016

Abstract

In order to identify the best maximum likelihood estimator for the Gutenberg-Richter $b-$value, we have to compare the properties of the two most suitable methods. There are four different methods available but the maximum likelihood estimate for β, proposed by Kijko-Smit (2012) and Kijko-Sellevoll (1989), are the main focus of this paper [4, 3]. Both a theoretical and empirical comparison are required, where the theoretical part will entail the derivations of the maximum likelihood estimator and a full explanation of all relevant parameters. A hypothetic seismic event catalogue is also provided, using the Monte Carlo simulation for the comparison of empirical properties. Emphasis is placed on the different properties in order to identify the best method for the $b-$value estimate.

Declaration

I, *Hannaline Roux,* 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.

_____
Hannaline Roux

_____
Mr MT Loots, Prof A Kijko, Ms A Smit

_____
Date

Acknowledgements

A very special thank you to my supervisors, Mr Theodor Loots, Ms Ansie Smit and Professor Andrzej Kijko; your guidance and support have been invaluable. Acknowledgement must also be made to the University of Pretoria Natural Hazard Centre. I would also like to thank the National Research Foundation (NRF) for their financial support.

Contents

1 Introduction **6**

2 Theoretical Background **7**
2.1 The generalized Aki-Utsu method 7
2.2 Kijko-Sellevol (1989) method 9

3 Application **11**
3.1 Investigation of the generalized Aki-Utsu method 11
3.1.1 Generating the $\hat{\beta}$-values 11
3.1.2 Distribution of the b-values 13
3.1.3 Using other statistical methods for comparison 14
3.2 Investigation of the Kijko-Sellevol (1989) method 14
3.2.1 Generating the $\hat{\beta}$-values 14
3.2.2 Distribution of the b-values 15
3.2.3 Using other statistical methods for comparison 16
3.3 Comparison between the generalized Aki-Utsu and the Kijko-Sellevoll (1989) method 17

4 Conclusion **19**

Appendix A **21**

Appendix B **26**

List of Figures

1 A schematic illustration of a seismic event catalogue for s level of completeness [4]. 7
2 The 500 values generated for the generalized Aki-Utsu $\hat{\beta}$-value estimator. 12
3 The 500 generated b-values. 12
4 Q-Q Plot based on the b-values. 13
5 Histogram based on the b-values. 13
6 The 500 values generated for the Kijko-Sellevol (1989) $\hat{\beta}$-value estimator. 14
7 The 500 generated b-values. 15
8 Q-Q Plot based on the b-values. 15
9 Histogram based on the b-values. 16
10 Boxplot for the generalized Aki-Utsu method. 17
11 Boxplot for the Kijko-Sellevoll (1989) method 18

List of Tables

1 Comparison between the generalized Aki-Utsu method and the Kijko-Sellevoll (1989) method 17

1 Introduction

The occurrence of earthquakes can be represented by a likelihood function of the sample data given a probability distribution function (PDF), which is derived from the frequency-magnitude Gutenberg-Richter law [4]. The expression contains an unknown parameter b, which is also known as a seismic activity parameter [3], where the value of this parameter that maximises the sample likelihood is known as the maximum likelihood estimator (in short, MLE).

Studying the distribution of earthquakes provides a better understanding of the physics and kinetics behind earthquake processes [7]. The empirical relation between the frequency and magnitude of earthquakes, also known as the frequency-magnitude Gutenberg-Richter law, can be expressed by the equation:

$$\log(n) = a - bm$$

where a is a measure of the level of seismicity, parameter b the ratio between the number of small and large events (also referred to as a size distribution) and m represents magnitude [4]. The estimation of the parameter b is crucial in seismic hazard studies, as well as in verifying theoretical assertions, since it varies over space and time [6][7].

In order to identify the best maximum likelihood estimator for the b-value estimate we are considering two different methods, from the four methods given, where extreme observations are ignored. The first method, proposed by Kijko-Smit (2012), entails a generalized Aki-Utsu $\hat{\beta}$-value estimator which measures different levels of completeness of multiple catalogues where $\beta = b\ln(10)$ [4]. This estimator is known for its simplicity and the fact that the incomplete catalogues can be divided into sub-catalogues, each with a different level of completeness [4]. The second method, proposed by Kijko-Sellevoll (1989); consists of a standard method only used for the complete younger parts of the catalogue, allowing us to derive the maximum likelihood estimate for β [3]. The properties of these two methods are discussed in depth in order to identify the best maximum likelihood estimator.

Since we are working with a multivariate distribution, the magnitudes m, of seismic events are independent and identically distributed random variables, where the probability distribution (or mass) function of each m_j is $f(m_j^i, \beta)$ [4]. The joint density of the magnitudes, by independence, will then be equal to the product of the marginal densities which can be used to calculate the log-likelihood function. In order to maximise this function we need to calculate the first order conditions with respect to β[8] .

The Monte Carlo methods are stochastic techniques which are based on the use of random numbers and probability statistics to simulate problems [5]. The Statistical Analysis Software (SAS®), along with the Monte Carlo simulation, are used to generate random numbers for the different magnitudes according to the relevant probability distribution function (PDF). These random numbers then create a hypothetic seismic event catalogue.

To summarise; given the frequency-magnitude Gutenberg-Richter earthquake distribution we derive the theoretical properties of the identified maximum likelihood estimators. These properties are then compared using the two different methods mentioned above. Both a theoretical and empirical comparison are made through various statistical procedures before the best method can be selected.

2 Theoretical Background

The theory and derivations of the first method are explained in the article by A. Kijko and A. Smit entitled *Extension of the Aki-Utsu b-Value Estimator for Incomplete catalogues* [4] as well as the article by Dieter H. Weichert entitled *Estimation of the earthquake recurrence parameters for unequal observation periods for different magnitudes* [9]. The theory of the second method is explained by A. Kijko and M. A. Sellevoll in the article; *Estimation of earthquake hazard parameters from incomplete data files. Part I. Utilization of extreme and complete catalogues with different threshold magnitudes* [3]. The theory and derivations of the maximum likelihood estimators and a full explanation of all relevant parameters will be discussed in the following sub-sections.

2.1 The generalized Aki-Utsu method

The generalized Aki-Utsu $\hat{\beta}$-value estimator measures different levels of completeness of multiple catalogues $m^1_{min}, m^2_{min}, m^3_{min}, ..., m^s_{min}$, which is also known for its simplicity and the fact that the incomplete catalogues can be divided into sub-catalogues, each with a different level of completeness [4]. This estimator is an extension of the maximum likelihood estimate for the Gutenberg-Richter $b-$value proposed by Aki [1], where extreme observations are ignored for simplicity.

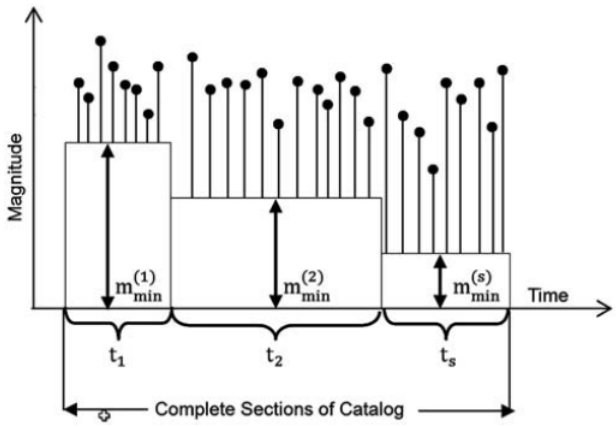


Figure 1: A schematic illustration of a seismic event catalogue for s level of completeness [4].

In order for us to derive the maximum likelihood estimator for the $b-$value estimate we need to create a likelihood function, given the probability distribution function (PDF) of earthquake magnitudes, also known as a shifted exponential distribution [1]:

$$f(m;\beta) = \begin{cases} 0 & m < m_{min} \\ \beta(\exp(-\beta(m - m_{min})) & m \geq m_{min} \end{cases} \tag{1}$$

where m, represents the magnitude of a seismic event which is assumed to be a continuous, independent and identically distributed random variable [4]. For the derivations we assume that the magnitude will always be greater or equal than the level of completeness m_{min}, where $\beta = b\ln(10)$.
The likelihood function of the parameter β is defined as the product of the probability distribution function, but we observe data within the i-th sub-catalogue where $i = 1, 2...s$, we then define the likelihood function for the i-th sub-catalogue as follows:

$$L_i(\beta) = \prod_{j=1}^{n_i} f(m^i_j, \beta) = \prod_{j=1}^{n_i} \beta(\exp(-\beta(m^i_j - m^i_{min}))$$

Since the generalized Aki-Utsu $\hat{\beta}$-value estimator measures s levels of completeness, we can now define the joint likelihood function of all earthquakes that occurred within the entire time span of the catalogue (refer to Figure 1) as follows:

$$\begin{aligned} L(\beta) = \prod_{i=1}^{s}\prod_{j=1}^{n_i} f(m_j^i, \beta) &= \prod_{i=1}^{s}\prod_{j=1}^{n_i} \beta(\exp(-\beta(m_j^i - m_{min}^i)) \\ &= \sum_{i=1}^{s}\sum_{j=1}^{n_i} \beta(\exp(-\beta(m_j^i - m_{min}^i)) \end{aligned} \tag{2}$$

where m_j^i is the sample of n_i earthquake magnitudes observed during the time span of the i-th sub-catalogue. From the above we can now determine the log-likelihood function:

$$\ln L(\beta) = \sum_{i=1}^{s}\sum_{j=1}^{n_i} \ln(\beta) + \sum_{i=1}^{s}\sum_{j=1}^{n_i} (-\beta(m_j^i - m_{min}^i))$$

Now solving the partial derivative with respect to β we obtain the following:

$$\frac{\partial \ln L(\beta)}{\partial \beta} = \sum_{i=1}^{s}\sum_{j=1}^{n_i} \frac{1}{\beta} + \sum_{i=1}^{s}\sum_{j=1}^{n_i} \left[-(m_j^i - m_{min}^i)\right]$$

By setting the above equal to zero we can then obtain the maximum likelihood estimate for β:

$$\frac{\partial \ln L(\beta)}{\partial \beta} = \sum_{i=1}^{s}\sum_{j=1}^{n_i} \frac{1}{\beta} + \sum_{i=1}^{s}\sum_{j=1}^{n_i} \left[-(m_j^i - m_{min}^i)\right] = 0$$

$$\therefore \frac{1}{\beta_1} = \frac{\sum_{j=1}^{n_i} m_j^1}{n_1} - m_{min}^1,\ \frac{1}{\beta_2} = \frac{\sum_{j=1}^{n_i} m_j^2}{n_2} - m_{min}^2, ..., \frac{1}{\beta_s} = \frac{\sum_{j=1}^{n_i} m_j^s}{n_s} - m_{min}^s$$

Now let $r_1 = \dfrac{n_1}{n_1 + n_2}$, $r_2 = \dfrac{n_2}{n_1 + n_2}$,...,$r_s = \dfrac{n_s}{n_1 + n_s}$
Then we can say that the maximum likelihood estimate for β is equal to the following:

$$\frac{1}{\hat{\beta}} = \left(\frac{r_1}{\hat{\beta}_1} + \frac{r_2}{\hat{\beta}_2} + ... + \frac{r_s}{\hat{\beta}_s}\right)$$

Or equivalently:

$$\hat{\beta} = \left(\frac{r_1}{\hat{\beta}_1} + \frac{r_2}{\hat{\beta}_2} + ... + \frac{r_s}{\hat{\beta}_s}\right)^{-1} \tag{3}$$

where $r_i = \frac{n_i}{n}$; $n = \sum_{i=1}^{s} n_i$ is the total number of earthquakes occurred with magnitudes equal to or exceeding the level of completeness. The $\hat{\beta}_i$'s are the Aki-Utsu estimators calculated for the individual sub-catalogues i [4].
The main feature of the the generalized Aki-Utsu $\hat{\beta}$-value estimator is its simplicity which can be seen by equation 3, this estimator is also a straightforward way to measures different levels of completeness of multiple catalogues.

2.2 Kijko-Sellevol (1989) method

The Kijko-Sellevoll (1989) method consists of a standard method, only used for the complete younger parts of the catalogue [3]. In order for us to derive the maximum likelihood estimate for β, we need to denote each sub-catalogue with a time span T_i. We also assume that $i = 1, ..., s$ (number of sub-catalogues) and $j = 1, ..., n_i$(number of events in a sub-catalogue) where m_j^i is then the sample of n_i magnitudes observed in a given time span T_i [3]. The cumulative distribution function (CDF) for magnitudes of events larger than the magnitude of threshold level m_{min} are defined as:

$$F_M(m \mid m_{min}) = F_M(M > m \mid m_{min}) = \begin{cases} 0 & m < m_{min} \\ 1 - \exp(-\beta(m - m_{min})) & m \geq m_{min} \end{cases}$$

Let m_{min}^i be the level of completeness for the i-th sub-catalogue and $m_0 = min(m_{min}^i)$ which denote the minimum of all the sub-catalogues. We then assume that the number of earthquake occurrences per unit of time have a Poisson distribution and secondly, we assume that $\log(n) = a - bm$, where $\beta = b\ln(10)$.The number of earthquakes per unit of time for the i-th level of completeness can be described by the following CDF :

$$P(m_i, T_i) = \frac{\exp(-\lambda_i T_i)(\lambda_i T_i)^{n_i}}{n_i!}$$

where $\lambda_i = \lambda(m_{min}^i) = \lambda(m_0)[1 - F_M(m_{min}^i \mid m_0; \beta)] = \lambda(m_0)[\exp(-\beta(m_{min}^i - m_0))]$ and $\lambda(m_0) = \lambda_0$. Now we can derive the maximum likelihood estimate for β and λ_0 by defining the likelihood function for λ:

$$\begin{aligned} L(n_1, n_2, ..., n_s; T_1, T_2, ..., T_s) &= L(\lambda_0, \beta) \\ &= \prod_{i=1}^{s} P(n_i, T_i) \\ &= \prod_{i=1}^{s} \frac{\exp(-\lambda_i T_i)(\lambda_i T_i)^{n_i}}{n_i!} \\ &= \frac{\exp(-\sum_{i=1}^{s} \lambda_i T_i) \prod_{i=1}^{s} (\lambda_i T_i)^{n_i}}{\prod_{i=1}^{s} n_i!} \end{aligned}$$

where the likelihood function for β follows from 2. We can now define the joint likelihood function for β and λ as follows:

$$\begin{aligned} L &= L_\lambda L_\beta \\ &= \frac{\exp(-\sum_{i=1}^{s} \lambda_i T_i) \prod_{i=1}^{s} (\lambda_i T_i)^{n_i} \sum_{i=1}^{s} \sum_{j=1}^{n_i} \beta(\exp(-\beta(m_j^i - m_{min}^i))}{\prod_{i=1}^{s} n_i!} \\ &= \frac{\exp(-\sum_{i=1}^{s} T_i \lambda_0 [\exp(-\beta(m_{min}^i - m_0))]) \prod_{i=1}^{s} (T_i \lambda_0 [\exp(-\beta(m_{min}^i - m_0))])^{n_i} \sum_{i=1}^{s} \sum_{j=1}^{n_i} \beta(\exp(-\beta(m_j^i - m_{min}^i))}{\prod_{i=1}^{s} n_i!} \end{aligned}$$

From the equation above we can now determine the log-likelihood function:

$$\begin{aligned} \ln(L) &= -\sum_{i=1}^{s} T_i \lambda_0 \exp(-\beta(m_{min}^i - m_0)) + \sum_{i=1}^{s} n_i [\ln(T_i) + \ln(\lambda_0) - \beta(m_{min}^i - m_0)] + \sum_{i=1}^{s} \sum_{j=1}^{n_i} [\ln(\beta) - \beta(m_j^i - m_{min}^i)] \\ &\quad - \ln[\prod_{i=1}^{s} n_i!] \end{aligned}$$

Now solving the partial derivatives with respect to λ_0 we obtain the following:

$$\begin{aligned}\frac{\partial \ln L}{\partial \lambda_0} &= -\sum_{i=1}^{s}\left[T_i[\exp(-\beta(m^i_{min}-m_0))]\right]+\sum_{i=1}^{s} n_i \frac{1}{\lambda_0[\exp(-\beta(m^i_{min}-m_0))]T_i}[\exp(-\beta(m^i_{min}-m_0))] \\ &= -\sum_{i=1}^{s}\left[T_i[\exp(-\beta(m^i_{min}-m_0))]\right]+\sum_{i=1}^{s} n_i \frac{1}{\lambda_0}\end{aligned}$$

By setting the above equal to zero we can then obtain the maximum likelihood estimate for λ_0:

$$\hat{\lambda_0} = \sum_{i=1}^{s}\left(\frac{n_i}{T_i \exp(-\beta(m^i_{min}-m_0))}\right) \tag{4}$$

We can now substitute 4 into the log-likelihood function to obtain the following:

$$\begin{aligned}\ln(L) &= -\sum_{i=1}^{s} T_i \exp(-\beta(m^i_{min}-m_0))\sum_{i=1}^{s}\left(\frac{n_i}{T_i \exp(-\beta(m^i_{min}-m_0))}\right)+\sum_{i=1}^{s} n_i \ln(T_i) \\ &+\sum_{i=1}^{s} n_i \ln\left(\sum_{i=1}^{s}\left(\frac{n_i}{T_i \exp(-\beta(m^i_{min}-m_0))}\right)\right)-\sum_{i=1}^{s} n_i\beta(m^i_{min}-m_0) \\ &+\sum_{i=1}^{s}\sum_{j=1}^{n_i}[\ln(\beta)-\beta(m^i_j-m^i_{min})]-\ln[\prod_{i=1}^{s} n_i!] \\ &= -\sum_{i=1}^{s} T_i \exp(-\beta(m^i_{min}-m_0))\sum_{i=1}^{s}\left(\frac{n_i}{T_i \exp(-\beta(m^i_{min}-m_0))}\right)+\sum_{i=1}^{s} n_i \ln(T_i) \\ &+2\sum_{i=1}^{s} n_i \ln\left(\frac{T_i \exp(-\beta(m^i_{min}-m_0))}{n_i}\right)^{-1}-\sum_{i=1}^{s} n_i\beta(m^i_{min}-m_0) \\ &+\sum_{i=1}^{s}\sum_{j=1}^{n_i}[\ln(\beta)-\beta(m^i_j-m^i_{min})]-\ln[\prod_{i=1}^{s} n_i!] \\ &= -\sum_{i=1}^{s} n_i+\sum_{i=1}^{s} n_i \ln(T_i)-2\sum_{i=1}^{s} n_i \ln\left(\left(\frac{T_i}{n_i}\right)-\beta(m^i_{min}-m_0)\right)-\sum_{i=1}^{s} n_i\beta(m^i_{min}-m_0) \\ &+\sum_{i=1}^{s}\sum_{j=1}^{n_i}[\ln(\beta)-\beta(m^i_j-m^i_{min})]-\ln[\prod_{i=1}^{s} n_i!]\end{aligned}$$

Now we can solve the partial derivatives with respect to β:

$$\frac{\partial \ln L}{\partial \beta} = 2\sum_{i=1}^{s} n_i(m^i_{min}-m_0)-\sum_{i=1}^{s} n_i(m^i_{min}-m_0)+\sum_{i=1}^{s}\sum_{j=1}^{n_i}\frac{1}{\beta}-\sum_{i=1}^{s}\sum_{j=1}^{n_i}(m^i_j-m^i_{min})$$

By setting the above equal to zero we can then obtain the maximum likelihood estimate for β:

$$\begin{aligned}\sum_{i=1}^{s}\sum_{j=1}^{n_i}\frac{1}{\beta} &= \sum_{i=1}^{s}\sum_{j=1}^{n_i}(m^i_j-m^i_{min})-\sum_{i=1}^{s} n_i(m^i_{min}-m_0) \\ \frac{1}{\hat{\beta}} &= \frac{\sum_{i=1}^{s}\sum_{j=1}^{n_i}(m^i_j-m^i_{min})-\sum_{i=1}^{s} n_i(m^i_{min}-m_0)}{\sum_{i=1}^{s} n_i}\end{aligned}$$

Or equivalently:

$$\hat{\beta} = \frac{\sum_{i=1}^{s} n_i}{\sum_{i=1}^{s}\sum_{j=1}^{n_i}(m_j^i - m_{min}^i) - \sum_{i=1}^{s} n_i(m_{min}^i - m_0)} \tag{5}$$

Although we have another parameter $\hat{\lambda_0}$, our main focus are the values for $\hat{\beta}$.

3 Application

The performance of the two methods are investigated by using Monte Carlo simulation. The Monte Carlo methods are stochastic techniques which is based on the use of random numbers and probability statistics to simulate problems [5]. The Statistical Analysis Software (SAS®), along with the Monte Carlo simulation, are used to generate random numbers for the different magnitudes according to the relevant probability distribution function (PDF). These random numbers then create a hypothetical seismic event catalogue. The SAS code used throughout the application can be found in Appendix A and the output in Appendix B.

3.1 Investigation of the generalized Aki-Utsu method

We assume that the hypothetic seismic event catalogue can be divided into two sub-catalogues with level of completeness $m_{min}^1 = 4.0$ and $m_{min}^2 = 3.0$ respectively [4]. The magnitudes are generated according to the probability distribution function (PDF) of earthquake magnitudes given in equation 1 ,where $\beta = 2.303$. Equivalently, the Gutenberg-Richter b-value was equal to 1 since $\beta = b\ln(10)$. The simulation was repeated 500 times for the different number of events, which is defined as the total number of events in both sub-catalogues.
It is necessary to calculate the average of the 500 solutions of the Gutenberg-Richter b-value accordingly to the generalized Aki-Utsu $\hat{\beta}$-value estimator, which measures different levels of completeness of multiple catalogues, in order to investigate how well the estimator performs. We can then determine if the b-value is overestimated and biased. Confidence intervals are also an important aspect to consider since it will allow us to see the relationship between the width of the intervals and the different number of events.

3.1.1 Generating the $\hat{\beta}$-values

The magnitudes generated for the first level of completeness $m_{min}^1 = 4.0$, contains $n_1 = 41$ number of events, where we calculate the average of the magnitudes. These values are then used to calculate $\hat{\beta_1}$in order for us to generate values for $\hat{\beta}$. Similarly we generated values for the second level of completeness $m_{min}^2 = 3.0$, which contains $n_2 = 278$ number of events. We can now substitute our values in equation 3 to generate the 500 solutions for the generalized Aki-Utsu $\hat{\beta}$-value estimator. These values can be shown by the following scatter plot where we compare the values to $\beta = 2.303$:

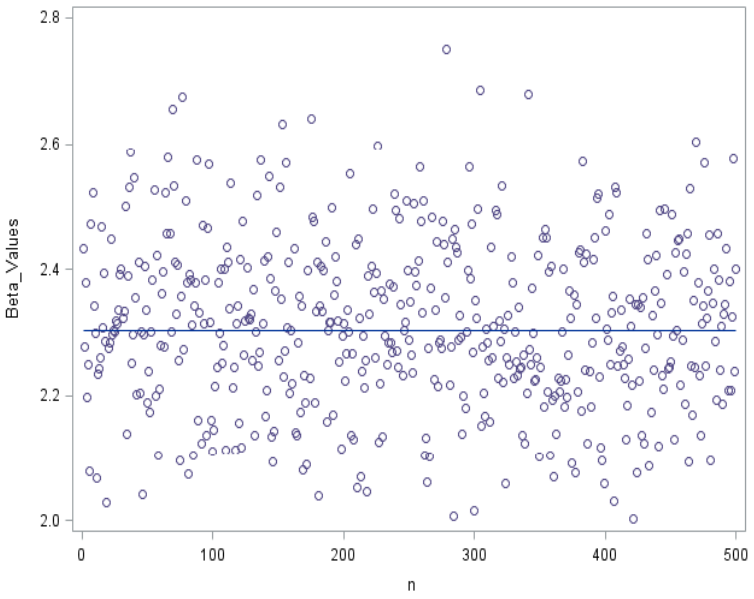


Figure 2: The 500 values generated for the generalized Aki-Utsu $\hat{\beta}$-value estimator.

From Figure 2 we can clearly see that all the values for $\hat{\beta}$ fluctuate around the value $\beta = 2.303$ (demonstrated by the straight line). Converting the $\hat{\beta}$-values to b-values by using the equation $b = \frac{\hat{\beta}}{\ln(10)}$ we get results which are easier to interpret, since we compare these values to $b = 1$, which can be seen by the following figure:

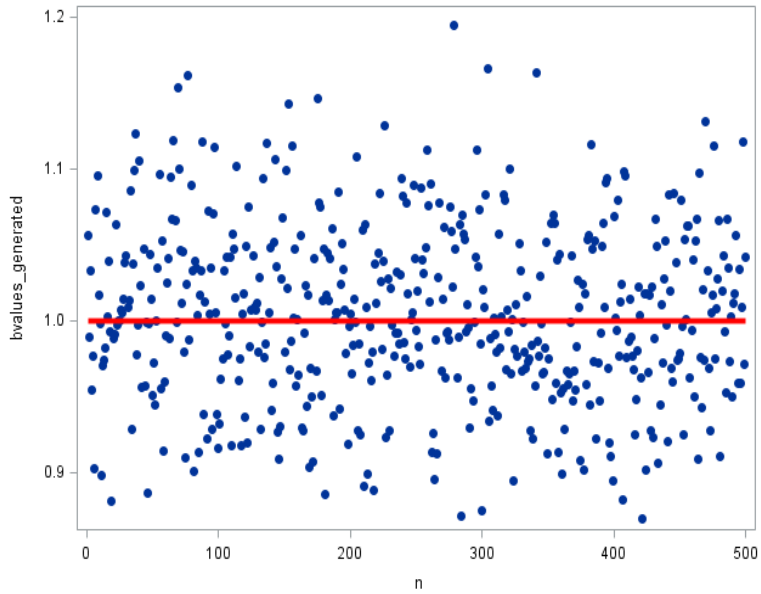


Figure 3: The 500 generated b-values.

3.1.2 Distribution of the b-values

By using the PROC TTEST procedure in SAS, we can test the distribution of the b-values based on the Q-Q plot obtained from the output:

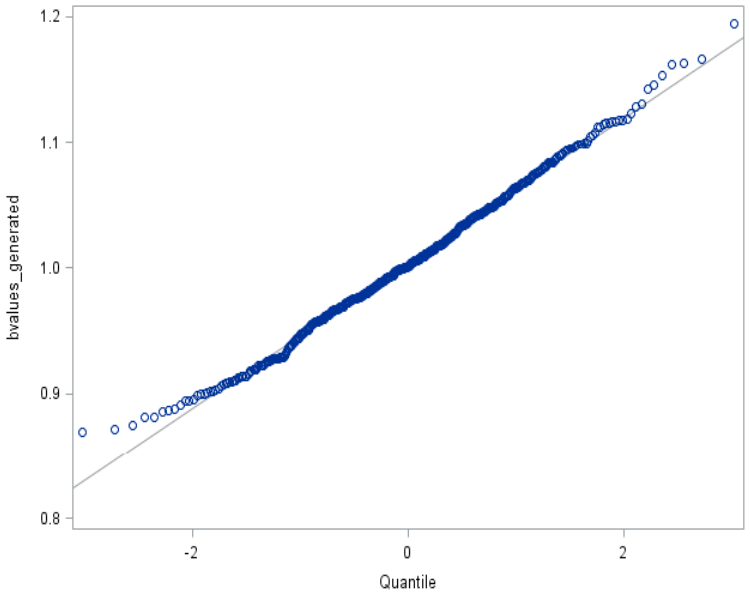


Figure 4: Q-Q Plot based on the b-values.

From Figure 4 we can clearly see that some of the b-values deviate from the 45 deg straight line, indicating that the b-values can not be fitted by a normal distribution as expected.

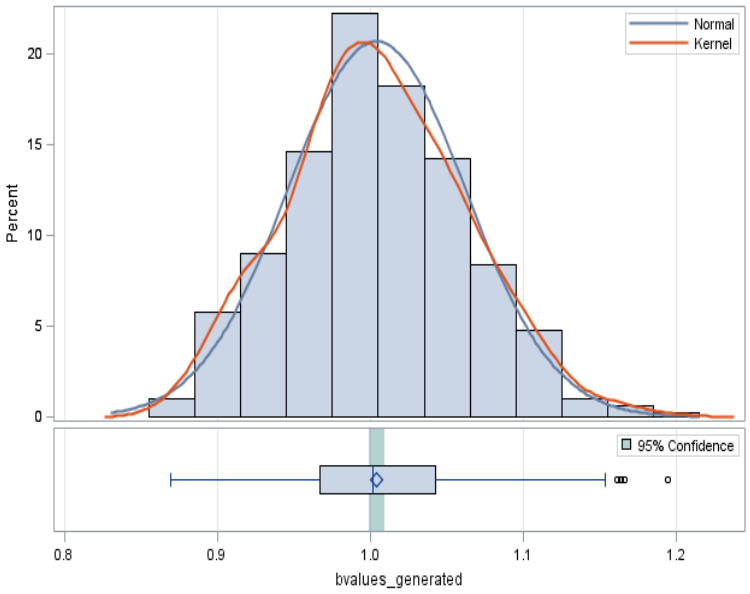


Figure 5: Histogram based on the b-values.

Figure 5 shows that Kernel estimation will solve our data smoothing problem, since the data is gathered over time. It is also a non-parametric way to estimate the probability density function (PDF) [2].

3.1.3 Using other statistical methods for comparison

Our main focus is to compare our b-values generated with the theoretical value of $b = 1$. In order for us to make the best conclusion we need to look at other statistical methods such as observing the mean square error (MSE) and confidence intervals. By using the PROC REG procedure in SAS® we determined the MSE as 0.05792, which is a clear indication that there is almost no difference between the b-values generated and the theoretical value. The mean value of our b-values generated is 1.0040 which is very close to the theoretical value. A 95% confidence interval for the mean, obtained from the PROC TTEST procedure, is equal to $(0.9989, 1.0090)$, which clearly indicates that the mean for the b-values falls within this region, indicating how well the model fits. The R^2value (obtained from PROC REG) is equal to 0.9967, which shows that the model explains almost 100% of the variability of the response data around its mean. Our final measurement is the bias, which is calculated as the mean of the generated magnitude values minus the theoretical value, which is equal to 0.0039517 (obtained from PROC IML). A value of almost zero also concludes that this is indeed an excellent method for obtaining accurate b-values.

3.2 Investigation of the Kijko-Sellevol (1989) method

Similar to the investigation of the generalized Aki-Utsu method we also assume that the hypothetic seismic event catalogue can be divided into two sub-catalogues with level of completeness $m^1_{min} = 4.0$ and $m^2_{min} = 3.0$ respectively [4]. The magnitudes are generated according to the probability distribution function (PDF) of earthquake magnitudes given in equation 1 as before. The simulation was also repeated 500 times for the different number of events.

3.2.1 Generating the $\hat{\beta}$-values

The magnitudes generated for the first level of completeness $m^1_{min} = 4.0$, contains $n_1 = 41$ number of events, and the generated values for the second level of completeness $m^2_{min} = 3.0$, contains $n_2 = 278$ number of events. We can now substitute the necessary values in equation 5 to generate the 500 solutions for the $\hat{\beta}$-value estimator. These values can be shown by the following scatter plot where we compare the values to $\beta = 2.303$:

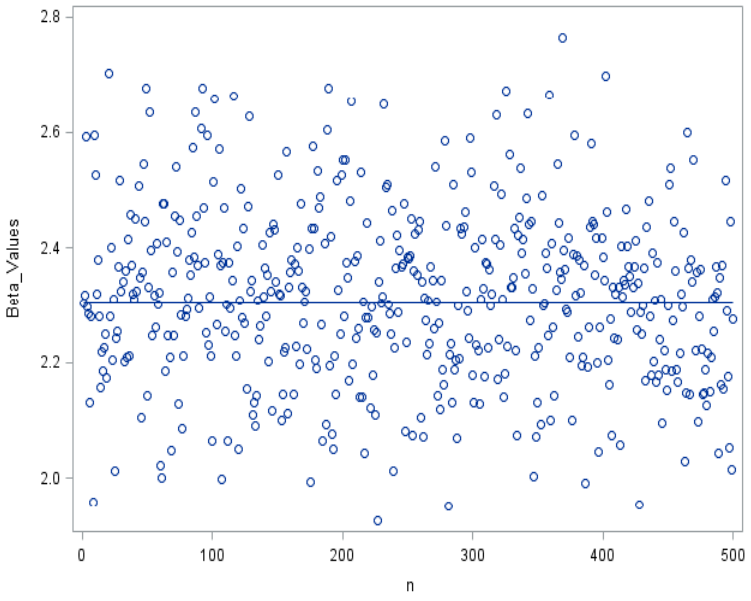


Figure 6: The 500 values generated for the Kijko-Sellevol (1989) $\hat{\beta}$-value estimator.

From Figure 6 we can clearly see that all the values for $\hat{\beta}$ fluctuate around the value $\beta = 2.303$ (demonstrated

by the straight line). By converting the $\hat{\beta}$-values to the b-values we again obtain results which are easier to interpret, which can be seen by the following figure:

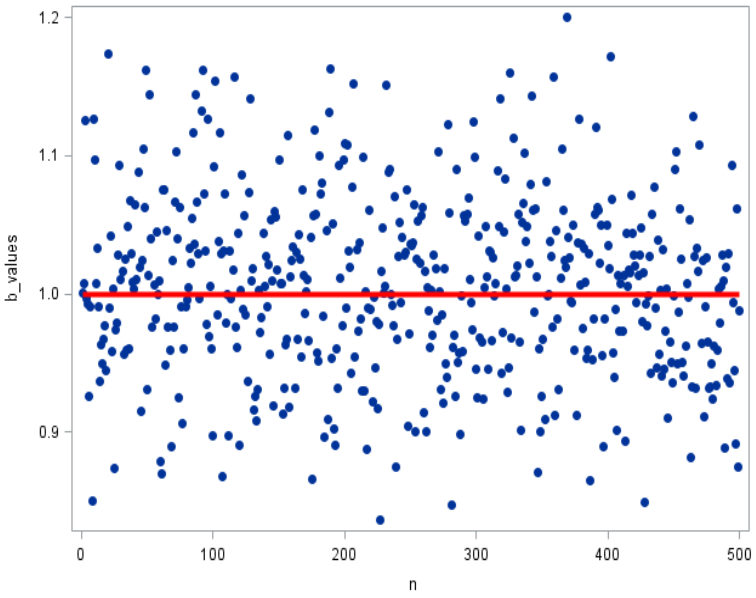


Figure 7: The 500 generated b-values.

3.2.2 Distribution of the b-values

By using the PROC TTEST procedure in SAS, we test the distribution of the b-values based on the Q-Q plot obtained from the output:

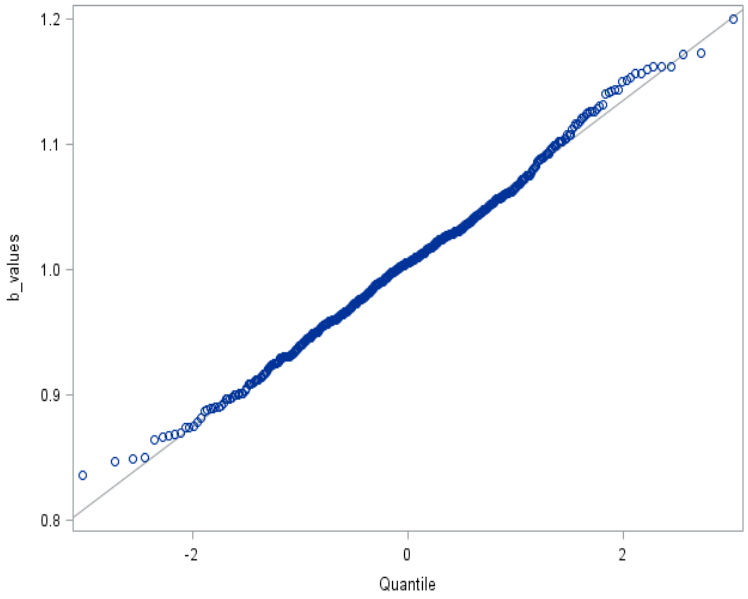


Figure 8: Q-Q Plot based on the b-values.

Similar to the generalized Aki-Utsu method, we can clearly see form Figure 8 that some of the b-values

deviate from the 45 deg straight line, indicating that the b-values can not be fitted by a normal distribution, as expected.

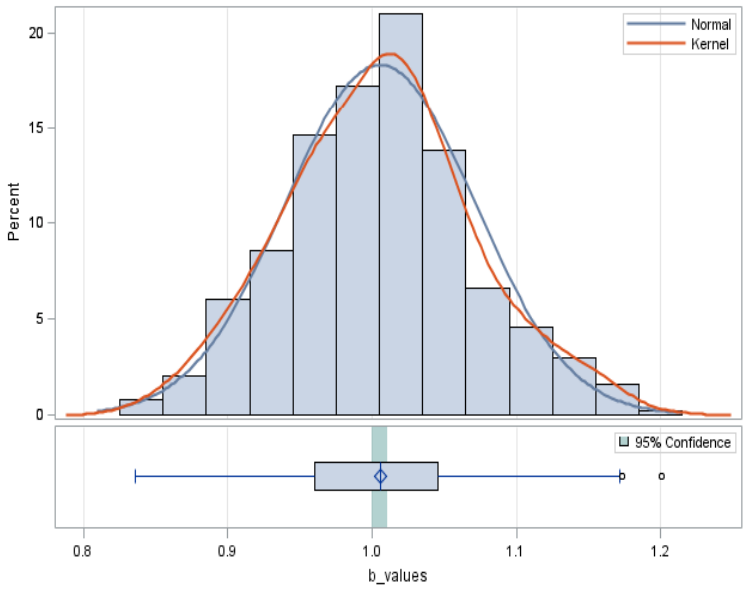


Figure 9: Histogram based on the b-values.

Kernel estimation will again solve our data smoothing problem, as seen by Figure 9.

3.2.3 Using other statistical methods for comparison

In order for us to make the best conclusion we need to look at other statistical methods such as observing the mean square error (MSE) and confidence intervals for the Kijko-Sellevoll (1989) method as well. By using the PROC REG procedure in SAS® we determined the MSE as 0.06528, which is a clear indication that there is almost no difference between the b-values generated and the theoretical value. The mean value of our b-values generated is 1.0054 which is very close to the theoretical value. A 95% confidence interval for the mean, obtained from the PROC TTEST procedure, is equal to $(0.9996, 1.0111)$, which clearly indicates that the mean for the b-values falls within this region, indicating how well the model fits. The R^2value (obtained from PROC REG) is equal to 0.9958, which shows that the model explains almost 100% of the variability of the response data around its mean. Our final measurement is the bias, which is calculated as the mean of the generated magnitude values minus the theoretical value, which is equal to 0.0053648 (obtained from PROC IML). A value of almost zero concludes that this is also a good method for obtaining accurate b-values.

3.3 Comparison between the generalized Aki-Utsu and the Kijko-Sellevoll (1989) method

The following table summarises all the values and statistical conclusions for both the generalized Aki-Utsu and the Kijko-Sellevoll (1989) method, so that a clear comparison can be made:

	The generalized Aki-Utsu method	Kijko-Sellevoll (1989) method
First level of completeness	4.0	4.0
Second level of completeness	3.0	3.0
Number of solutions generated	500	500
Q-Q Plot indication	p-value$< 0.0001 < 0.05$	p-value$< 0.0001 < 0.05$
Decision	Reject Normality (as expected)	Reject Normality (as expected)
MSE (Mean Square Error)	0.05792	0.06528
Bias	0.0039517	0.0053648
R^2	0.9967	0.9958
Mean value of the b-values generated	1.0040	1.0054
95% confidence interval for the mean	$(0.9989, 1.0090)$	$(0.9996, 1.0111)$

Table 1: Comparison between the generalized Aki-Utsu method and the Kijko-Sellevoll (1989) method

By using the PROC BOXPLOT procedure in SAS® we obtained a boxplot for each method, (as shown in Figure 10 and Figure 11). This is also a clear indication that the generalized Aki-Utsu method obtain the most accurate values for b, since the distance between the minimum and the first-quartile, as well as the distance between the third-quartile and the maximum, are smaller than those of the Kijko-Sellevoll (1989) method.

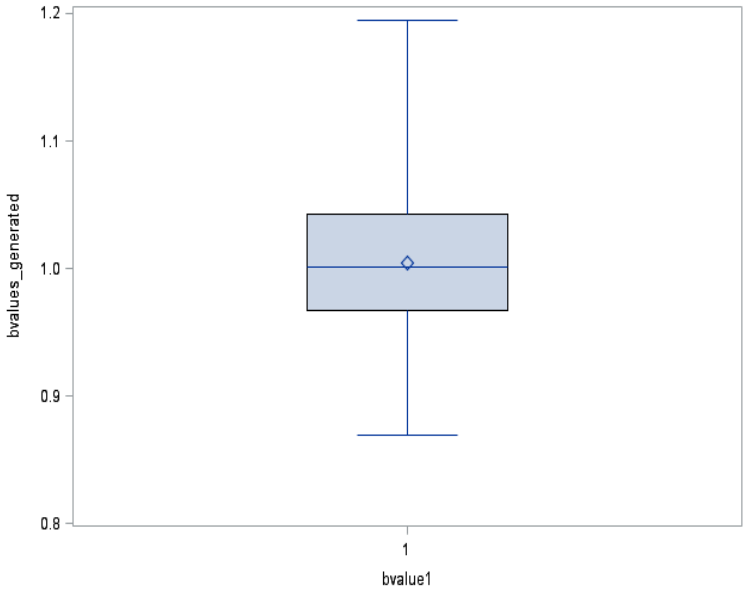


Figure 10: Boxplot for the generalized Aki-Utsu method.

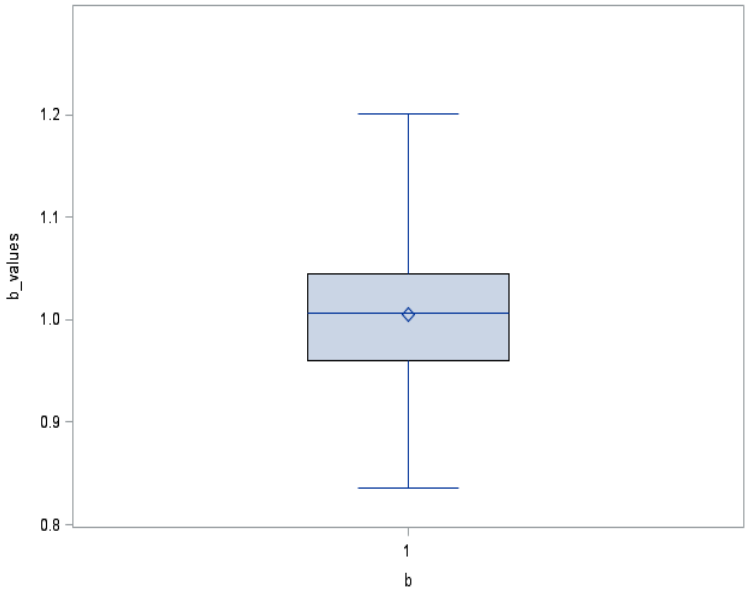


Figure 11: Boxplot for the Kijko-Sellevoll (1989) method

4 Conclusion

This research report investigated how we can identify the best maximum likelihood estimate for the Gutenberg-Richter b-value through theoretical derivations and application, given the frequency-magnitude Gutenberg-Richter earthquake distribution. We considered the generalized Aki-Utsu $\hat{\beta}$-value estimator which measures different levels of completeness of multiple catalogues. This model was derived using the method of maximum likelihood estimation where we then applied it to a hypothetic event catalogue. We found that this method yields a very low bias which showed that there is almost no difference between the mean of the magnitude values generated and the theoretical value. It was also shown that the model almost fits perfectly according to the high value of R^2, where we then verified this conclusion with the very low value of the mean square error.

Thereafter we considered the the Kijko-Sellevoll (1989) method, which is known as a standard method only used for the complete younger parts of the catalogue. This model was again derived using the method of maximum likelihood estimation to obtain the equation for $\hat{\beta}$ which we also applied to the same hypothetic event catalogue as before. We then found that this method yields a very low bias but slightly higher than that of the generalized Aki-Utsu method. With a high value of R^2 we could also see this model fits well, however the generalized Aki-Utsu method with a slightly higher value will be preferred. Lastly we also calculated the mean square error which was indeed higher than that of the generalized Aki-Utsu method.

We can therefore conclude that the generalized Aki-Utsu method, proposed by Kijko-Smit (2012) , is indeed an excellent method for obtaining accurate b-values. Future research can be done to show that maximum likelihood estimation is not the only appropriate technique, since there is no evidence suggesting that method of moments estimation (in short, MME) cannot be used.

References

[1] Keiiti Aki. Maximum likelihood estimate of b in the formula $logn = a - bm$ and its confidence limits. *Bulletin of the Earthquake Research Institute*, 43:237–239, 1965.

[2] Theo Gasser and Hans-Georg Müller. Kernel estimation of regression functions. In *Smoothing techniques for curve estimation*, pages 23–68. Springer, 1979.

[3] A Kijko and MA Sellevoll. Estimation of earthquake hazard parameters from incomplete data files. part i. utilization of extreme and complete catalogs with different threshold magnitudes. *Bulletin of the Seismological Society of America*, 79(3):645–654, 1989.

[4] Andrzej Kijko and Ansie Smit. Extension of the Aki-Utsu b-value estimator for incomplete catalogs. *Bulletin of the Seismological Society of America*, 102(3):1283–1287, 2012.

[5] Sankaran Mahadevan. Monte Carlo simulation. *Mechanical Engineering-New York and Basel-Marcel Dekker*, pages 123–146, 1997.

[6] Warner Marzocchi and Laura Sandri. A review and new insights on the estimation of the b-valueand its uncertainty. *Annals of Geophysics*, 2003.

[7] Lev A Maslov and Vladimir M Anokhin. Derivation of the gutenberg-richter empirical formula from the solution of the generalized logistic equation. *Natural Science*, 4(28):648–651, 2012.

[8] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90–100, 2003.

[9] Dieter H Weichert. Estimation of the earthquake recurrence parameters for unequal observation periods for different magnitudes. *Bulletin of the Seismological Society of America*, 70(4):1337–1346, 1980.

Appendix A

```
*/the generalized Aki-Utsu method*/;
proc iml;
mmin1=4.0;
mmin2=3.0;
beta=2.303;
b1=1;
T1=39.997;*/time span*/;
T2=29.996;
n1=41;
n2=278;
r1=n1/(n1+n2);
r2=n2/(n1+n2);
do i=1 to 500;
*Monte Carlo Simulation for mmin=4.0*;
seed=j(n1,1,1);
u1=uniform(seed)+1;
mc1=u1[1:n1,1];
lg1=log(mc1-1);
x1=(-1*lg1/beta)+mmin1;
sumx1=x1[+];
mean1=x1[:,];
bhat1=1/(mean1-mmin1);
*Monte Carlo Simulation for mmin=3.0*;
seed=j(n2,1,1);
u2=uniform(seed)+1;
mc2=u2[1:n2,1];
lg2=log(mc2-1);
x2=(-1*lg2/beta)+mmin2;
sumx2=x2[+];
mean2=x2[:,];
bhat2=1/(mean2-mmin2);
bhat=1/((r1/bhat1)+(r2/bhat2)); *Calculation for Betahat*;
betavalue=J(i,1,beta);
bvalue=bhat/log(10);
bvalues = bvalues // bvalue;
b=J(i,1,1);
totaln= totaln // i;
allbhats = allbhats // bhat;
print totaln allbhats betavalue bvalues b;
end;
meanb=bvalues[:,];
bias=meanb-b1;
print bias;
nm={"n" "Beta_Values" "bvalue" "bvalues_generated" "bvalue1"};
nbeta = totaln || allbhats || betavalue || bvalues || b;
create bvalues from nbeta[colname=nm];
append from nbeta;
quit;
*Creating Histogram for 500 Beta values*;
proc template;
define statgraph sgdesign;
```

```
dynamic _BETA_VALUES;
begingraph;
entrytitle halign=center 'Beta Values for the generalized Aki-Utsu method';
layout lattice / rowdatarange=data columndatarange=data
rowgutter=10 columngutter=10;
layout overlay;
histogram _BETA_VALUES / name='histogram' binaxis=false;
endlayout;
endlayout;
endgraph;
end;
run;
proc sgrender data=WORK.BVALUES template=sgdesign;
dynamic _BETA_VALUES="'BETA_VALUES'n";
run;
*Histogram code end*;
*Creating Scatterplot for 500 Beta values and comparing to Beta=2.303*;
proc template;
define statgraph Graph;
dynamic _N _BETA_VALUES _N2 _BVALUE;
begingraph;
entrytitle halign=center 'Beta Values for the generalized Aki-Utsu method';
layout lattice / rowdatarange=data columndatarange=data
rowgutter=10 columngutter=10;
layout overlay;
scatterplot x=_N y=_BETA_VALUES / name='scatter'
markerattrs=(color=CX5A518C );
seriesplot x=_N2 y=_BVALUE / name='series' connectorder=xaxis;
endlayout;
endlayout;
endgraph;
end;
run;
proc sgrender data=WORK.BVALUES template=Graph;
dynamic _N="N" _BETA_VALUES="'BETA_VALUES'n" _N2="N" _BVALUE="BVALUE";
run;
*Scatterplot code end*;
proc ttest data=bvalues;
var bvalues_generated;
run;
proc reg data=bvalues;
model bvalues_generated=bvalue1/ noint;
run;
*Scatterplot of bvalues;
proc template;
define statgraph sgdesign;
dynamic _N _BVALUES_GENERATED _N2 _BVALUE1A;
begingraph;
entrytitle halign=center 'The generalized Aki-Utsu method: b-values generated
compared to b=1 ';
layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
layout overlay;
```

```
scatterplot x=_N y=_BVALUES_GENERATED / name='scatter'
markerattrs=(symbol=CIRCLEFILLED size=7 );
seriesplot x=_N2 y=_BVALUE1A / name='series' connectorder=xaxis
lineattrs=(color=CXFF0000 thickness=4 );
endlayout;
endlayout;
endgraph;
end;
run;
proc sgrender data=WORK.BVALUES template=sgdesign;
dynamic _N="N" _BVALUES_GENERATED="'BVALUES_GENERATED'n"
_N2="N" _BVALUE1A="BVALUE1";
run;
*Scatterplot code end*;
proc boxplot data=bvalues;
plot bvalues_generated*bvalue1;
run;

*/the Kijko-Sellevoll (1989) method*/;
proc iml;
n1=41;
n2=200;
ntotal=n1+n2;
T1=39.997;*/time span*/;
T2=29.996;
T=T1+T2;
mmin1=4.0;
mmin2=3.0;
beta=2.303;
b1=1;
do i=1 to 500;
*Monte Carlo simulation for first LOC*;
seed=j(n1,1,1);
u1=uniform(seed)+1;
mc1=u1[1:n1,1];
lg1=log(mc1-1);
x1=(-1*lg1/beta)+mmin1;
sumx1=x1[+];
mean1=x1[:,];
*determine magnitudes greater than loc1;
lci=J(n1,1,mmin1);
cnt1=(x1>=lci)[+] || (x1<lci)[+];
class1=cnt1[<:>];
res1 = res1 // class1;
*print cnt1 res1;
*Monte Carlo simulation for second LOC*;
seed=j(n2,1,1);
u2=uniform(seed)+1;
mc2=u2[1:n2,1];
lg2=log(mc2-1);
x2=(-1*lg2/beta)+mmin2;
*determine magnitudes greater than loc2;
lcii=J(n2,1,mmin2);
```

```
cnt2=(x2>=lcii)[+] || (x2<lcii)[+];
class2=cnt2[<:>];
res2 = res2 // class2;
*print cnt2 res2;
*Calculations for bhat;
sumx=x1-mmin2;
d=sumx[+];
sumxx=x2-mmin2;
d1=sumxx[+];
totalsum=d+d1;
su2=(n1*(mmin1-mmin2));
betahat1 = ntotal/(totalsum-su2); *Calculating betahat*;
bvalue1=betahat1/log(10);
betav=J(i,1,beta);
bv=J(i,1,1);
n= n // i;
abhats1 = abhats1 // betahat1;
bvalues1 = bvalues1 // bvalue1;
print n abhats1 betav bvalues1 bv;
end;
meanb=bvalues1[:,];
bias=meanb-b1;
print bias;
nm={"n" "Beta_Values" "Beta" "b_values" "b"};
nbeta1= n || abhats1 || betav || bvalues1 || bv;
create betavalues from nbeta1[colname=nm];
append from nbeta1;
quit;
*end of 500 bhats generated;
*Histogram for the beta values comparing to beta=2.303;
proc template;
define statgraph sgdesign;
dynamic _BETA_VALUES;
begingraph;
entrytitle halign=center 'Beta values for Kijko-Sellevol (1989) method';
layout lattice / rowdatarange=data columndatarange=data
rowgutter=10 columngutter=10;
layout overlay;
histogram _BETA_VALUES / name='histogram' binaxis=false;
endlayout;
endlayout;
endgraph;
end;
run;
proc sgrender data=WORK.BVALUES template=sgdesign;
dynamic _BETA_VALUES="'BETA_VALUES'n";
run;
*Scatterplot for the beta values comparing to beta=2.303;
proc template;
define statgraph sgdesign;
dynamic _N _BETA_VALUES _N2 _BETA;
begingraph;
entrytitle halign=center 'Beta values for the Kijko-Sellevol (1989) method';
```

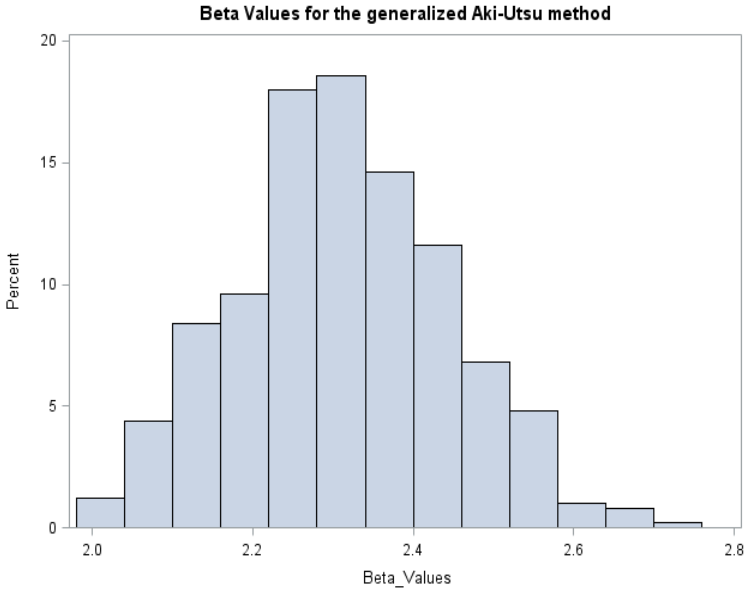
```
layout lattice / rowdatarange=data columndatarange=data
rowgutter=10 columngutter=10;
layout overlay;
scatterplot x=_N y=_BETA_VALUES / name='scatter';
seriesplot x=_N2 y=_BETA / name='series' connectorder=xaxis;
endlayout;
endlayout;
endgraph;
end;
run;
proc sgrender data=WORK.BETAVALUES template=sgdesign;
dynamic _N="N" _BETA_VALUES="'BETA_VALUES'n" _N2="N" _BETA="BETA";
run;
proc ttest data=betavalues;
var b_values;
run;
proc reg data=betavalues;*Calculating the mean square error and r-square;
model b_values=b/ noint;
run;
*Scatterplot for the b values comparing to b=1*;
proc template;
define statgraph Graph;
dynamic _N _B_VALUES _N2 _B;
begingraph;
entrytitle halign=center 'Kijko-Sellevoll (1989) method: b-values
generated compared to b=1';
layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
layout overlay;
scatterplot x=_N y=_B_VALUES / name='scatter'
markerattrs=(symbol=CIRCLEFILLED size=7 );
seriesplot x=_N2 y=_B / name='series' connectorder=xaxis
lineattrs=(color=CXFF0000 thickness=4 );
endlayout;
endlayout;
endgraph;
end;
run;
proc sgrender data=WORK.BETAVALUES template=Graph;
dynamic _N="N" _B_VALUES="'B_VALUES'n" _N2="N" _B="B";
run;
*Scatterplot code ends*;
proc boxplot data=betavalues;
plot b_values*b;
run;
```

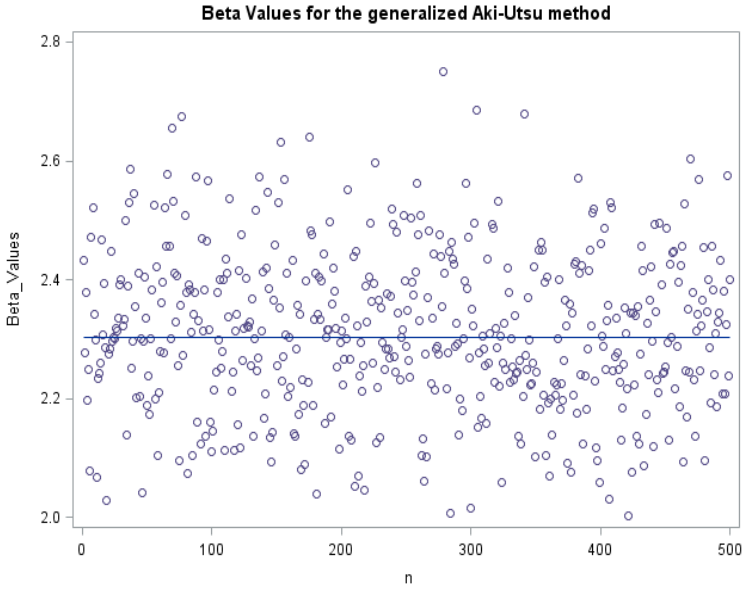
Appendix B

totaln	allbhats	betavalue	bvalues	b
1	2.4317966	2.303	1.0561159	1
2	2.2777041	2.303	0.9891943	1
3	2.3783915	2.303	1.0329223	1
4	2.1977033	2.303	0.9544504	1
5	2.2491893	2.303	0.9768105	1
6	2.0779641	2.303	0.9024483	1
7	2.4714176	2.303	1.073323	1
8	2.5215526	2.303	1.0950964	1
9	2.3414807	2.303	1.0168921	1
10	2.297756	2.303	0.9979027	1
11	2.067803	2.303	0.8980354	1
12	2.2347216	2.303	0.9705273	1
13	2.2432773	2.303	0.974243	1
14	2.2600792	2.303	0.9815399	1
15	2.4667031	2.303	1.0712756	1
16	2.3081675	2.303	1.0024244	1
17	2.3929444	2.303	1.0392426	1
18	2.2862878	2.303	0.9929222	1
19	2.0294102	2.303	0.8813616	1
20	2.2756469	2.303	0.9883009	1
21	2.2825217	2.303	0.9912866	1

(Continues till totaln=500)

bias
0.0039517





The SAS System

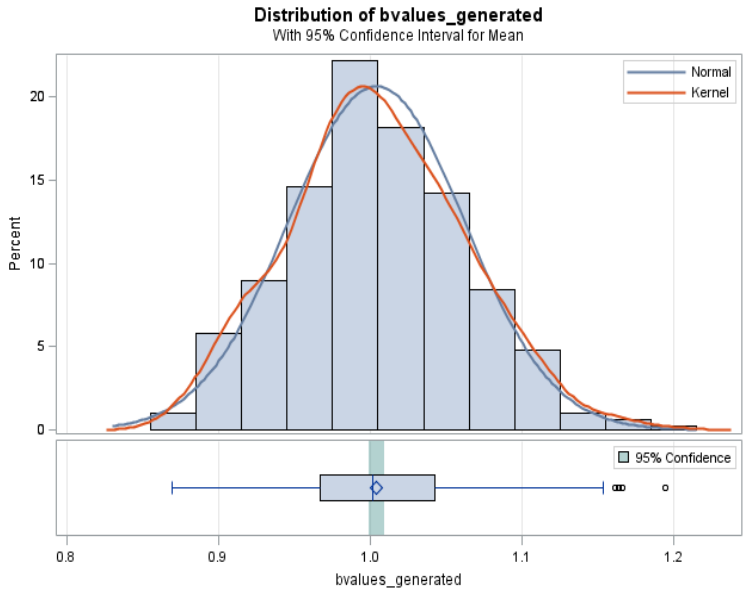
The TTEST Procedure

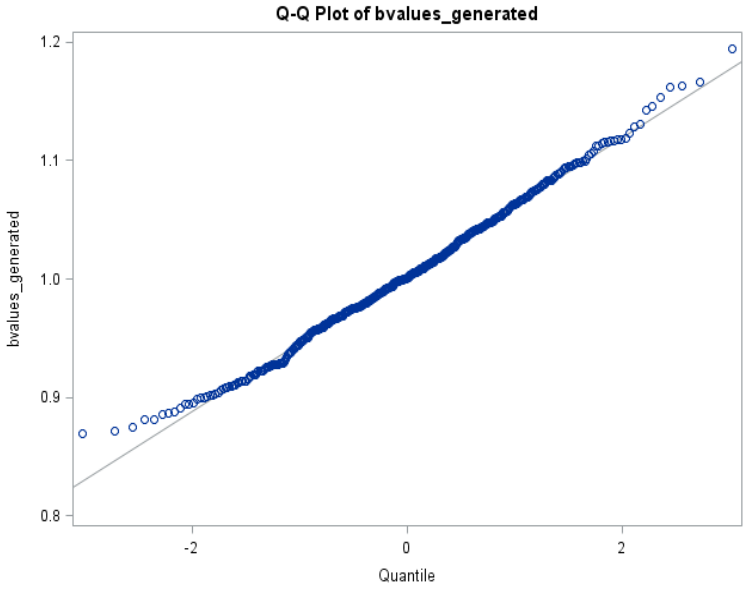
Variable: bvalues_generated

N	Mean	Std Dev	Std Err	Minimum	Maximum
500	1.0040	0.0579	0.00259	0.8693	1.1945

Mean	95% CL Mean		Std Dev	95% CL Std Dev	
1.0040	0.9989	1.0090	0.0579	0.0545	0.0618

DF	t Value	Pr > |t|
499	387.57	<.0001





The SAS System

The REG Procedure
Model: MODEL1
Dependent Variable: bvalues_generated

Number of Observations Read	500
Number of Observations Used	500

Note: No intercept in model. R-Square is redefined.

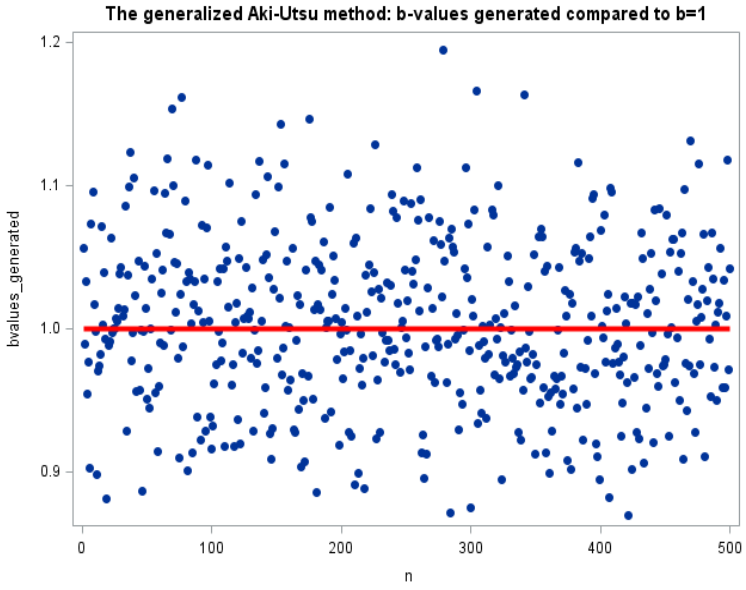
Analysis of Variance

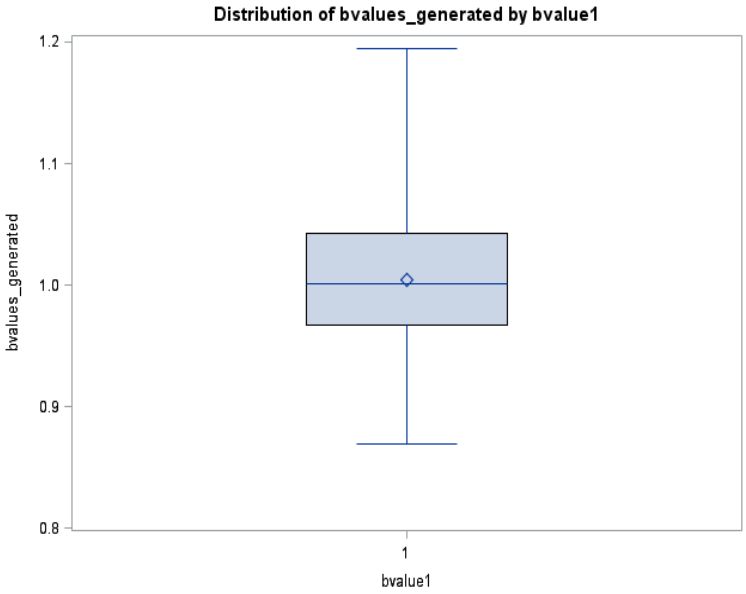
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	503.95952	503.95952	150208	<.0001
Error	499	1.67419	0.00336		
Uncorrected Total	500	505.63371			

Root MSE	0.05792	R-Square	0.9967
Dependent Mean	1.00395	Adj R-Sq	0.9967
Coeff Var	5.76951		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
bvalue1	1	1.00395	0.00259	387.57	<.0001


The generalized Aki-Utsu method: b-values generated compared to b=1
bvalues_generated
1.2
1.1
1.0
0.9
0
100
200
300
400
500
n

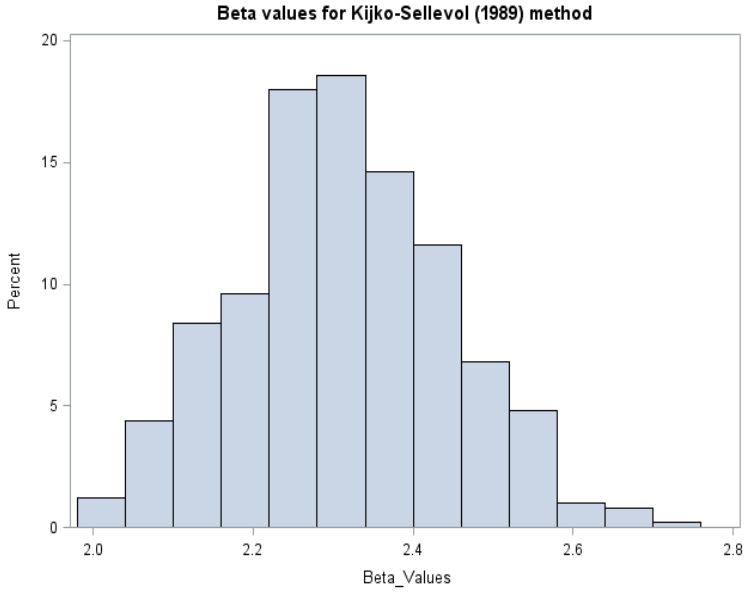

Distribution of bvalues_generated by bvalue1
bvalues_generated
1.2
1.1
1.0
0.9
0.8
1
bvalue1

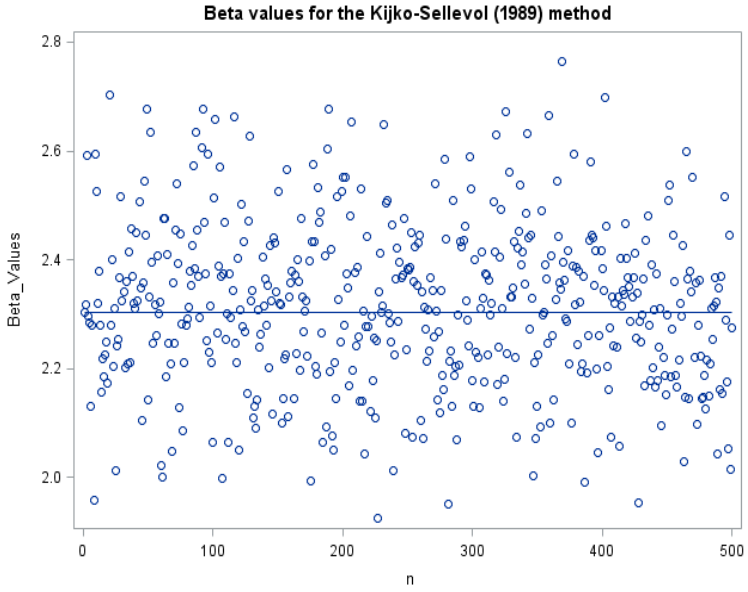
The Kijko-Sellevoll (1989) method:

n	abhats1	betav	bvalues1	bv
1	2.303513	2.303	1.000403	1
2	2.3185749	2.303	1.0069443	1
3	2.5916182	2.303	1.1255255	1
4	2.2957469	2.303	0.9970302	1
5	2.2846739	2.303	0.9922213	1
6	2.1308524	2.303	0.9254174	1
7	2.2805003	2.303	0.9904087	1
8	1.9574488	2.303	0.8501092	1
9	2.594454	2.303	1.1267571	1
10	2.5249975	2.303	1.0965925	1
11	2.3203777	2.303	1.0077272	1
12	2.3786638	2.303	1.0330406	1
13	2.2807824	2.303	0.9905312	1
14	2.1569018	2.303	0.9367306	1
15	2.2180031	2.303	0.9632665	1
16	2.1863184	2.303	0.949506	1
17	2.2264292	2.303	0.9669259	1
18	2.248441	2.303	0.9764855	1
19	2.1735153	2.303	0.9439457	1
20	2.7021547	2.303	1.1735309	1

(Continues till n=500)

bias
0.0053648





The SAS System

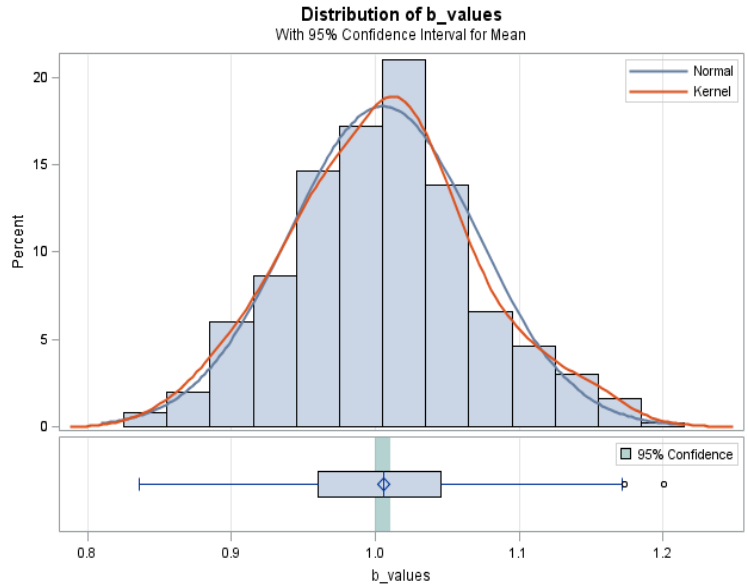
The TTEST Procedure

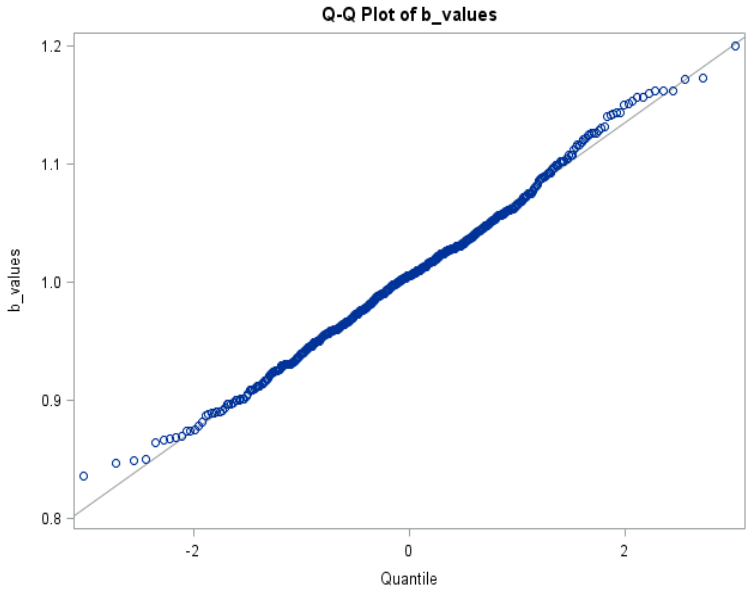
Variable: b_values

N	Mean	Std Dev	Std Err	Minimum	Maximum
500	1.0054	0.0653	0.00292	0.8360	1.2003

Mean	95% CL Mean		Std Dev	95% CL Std Dev	
1.0054	0.9996	1.0111	0.0653	0.0615	0.0696

DF	t Value	Pr > |t|
499	344.38	<.0001





The SAS System

The REG Procedure
Model: MODEL1
Dependent Variable: b_values

Number of Observations Read	500
Number of Observations Used	500

Note: No intercept in model. R-Square is redefined.

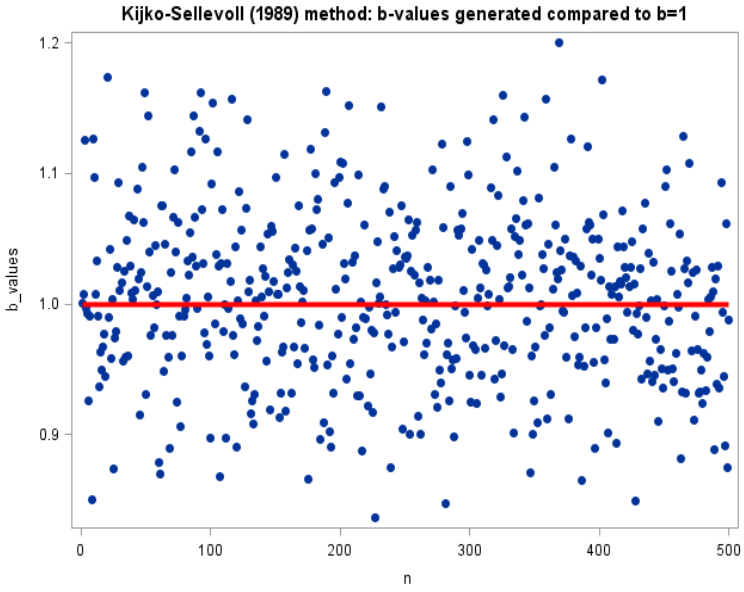
Analysis of Variance

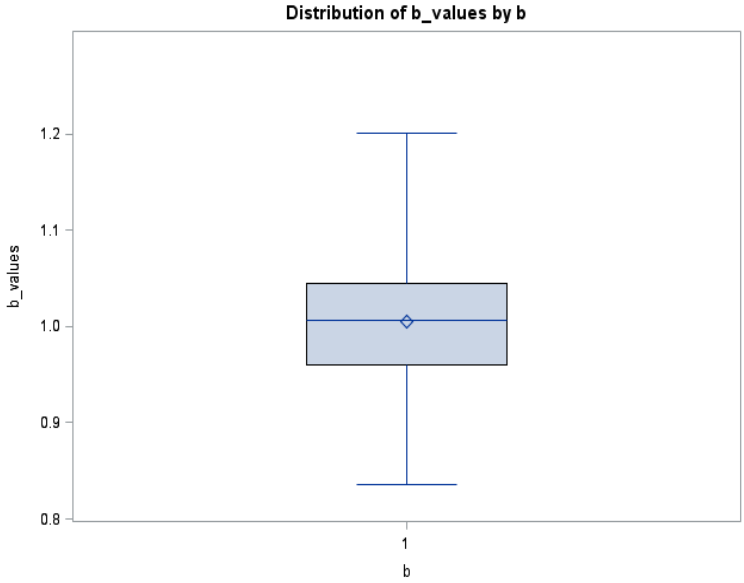
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	505.37924	505.37924	118598	<.0001
Error	499	2.12637	0.00426		
Uncorrected Total	500	507.50561			

Root MSE	0.06528	R-Square	0.9958
Dependent Mean	1.00536	Adj R-Sq	0.9958
Coeff Var	6.49300		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
b	1	1.00536	0.00292	344.38	<.0001


Kijko-Sellevoll (1989) method: b-values generated compared to b=1
b_values
1.2
1.1
1.0
0.9
0
100
200
300
400
500
n


Distribution of b_values by b
b_values
1.2
1.1
1.0
0.9
0.8
1
b

Estimation of the b-value on the Gutenberg-Richter earthquake distribution using maximum likelihood estimators and method of moments estimators

Geervana Bye Rugjee 12013677

WST795 Research Report

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

Supervisor: Mr MT Loots, Co-supervisors: Prof. A Kijko, Ms A Smit

Department of Statistics, University of Pretoria

2 November 2016

Abstract

Various estimators for the b-value of the Gutenberg-Richter earthquake relation have been derived in previous literature [8, 14, 6].The identification of the best estimator of the b-value estimator is examined in the report. Two models are discussed where their properties are analysed and the estimators are derived which are ultimately rated. The first model uses maximum likelihood estimation which is still the preferred estimator as discussed in [2] and the second model is based on method of moments. Monte Carlo is used for simulations and the statistical methods used to analyse the estimators include mean squared error and bias.

Declaration

I, *Geervana Bye Rugjee,* 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.

_____
Geervana Bye Rugjee

_____
MT Loots

_____
A Kijko

_____
A Smit

_____
Date

Acknowledgments

I would like to acknowledge funding received from the National Research Foundation(NRF), which has helped financially with my honours studies.
I also would like to express my appreciation to my supervisor, Prof Kijko, who has offered guidance through his knowledge and expertise for the completion of this report. Lastly, I would like to thank Mr MT Loots and Ms A Smit who have been a source of inspiration and support throughout this journey.

Contents

1 Introduction **6**

2 Literature Review **7**

3 Background Theory **7**
3.1 Mathematical Background 7
3.1.1 Weichert(1980) 7
3.1.2 Kijko-Smit(2016) 9

4 Application **11**
4.1 Weichert(1980) 11
4.2 Kijko-Smit(2016) 12

5 Conclusion **13**

Appendix **31**

List of Figures

1 An illustration of a seismic event catalogue with different time periods and levels of completeness as described in [14]. m_{min} are the levels of completeness and $\triangle m$ is the difference on level of completeness between two time periods. 7
2 Histogram with an hypothetical seismic catalogue with the apparent frequency distribution and the Gutenberg-Richter frequency-magnitude distribution law. 9
3 A line plot of the b-values generated from Weichert(1980)[14] 11
4 A histogram of the b-values from Weichert(1980)[14] 12
5 A line plot of the b-values generated from Kijko-Smit(2016)[7] 12
6 A histogram of the b-values from Kijko-Smit(2016)[7] 13

List of Tables

1 Table comparing results from the two methods. 13

1 Introduction

An important issue of seismological studies is to assess the hazard involved with the event of an earthquake. One of the most important equations in seismology, The Gutenberg-Richter magnitude relation [14] is expressed as

$$\log(N) = a - bm \tag{1}$$

where N is the number of earthquakes with a magnitude of m or greater, a is the seismic level and b provides the relationship between large and small events. This equation is of important value since it helps to analyse tectonic and induced seismic activities and is applicable for unequal time periods, a crucial feature in seismology. The b-value, a characteristic of the seismic rate, is obtained by plotting the number of earthquakes against the magnitudes. To estimate the b-value, the relationship $\beta = b \ln 10$ is used. The$b$-value tends to be close to 1 in seismically active places. Estimating both a and b is of high importance since they are used in earthquake prediction as well as hazard assessment, etc.The magnitudes follow an exponential distribution with parameter α while the frequency of the earthquakes follow a Poisson distribution with parameter λ, which is used to estimate parameter a in equation 1. This paper will specifically look at methods of estimating the seismic activity rate parameter (parameter b from equation 1). If it is assumed that the magnitudes are independent, identically distributed random variables following equation 1 then the probability density function of the magnitudes, from [7], is

$$f(m;\beta) = \begin{cases} 0 & m \leq m_{min} \\ \beta \exp[-\beta(m - m_{min})] & m \geq m_{min} \end{cases} \tag{2}$$

$$F(m,\beta) = \begin{cases} 0 & m \leq m_{min} \\ -\exp[\beta(m - m_{min})] + 1 & m \geq m_{min} \end{cases} \tag{3}$$

and the preferred estimator of the b-value, which was derived by [2] is

$$\hat{\beta} = \frac{1}{\bar{m} - m_{min}}. \tag{4}$$

where $\bar{m}$ = average magnitude and m_{min} = lowest magnitude in complete catalogues observations (also called level of completeness) and m_{max} =maximum magnitude over the catalogue observations. Equation (4) was derived first by [12] where he used the method of moments and [2] used the maximum likelihood method instead. There are four methodologies in previous literature from [8, 14, 6, 7], where different and reliable estimates for β are given, but this paper will assess the accuracy of the estimation of the b-value by considering only two methodologies namely the first from [14] :

$$\frac{1}{\hat{\beta}} = \bar{m} - m_{min} - \frac{(m_{max})\exp(-\beta(m_{max} - m_{min}))}{1 - \exp(-\beta(m_{max} - m_{min}))} \tag{5}$$

and the second from [7] :

$$\hat{\beta} = \frac{(2\bar{m}_2)}{\bar{m}_3} \tag{6}$$

where $\bar{m}_2 = \frac{\sum_{i=1}^{n}(X_i - \bar{m})^2}{n}$ and $\bar{m}_3 = \frac{\sum_{i=1}^{n}(X_i - \bar{m})^3}{n}$.

To test these two methods, the Monte Carlo technique is used to simulate the values and their mean square errors and their bias are calculated and assessed respectively. A conclusion is given on the better estimator of the b-value and of its reliability thereof.

2 Literature Review

The main references used in this project are 'Estimation of the earthquake recurrence parameters for unequal observation periods for different magnitudes by Weichert from [14]. In this article, the MLE of the earthquake parameters are taken in the case of events with equal magnitudes but over individual time periods. Next is 'Maximum likelihood estimate of b In the Formula $logn = a - bm$ And its confidence limits' by Aki from [2] where Aki[2] derives an estimator for the b-value, which is used extensively for estimation due to its simplicity.'Estimation of the frequency-magnitude Gutenberg-Richter b-value without making assumptions on levels of completeness' from [7] is also consulted. In this paper, both the method of moments and the maximum likelihood estimation are used to derive estimators for the b-value when the level of completeness is unknown. Lastly,'A method for determining b-value in the formula $logn = a - bm$ showing the magnitude-frequency relation for earthquakes' from [12] in which an expression is derived by [12] to correct for bias due to rounding when estimating the b-value.

3 Background Theory

3.1 Mathematical Background

This report will be based on the assumption of rejecting the macro seismic observations which are incomplete and only consider the complete part of the catalogues as discussed in [6]. Since it is assumed that the seismic events are independent and identically distributed, the likelihood functions are derived by taking the products of the likelihood of each sub catalogue. Then the MLE is calculated by maximizing the likelihood function. For the method of moments, the sample moments are equated to the population moments and we solve the equations to obtain an estimator.

3.1.1 Weichert(1980)

Our assumptions when dealing with this methodology are that the events are grouped by magnitudes where each group has individual time periods. Weichert [14] assumes, very realistically, unequal observations periods for different magnitudes. Figure 1 shows this assumption as the shaded block, which represents the difference between the first and second levels of completeness. Another assumption in [14] is that a maximum magnitude m_{max} is imposed. Since the chance of possible variability exists because of unequal observations, the magnitudes are assumed to be independent identically random variables.

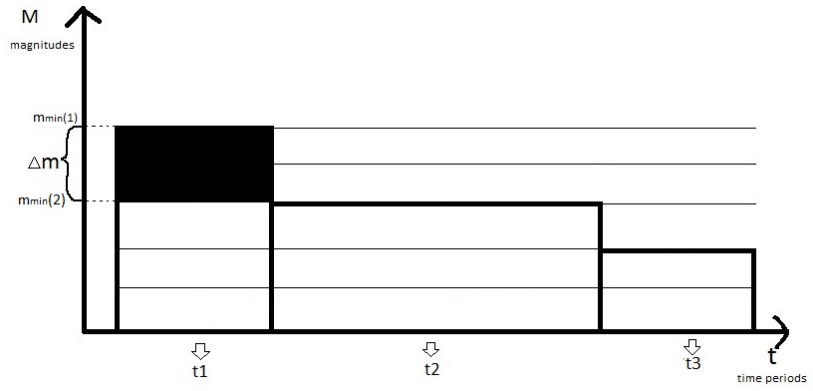


Figure 1: An illustration of a seismic event catalogue with different time periods and levels of completeness as described in [14]. m_{min} are the levels of completeness and $\triangle m$ is the difference on level of completeness between two time periods.

From [10], the frequency distribution $n(x)$ is given as $n(x) = Kx^{-\beta}$ where we need to find the value of β that maximises the probability of getting n samples of x in the interval $[x_{min}, x_{max}]$. So taking ln, we get

$$ln(n(x)) = ln(K) - \beta ln(x).$$

This follows the Gutenberg-Richter equation (1) if we let $ln(x) = m$. Now the probability that the ith sample will fall between $x_i$and $x_i + dx$ is $f(x_i)dx$, where $f(x_i)$ is given by:

$$f_X(x_i;\beta) = \frac{(1-\beta)x_i^{-\beta}}{(x_{max})^{1-\beta} - (x_{min})^{1-\beta}} \tag{7}$$

The likelihood function defined as the jointed density which is the product of the marginal densities from equation(7), becomes

$$\begin{aligned} L(\beta|x) &= \prod_{i=1}^{n} f_X(x_i;\beta) \\ &= \frac{\prod_{i=1}^{n}(1-\beta)x_i^{-\beta}}{(x_{max})^{\sum_{i=1}^{n}(1-\beta)} - (x_{min})^{\sum_{i=1}^{n}(1-\beta)}} \\ &= \frac{n(1-\beta)\sum_{i=1}^{n} x_i^{-\beta}}{n((x_{max})^{(1-\beta)} - (x_{min})^{(1-\beta)})} \end{aligned}$$

The log-likehood function becomes

$$\ln L(\beta|x) = n\ln(1-\beta) + \sum_{i=1}^{n} -\beta\ln(x_i) - \ln[n((x_{max})^{1-\beta} - (x_{min})^{1-\beta})].$$

Differentiating with respect to β gives

$$\begin{aligned} \frac{\delta \ln L(\beta|x)}{\delta\beta} &= \frac{n}{1-\beta}(-1) - \sum_{i=1}^{n}\ln(x_i) - \\ &\quad \frac{1}{(x_{max})^{1-\beta} - (x_{min})^{1-\beta}}(x_{max})^{1-\beta}\ln(x_{max})(-1) + (x_{min})^{1-\beta}\ln(x_{min})(-1) \\ &= \frac{-n}{1-\beta} - \sum_{i=1}^{n}\ln(x_i) + \frac{(x_{max})^{1-\beta}\ln(x_{max}) - (x_{min})^{1-\beta}\ln(x_{min})}{(x_{max})^{1-\beta} - (x_{min})^{1-\beta}}. \end{aligned}$$

Setting equal to zero gives

$$\frac{-n}{1-\beta} - \sum_{i=1}^{n}\ln(x_i) + \frac{(x_{max})^{1-\beta}\ln(x_{max}) - (x_{min})^{1-\beta}\ln(x_{min})}{(x_{max})^{1-\beta} - (x_{min})^{1-\beta}} = 0.$$

$$\begin{aligned} \sum_{i=1}^{n}\ln(x_i) &= \frac{-n}{1-\beta} + \frac{(x_{max})^{1-\beta}\ln(x_{max}) - (x_{min})^{1-\beta}\ln(x_{min})}{(m_{max})^{1-\beta} - (m_{min})^{1-\beta}}, \\ \ln(\bar{x_i}) &= \frac{1}{\beta-1} + \frac{(x_{max})^{1-\beta}\ln(x_{max}) - (x_{min})^{1-\beta}\ln(x_{min})}{(x_{max})^{1-\beta} - (x_{min})^{1-\beta}}. \end{aligned}$$

but since $ln(x) = m$, we can solve for $\hat{\beta}$ as

$$\frac{1}{\hat{\beta}} = \bar{m} - m_{min} - \frac{m_{max}\exp(-\beta(m_{max}-m_{min}))}{1-\exp(-\beta(m_{max}-m_{min}))}.$$

3.1.2 Kijko-Smit(2016)

In [7], the problem of heavy dependence of the level of completeness m_{min} on the [2, 12] maximum likelihood estimator of the b-value is addressed. New estimators, which are not dependent on the level of completeness, are derived which are simpler to use since they are expressed in terms of the sample central moments namely the mean, standard deviation and skewness. This method works well when the incomplete distribution curves gradually and has one maximum, represented in figure (2). Another benefit of these new estimators is that they provide more accurate hazard and prediction assessments since they take into account the weaker seismic events as well. Aki[2] showed in his paper that the Gutenberg equation (1) can be assumed to be in the form of the probability density function (2) for $m \geq m_{min}$ and $\beta = b\ln(10)$.From [12, 2], the b-value estimator is of the form

$$\hat{\beta} = \frac{1}{\bar{m} - m_{min}}.$$

For this methodology it is assumed that the level of completeness is unknown and so the catalogue has a set of independent random missing seismic events. Let $P_c(m)$ be the probability that a catalogue contains a seismic activity of magnitude m. Then $P_c(m)$ is the probability of completeness of seismic catalogue or commonly known as the probability of detection. So, in general as the magnitudes get higher, the higher the probability of detection will be. Now since the frequency magnitude distribution is affected by $P_c(m)$, we define $f_A(m)$, the apparent frequency magnitude distibution, as the product of $P_c(m)$ and the PDF of the seismic event magnitude (2):

$$f_A(m;\theta) = c.P_c(m,\theta).f(m,\beta). \tag{8}$$

where $c = 1/\int_{m_0}^{m_{max}} P_c(m)f_M(m)dm$ is the normalising coefficient. By definition, the apparent frequency magnitude distribution after normalization is

$$f_A(m,\theta) = \frac{P_c(m,\theta)f_M(m,\beta)}{\int_{m_{min}}^{m_{max}} P_c(m)f)m)dm}$$

A graph of the theoretical Gutenberg-Richter magnitude relation (1) and the apparent frequency magnitude distibution (8) is depicted in figure (2). The figure shows the relationship between the observed(apparent) distribution and the exponential nature of the Gutenberg-Richter relation.

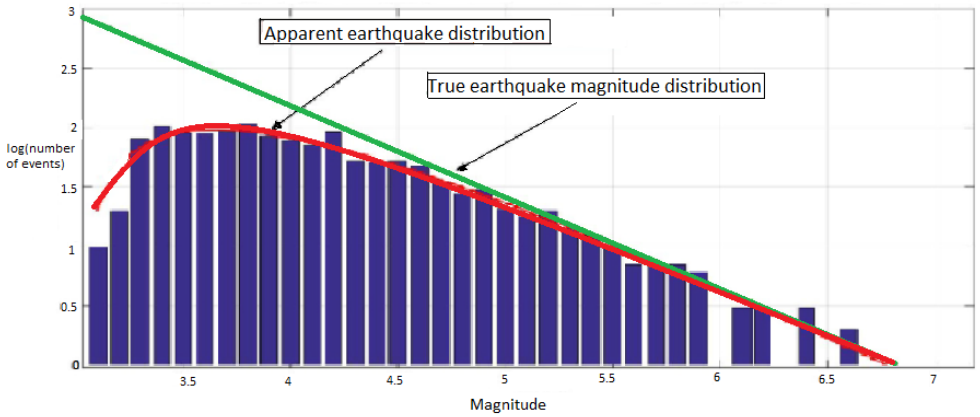


Figure 2: Histogram with an hypothetical seismic catalogue with the apparent frequency distribution and the Gutenberg-Richter frequency-magnitude distribution law.
Figure from [7]

We note that the assessment of the b-value comes from n independent random event magnitudes $M_i (i = 1, .., n)$ where they are all greater than or equal to the detection magnitude m_o. These magnitudes are distributed according to the apparent probability density function (8). In order to effectively work out estimates for the b-value, the functional form of the probability of completeness of the catalog $P_c(m)$ needs to be obtained. Therefore let $P_c(m) \propto (m-\gamma)^{\alpha-1}, (\alpha > 0; m > \gamma)$ where the shape parameter $\alpha > 0$ and the location parameter $\gamma < m$ from [5]. For convenience, all magnitudes M are replaced with $X = M - m_o$ where m_o is the lowest observation in the catalogue. So $f_A(m)$ can now be written as

$$f_A(x) \propto (x-\gamma)^{\alpha-1} \exp[-\beta(x)].$$

So after normalizing,

$$f_A(x) = \frac{(x_i-\gamma)^{\alpha-1} \exp\left[-\beta(x_i-\gamma\right]}{\beta^\alpha \Gamma(\alpha)}, (\alpha > 0, \beta > 0, X > \gamma) \tag{9}$$

where $\Gamma(\alpha)$ is a gamma function. (This equation is a three-parameter gamma distribution or a Type III of the Pearson System of Distributions). The population moments are thus defined as

$$\begin{aligned} \mu_k &= E[(X-\mu_X)^k] \\ &= \int_{-\infty}^{+\infty} (x-\mu_X)^k f_A(x) dx. \end{aligned}$$

The second methodology from [7] is derived by method of moments whereby the moment method estimators (MME) are found by equating the sample moments to the population moments. So the first sample moment is

$$\begin{aligned} \bar{m_1} &= \hat{\mu_1} \\ &= \bar{X} \\ &= \sum_{i=1}^{n} \frac{1}{n} X_i \end{aligned}$$

and the central sample moments are

$$\begin{aligned} \bar{m_k} &= \hat{\mu_k} \\ &= \frac{1}{n} \sum_{i=1}^{n} (X_i - \bar{m})^k. \end{aligned}$$

If equation (9) , the three parameter gamma function is used as a model, then from [3], the system of equations become

$$\begin{aligned} \frac{\alpha}{\beta} + \gamma &= \hat{\mu_1}, \\ \frac{\alpha}{\beta^2} &= \hat{\mu_2}, \\ \frac{2\alpha}{\beta^3} &= \hat{\mu_3}. \end{aligned}$$

Solving for β gives

$$\begin{aligned} \hat{\beta} &= \frac{2\hat{\mu}_2}{\hat{\mu}_3} \\ &= \frac{2\bar{m}_2}{\bar{m}_3} \end{aligned}$$

where $\bar{m}_2 = \frac{\sum_{i=1}^{n}(X_i - \bar{m})^2}{n}$ and $\bar{m}_3 = \frac{\sum_{i=1}^{n}(X_i - \bar{m})^3}{n}$.

4 Application

For the application, the Monte Carlo technique is used for simulations. A hypothetical simulated catalogue was taken from [7] where Equation (2) is used to to simulate the magnitudes with $\beta = 2.303$ or where the b-value is one. The level of completeness is $m_{min} = 4.0$. This is done for 500 simulations with 250 magnitudes each. Then both methodologies are be used to find estimators of the b-value, one for each catalogue. Their results are tested using bias and mean squared error. The data analysis for this essay was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA.

4.1 Weichert(1980)

For Weichert(1980), the data used excludes the curvature part of the distribution since this is one of [14]'s assumption. The m_{min} is taken to be 4 and the m_{max} is 7.5. An iterative process, the Newton Raphson method, is used to compute the b-values. The mean of the b-values is found to be 0.9894861. In Figure3, it is shown how the b-values lie with respect to the theoretical value of 1. In terms of analysis, the bias is worked out to be -0.010514 and the mean squared error is 0.02230.

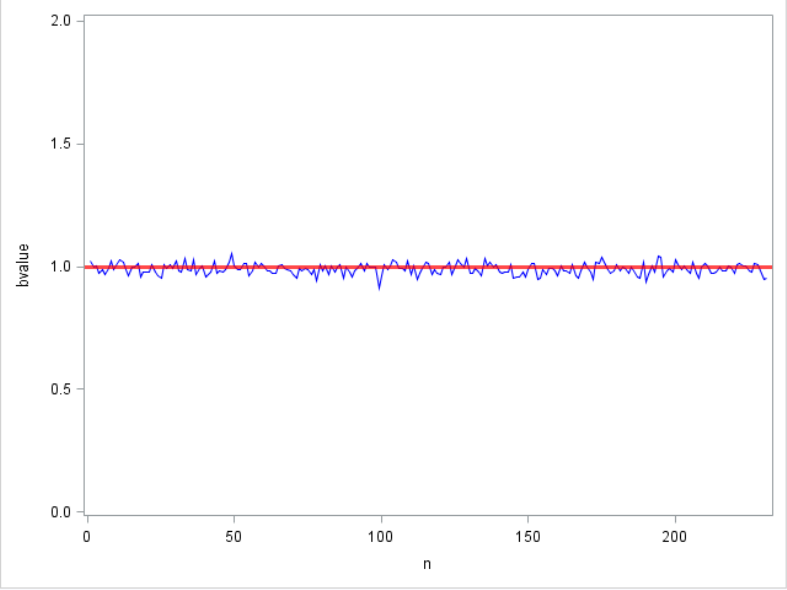


Figure 3: A line plot of the b-values generated from Weichert(1980)[14]

In Figure 4, a histogram of the b-values is shown and the data seems to be very similar to a normal distribution. Most of the values are centered around the value of mean value of 0.989 and the variance is 0.0004974.

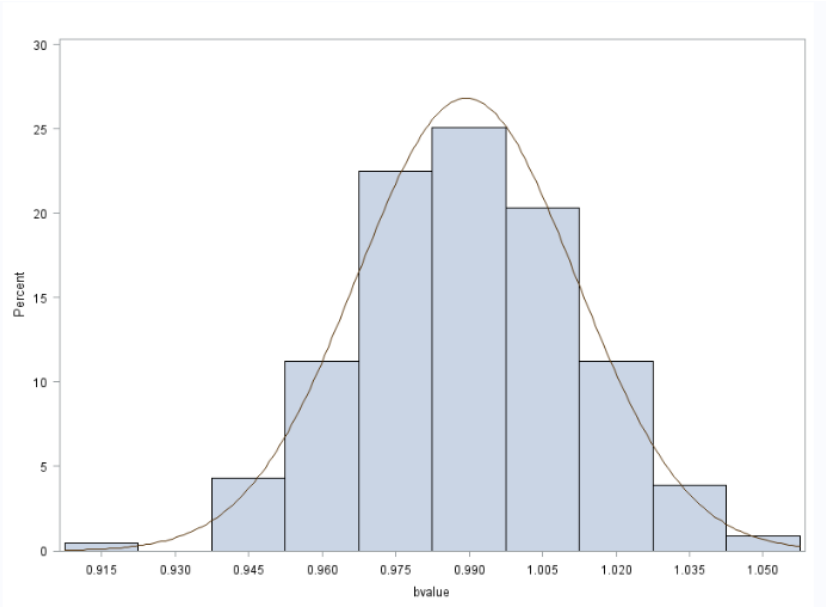


Figure 4: A histogram of the b-values from Weichert(1980)[14]

4.2 Kijko-Smit(2016)

For [7], all 500 catalogues are used. The m_o is assumed to be 3.5, as required by one of the assumptions of this method. The mean for the b-values under this method is 1.0161751. The bias is calculated to be 0.0161751. Figure 5 shows how closer the b-values are to the value of 1. However, the spread tends to be larger for [7].

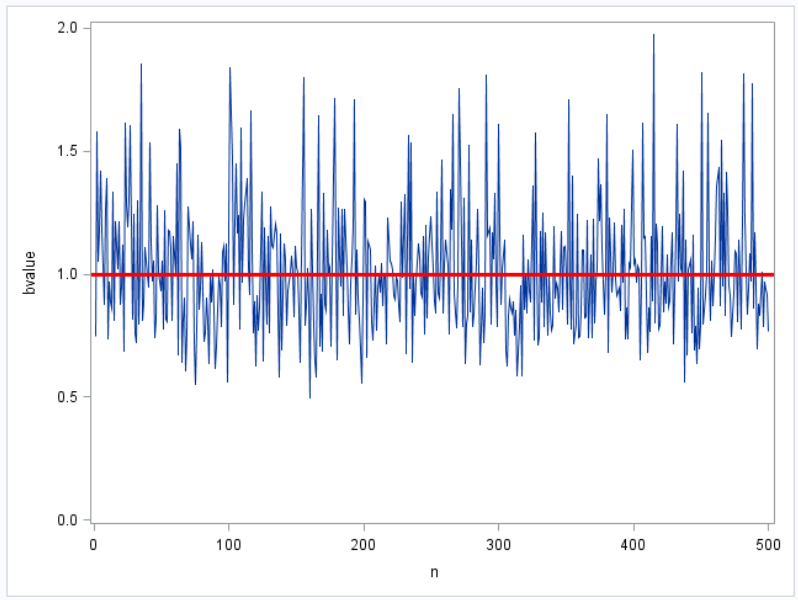


Figure 5: A line plot of the b-values generated from Kijko-Smit(2016)[7]

In Figure 6 most of the values are around the value of 0.9 with a variance of 0.06827. The data is more skewed to the right than [14].

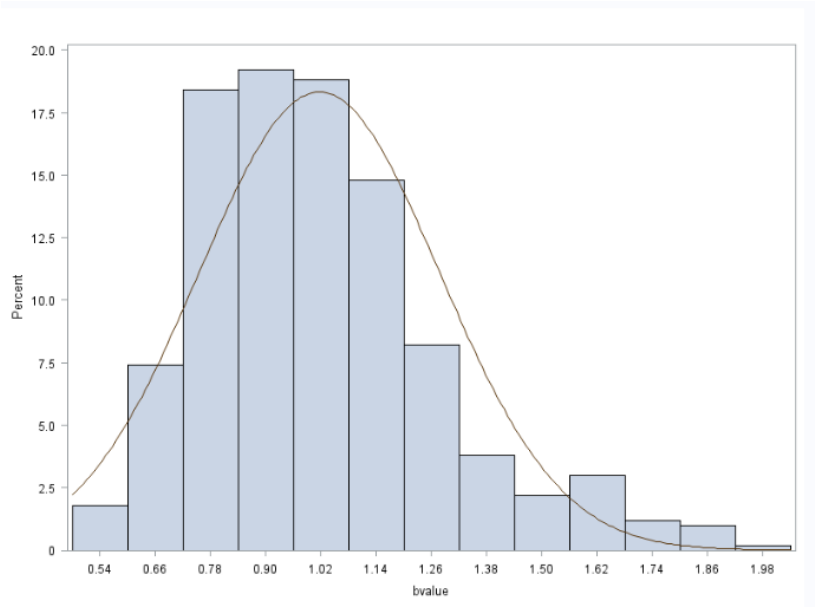


Figure 6: A histogram of the b-values from Kijko-Smit(2016)[7]

5 Conclusion

This paper compares two methods of estimating the b-values from equation (1), namely a Maximum Likelihood Estimation approach and a Method of Moments approach. [14] addresses the issue of unequal observation periods and uses Maximum Likelihood to obtain the estimate. As a result, an iterative process is required when processing. [7] has the advantage of being unaffected by the m_{min}. In addition, the definition of the apparent distribution keeps the simplicity of the Aki-Utsu formula(4) while being able to generate a moment estimate for the b-value.

	Bias	Mean squared error(MSE)	R^2
Weichert(1980)	-0.010514	0.02230	0.9995
Kijko-Smit(2016)	0.0161751	0.26129	0.9381

Table 1: Table comparing results from the two methods.

From table (1), the results of both methods are shown. The bias of [14] is lower than [7] and so is its MSE. The R^2 for [14] is higher than for [7] which suggests that more of the data is explained by [14]. The Method of moments is significantly easier to work out however its bias is higher than The Maximum Likelihood approach. In general, the Method of Moments estimates are less accurate than the Maximum Likehood estimates, but for a bigger sample, both should provide similar and more accurate results. [7] is also specifically effective when the apparent magnitude distribution is gradually curved.(Figure2). Further analysis namely certain non parametric tests, like a *ttest* , can be done in the future to improve results obtained.

References

[1] M Abramowitz and L.A Stegun. *Handbook of Mathematical Functions.* Dover Publishers, New York, 1970.

[2] K. Aki. Maximum Likelihood Estimate of b in the formula $\log n = a - bm$ and its confidence limits. *Bullettin Of The Earthquake Research Institute Tokyo University*, 43:237–239, 1965.

[3] Norman L Johnson, Samuel Kotz, and N Balakrishnan. *Continuous Multivariate Distributions, Volume 1, Models and Applications*, volume 59. New York: John Wiley & Sons, 2002.

[4] A. Kijko. Maximum Likelihood Estimation of Gutenberg-Richter b parameter for uncertain magnitude values. *Pure and Applied Geophysics.*, 127(4), December 1988.

[5] A. Kijko. Maximum Likelihood Estimation of b-value from incomplete catalogs. Part 1. a theoretical background. In *Council for Geosciences*, number 1996-0021, page 14, Pretoria, 1996.

[6] A Kijko and MA. Sellevol. Estimation of earthquake hazard parameters from incomplete data files part 1. utilization of extreme and complete catalogs with different threshold magnitudes. *Bulletin of the Seismological Society of America.*, 1989.

[7] A Kijko and A. Smit. Estimation of the frequency-magnitude Gutenberg-Richter b-value without making assumptions on levels of completeness. Manuscript in preparation.

[8] A. Kijko and A. Smit. Extension of the Aki-Utsu b-value estimator for incomplete catalogs. *Bulletin Of The Seismological Society Of America.*, 102:1283–1287, 2012.

[9] A Kijko, A Smit, and N. Van De Coolwijk. A scenario approach to estimate the maximum foreseeable loss for buildings due to an earthquake in Cape Town. *South African Actuarial Journal*, 2015.

[10] R Page. Aftershocks and microaftershocks of the Great Alaska earthquake of 1964. *Bulletin of the Seismological Society of America*, 1968.

[11] DA Rhoades. Estimation of the Gutenberg-Richter relation allowing for individual earthquake magnitude uncertainties. *Tectonophysics*, 258(1):71–83, 1996.

[12] T. Utsu. A method for determining the value of b in the formula $\log n = a - bm$ showing the magnitude-frequency relation for earthquakes. *Geophysical Journal International*, 1965.

[13] W Wang. Bias-corrected maximum likelihood estimation of the parameters of the Weighted Lindley Distribution. Master's thesis, Michigan Technological University, 2015.

[14] D. Weichert. Estimation of the earthquake recurrence parameters for equal observation periods for different magnitude. *Bulletin Of The Seismological Society Of America.*, 70:1337–1346, 1980.

Appendix A

Weichert(1980)

```
options nodate ps=60 ls=80;
proc import datafile="C:\Users\Deeksha\Documents\University\honours\Research
\my research\Research draft\mycatalog.txt"
dbms=dlm
out=catalog
replace;
delimiter='&';
getnames=yes;
run;
data mycatalog;
set catalog; if _5_25 <= 4 then delete; *cutting off the data; run;
proc iml;
use mycatalog;
read all into xy;
n=231;
start LogLik(b) global(x1);
mean = mean(x1);
max=7.5;
min=4;
f=-1/b+(mean-min) - (((max*exp(-b*(max-min)))/(1-exp(-b*(max-min)))));
betavalue= 1/f;
bvalue=betavalue/log(10);
return bvalue; finish;
do ii = 1 to n; *dividing 250 magnitudes in each catalogue;
n1=250;
        if ii = 1 then do;
        x1=xy[1:n1,1];
        end;
        if (ii>1) then do;
        if(ii<n)then do;
        g=(n1*(ii-1));
        x1=xy[g:g+n1,1];
        end;
        end;
        if(ii=n)then do;
        x1=xy[93250:93500,1];
        end;
num= num //ii;
b = 0.2;
opt={0,2};
call nlpnra(rc, result,"LogLik", b,opt) ; *Newton raphton method;
results = results//result;
end;
bvalues= num||J(n,1,1);
print results;
mean = mean(results);
print mean;
bias = mean - 1;
print bias;
```

```
nm={"n" "bvalue"};
numbeta= num||results;
bbeta= bvalues[,2]||numbeta[,2];
create bvaluematrix from numbeta[colname=nm] ;
append from numbeta ;
create bvaluestheo from bvalues ;
append from bvalues ;
name={'bvalues' 'bvaluehats'};
create together from bbeta[colname=name] ;
append from bbeta ; quit;
ods graphics off;
proc reg data = together;
model  bvaluehats=bvalues / noint; run;
**line plot;
proc template;
define statgraph sgdesign;
dynamic _N _BVALUE; begingraph;
layout lattice / rowdatarange=data columndatarange=data
rowgutter=10 columngutter=10;
layout overlay / yaxisopts=( linearopts=( viewmin=0.0 viewmax=2.0));
seriesplot x=_N y=_BVALUE / name='series' connectorder=xaxis;
referenceline y=1.0 / name='href' yaxis=Y
lineattrs=(color=CXFF0000 thickness=3 );
endlayout;
endlayout;
endgraph; end; run;
proc sgrender data=WORK.BVALUEMATRIX template=sgdesign;
dynamic _N="N" _BVALUE="BVALUE"; run;
**histogram;
proc template;
define statgraph Graph;
dynamic _BVALUE; begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
layout overlay;
histogram_BVALUE/name='histogram'binaxis=falsefillattrs=GraphDataDefault
(color=CX8CA6CE)outlineattrs=(color=CX000000pattern=SOLID thickness=1);
endlayout;
endlayout;
endgraph;
end;
run;
proc sgrender data=WORK.BVALUEMATRIX template=Graph;
dynamic _BVALUE="BVALUE"; run;
proc univariate data = bvaluematrix;
var bvalue;
run;
```

Kijko-Smit(2016)

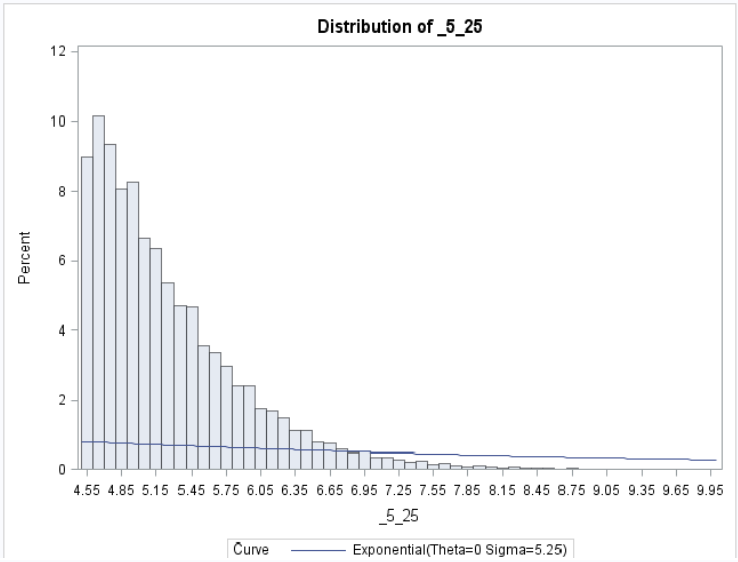
```
options nodate ps=60 ls=80;
proc import datafile="C:\Users\Deeksha\Documents\University\honours\
Research\my research\Research draft\mycatalog.txt"
dbms=dlm
out=catalog
replace;
delimiter='&';
getnames=yes;
run;
proc univariate data=catalog;
var _5_25; histogram / normal;
run;
proc iml;
use catalog; read all into xy;
n=500; mo= 3.5;
do ii = 1 to n;
        n1=250;
        if ii = 1 then do;
        x1=xy[1:n1,1];          end;
        if (ii>1) then do;
        if (ii<500) then do;
        g=(n1*(ii-1));
        x1=xy[g:g+n1,1];
        end;
        end;
        if (ii=500) then do;
        x1=xy[124750:125000,1];
        end;
        xstar= x1-mo;
        m1= mean(xstar);
        sumk=0;
        suml=0;
        do i = 1 to n1;
        x= xstar[i,];
        k=((x-m1)##2);
        l=((x-m1)##3);
        sumk = sumk + k;
        suml = suml + l;
        end;
        m2 = (sumk)/n1;
        m3 = (suml)/n1;
        beta= (2*m2)/m3;
        bvl= beta/log(10);
        num= num //ii;
        betam = betam//bvl;
        end;
bvalues= num||J(n,1,1);
print betam ;
mean = mean(betam);
print mean;
bias = mean - 1;
```

```
print bias;
nm={"n" "bvalue"};
numbeta= num||betam; bbeta= bvalues[,2]||numbeta[,2];
create betamatrix from numbeta[colname=nm] ; append from numbeta ;
create betavalues from bvalues ; append from bvalues ;
name={'bvalues' 'betahats'};
create bmatrix from bbeta[colname=name] ; append from bbeta ;
quit;
proc reg data = bmatrix;
model  betahats=bvalues / noint; run;
***line plot***;
proc template; define statgraph sgdesign;
dynamic _N _BVALUE; begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;        layout overlay / yaxisopts=( linearopts=
( viewmin=0.0 viewmax=2.0));
seriesplot x=_N y=_BVALUE / name='series' connectorder=xaxis;
referenceline y=1.0 / name='href' yaxis=Y lineattrs=(color=CXFF0000
thickness=3 );
endlayout;     endlayout; endgraph; end; run;
proc sgrender data=WORK.BETAMATRIX template=sgdesign;
dynamic _N="N" _BVALUE="BVALUE"; run;
***histogram***;
proc template; define statgraph Graph; dynamic _BVALUE;
begingraph;
layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
layout overlay;
histogram _BVALUE / name='histogram' binaxis=false;
endlayout;     endlayout; endgraph; end; run;
proc sgrender data=WORK.BETAMATRIX template=Graph;
dynamic _BVALUE="BVALUE"; run;
proc univariate data=betamatrix;
var bvalue; run;
```

Appendix B

Weichert(1980)

The UNIVARIATE Procedure



The UNIVARIATE Procedure
Fitted Exponential Distribution for _5_25

Parameters for Exponential Distribution		
Parameter	Symbol	Estimate
Threshold	Theta	0
Scale	Sigma	5.254407
Mean		5.254407
Std Dev		5.254407

Goodness-of-Fit Tests for Exponential Distribution				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.5761	Pr > D	<0.001
Cramer-von Mises	W-Sq	4554.0470	Pr > W-Sq	<0.001
Anderson-Darling	A-Sq	21015.4670	Pr > A-Sq	<0.001

Quantiles for Exponential Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	4.51000	0.05281
5.0	4.55000	0.26952
10.0	4.60000	0.55361
25.0	4.75000	1.51160
50.0	5.07000	3.64208
75.0	5.56000	7.28415
90.0	6.16000	12.09872
95.0	6.57000	15.74080
99.0	7.51000	24.19744

Newton-Raphson Optimization with Line Search

Without Parameter Scaling

Gradient Computed by Finite Differences

CRP Jacobian Computed by Finite Differences

Parameter Estimates	1

Optimization Start			
Active Constraints	0	Objective Function	-0.039270503
Max Abs Gradient Element	0.2714247212		

Iteration		Restarts	Function Calls	Active Constraints		Objective Function	Objective Function Change	Max Abs Gradient Element	Step Size	Slope of Search Direction
1	*	0	48	0		-12948076	12948076	9.69E14	0.0747	-15905
1		1	105	0		-12948076	0	9.69E14	0	-5039E9

Optimization Results			
Iterations	1	Function Calls	106
Hessian Calls	4	Active Constraints	0
Objective Function	-12948075.71	Max Abs Gradient Element	9.6895031E14
Slope of Search Direction	-5.038808E12	Ridge	0

WARNING: Optimization routine cannot improve the function value.

NEWRAP Optimization cannot be completed.

The SAS System

Optimization Results			
Parameter Estimates			
N	Parameter	Estimate	Gradient Objective Function
1	X1	0.946855	9.6895031E14

Value of Objective Function = -12948075.71

results
1.0242449
0.9970216
1.0046273
0.9747688
0.9854406
0.9700481
0.9922298
1.0218265
0.9869173
1.0118381
1.0282666
1.0170132
0.9883374
0.9632011
0.9974292
0.999647

.
.
.

0.9964004
0.9722206
1.0099441
1.0141241
1.0007285
1.0040803
0.9832886
0.9769505
1.0147316
1.009833
0.9822792
0.9468552
0.9468552

mean
0.9894605

bias
-0.01054

The SAS System

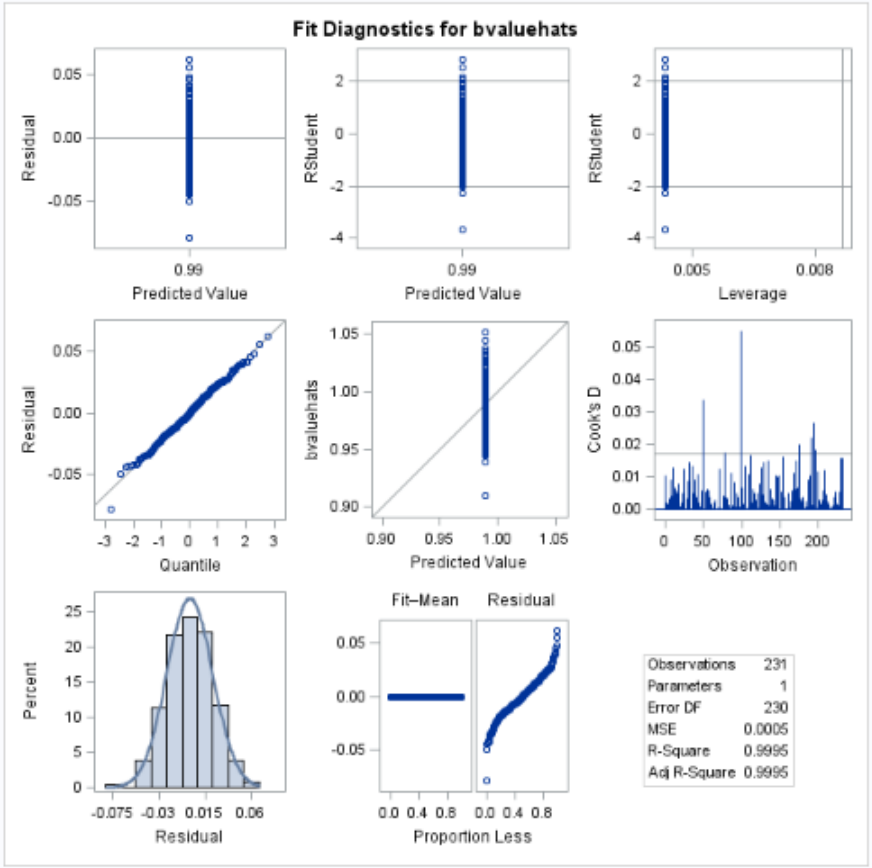
The REG Procedure
Model: MODEL1
Dependent Variable: bvaluehats

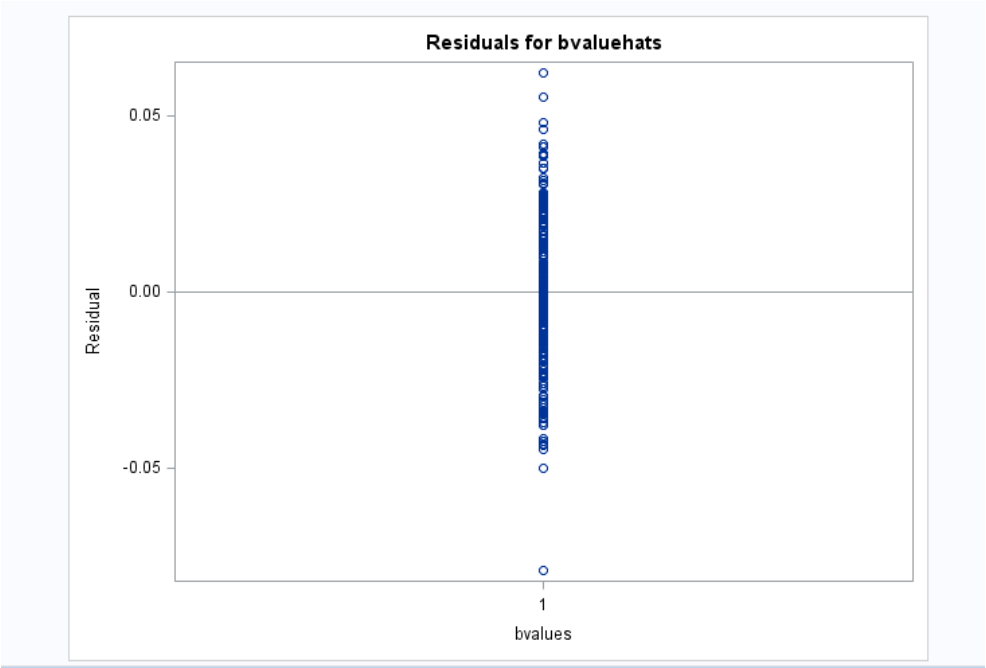
Number of Observations Read	231
Number of Observations Used	231

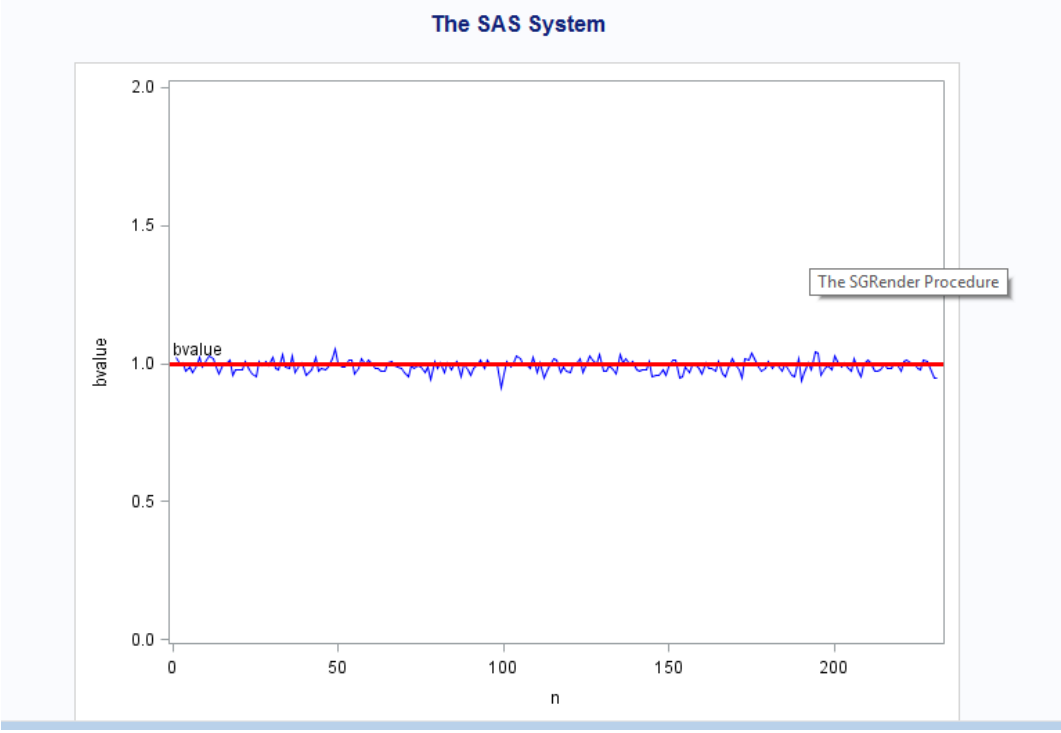
Note: No intercept in model. R-Square is redefined.

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	226.15641	226.15641	454643	<.0001
Error	230	0.11441	0.00049744		
Uncorrected Total	231	226.27082			

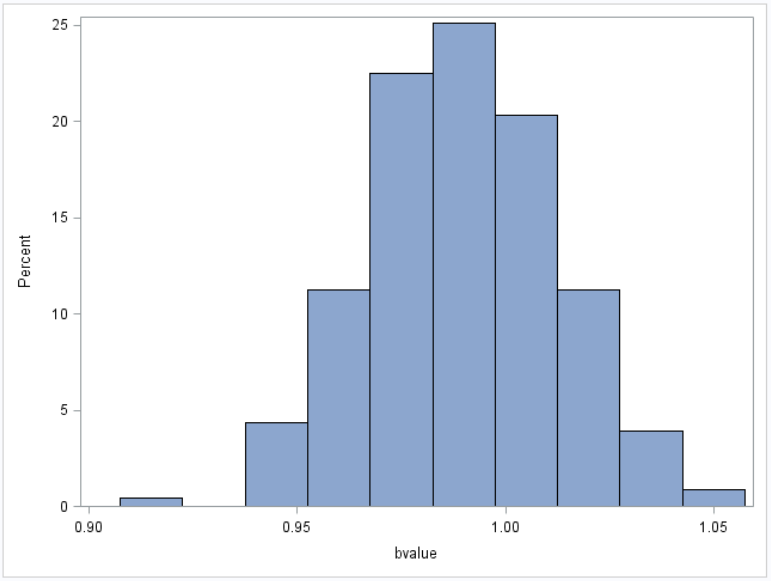
Root MSE	0.02230	R-Square	0.9995
Dependent Mean	0.98946	Adj R-Sq	0.9995
Coeff Var	2.25409		




Residuals for bvaluehats
0.05
0.00
-0.05
Residual
1
bvalues


The SAS System
2.0
1.5
1.0
0.5
0.0
bvalue
bvalue
The SGRender Procedure
0
50
100
150
200
n

The SAS System



The SAS System

The UNIVARIATE Procedure
Fitted Normal Distribution for bvalue

Parameters for Normal Distribution		
Parameter	Symbol	Estimate
Mean	Mu	0.98946
Std Dev	Sigma	0.022303

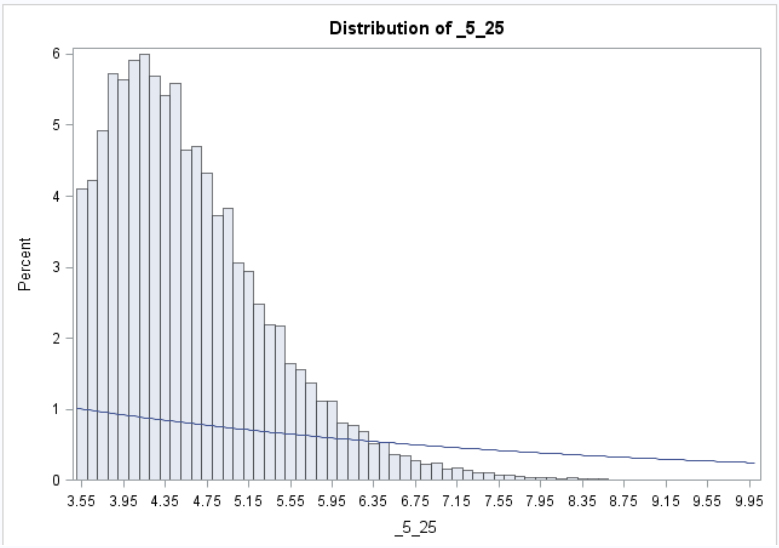
Goodness-of-Fit Tests for Normal Distribution				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.03300048	Pr > D	>0.150
Cramer-von Mises	W-Sq	0.03695989	Pr > W-Sq	>0.250
Anderson-Darling	A-Sq	0.24332967	Pr > A-Sq	>0.250

Quantiles for Normal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	0.94461	0.93758
5.0	0.95323	0.95277
10.0	0.95979	0.96088
25.0	0.97473	0.97442
50.0	0.98897	0.98946
75.0	1.00533	1.00450
90.0	1.01624	1.01804
95.0	1.02610	1.02615
99.0	1.03733	1.04135

Kijko-Smit(2016)

The SAS System

The UNIVARIATE Procedure



The UNIVARIATE Procedure
Variable: _5_25

Moments			
N	125000	Sum Weights	125000
Mean	4.59414408	Sum Observations	574268.01
Std Deviation	0.78897603	Variance	0.62248317
Skewness	1.18238042	Kurtosis	1.9127696
Uncorrected SS	2716079.75	Corrected SS	77809.7736
Coeff Variation	17.1735151	Std Error Mean	0.00223156

Basic Statistical Measures			
Location		Variability	
Mean	4.594144	Std Deviation	0.78898
Median	4.440000	Variance	0.62248
Mode	4.180000	Range	6.47000
		Interquartile Range	1.01000

Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	2058.713	Pr > |t|	<.0001
Sign	M	62500	Pr >= |M|	<.0001
Signed Rank	S	3.9063E9	Pr >= |S|	<.0001

Quantiles (Definition 5)	
Level	Quantile
100% Max	9.97
99%	7.09
95%	6.11
90%	5.66
75% Q3	5.01
50% Median	4.44
25% Q1	4.00
10%	3.73
5%	3.62
1%	3.52
0% Min	3.50

betam
0.7463958
1.5802571
1.0536411
1.178575
1.4186391
1.1811123
0.8783889
1.2613812
1.3864442
0.7379353
0.9669934
0.889488
0.8601197
1.3326045
0.812052

.
.
.

0.9728909
1.7763085
0.8646608
1.1659507
0.9522082
0.6987193
0.8798233
0.8304028
1.0091616
0.7874138
0.9680801
0.9387252
0.9179585
0.7678773

mean
1.0161751

bias
0.0161751

The REG Procedure
Model: MODEL1
Dependent Variable: betahats

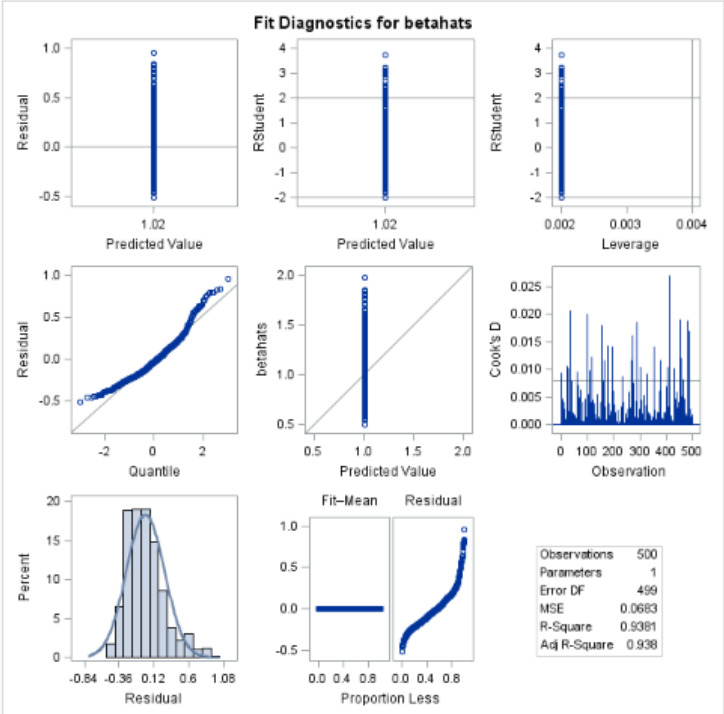
Number of Observations Read	500
Number of Observations Used	500

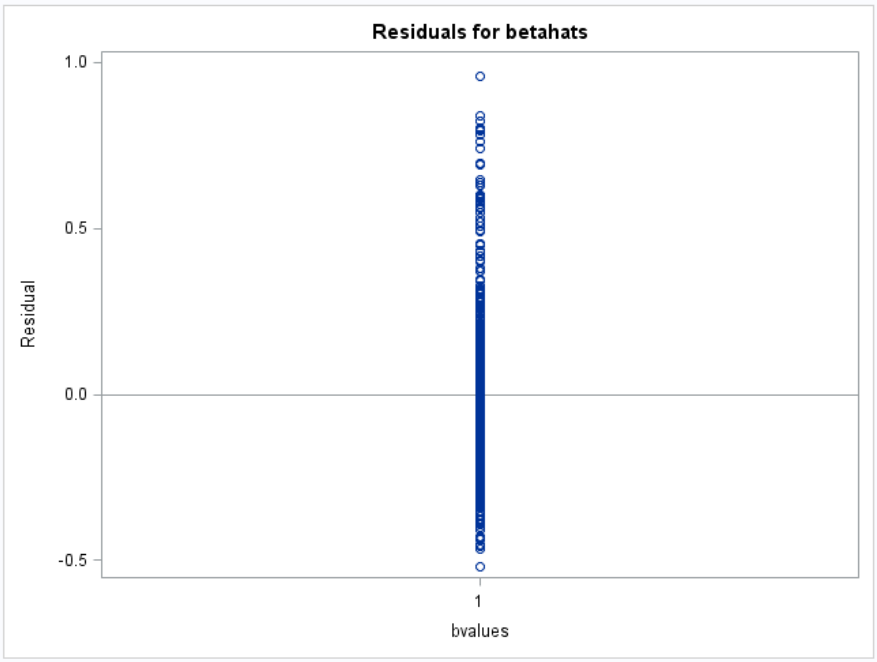
Note: No intercept in model. R-Square is redefined.

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	516.30594	516.30594	7562.32	<.0001
Error	499	34.06848	0.06827		
Uncorrected Total	500	550.37441			

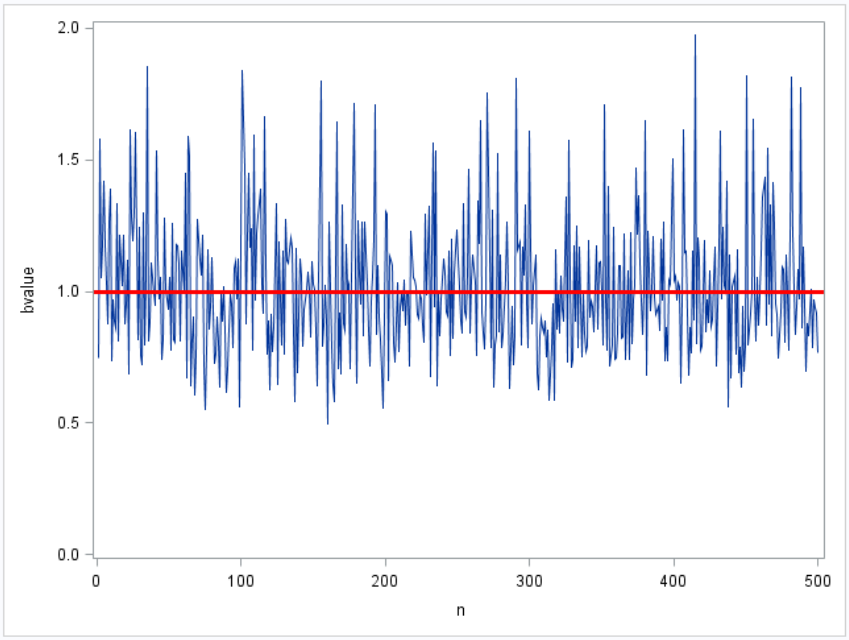
Root MSE	0.26129	R-Square	0.9381
Dependent Mean	1.01618	Adj R-Sq	0.9380
Coeff Var	25.71328		

The REG Procedure
Model: MODEL1
Dependent Variable: betahats

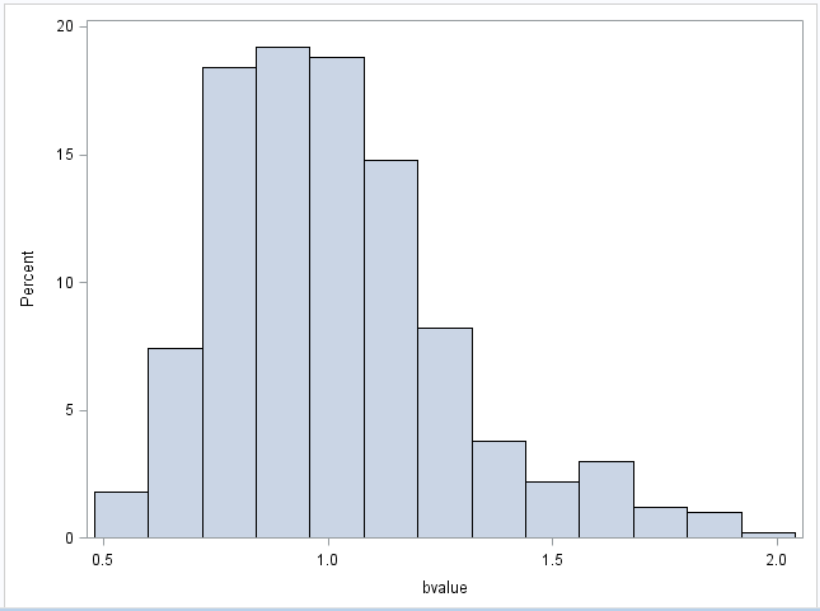




The SAS System



The SAS System



The SAS System

The UNIVARIATE Procedure
Variable: bvalue

Moments			
N	500	Sum Weights	500
Mean	1.01617512	Sum Observations	508.08756
Std Deviation	0.26129199	Variance	0.0682735
Skewness	0.91982161	Kurtosis	0.9016024
Uncorrected SS	550.374415	Corrected SS	34.0684778
Coeff Variation	25.7132833	Std Error Mean	0.01168533

Basic Statistical Measures			
Location		Variability	
Mean	1.016175	Std Deviation	0.26129
Median	0.973857	Variance	0.06827
Mode	.	Range	1.47704
		Interquartile Range	0.33075

Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	86.96159	Pr > |t|	<.0001
Sign	M	250	Pr >= |M|	<.0001
Signed Rank	S	62625	Pr >= |S|	<.0001

Quantiles (Definition 5)	
Level	Quantile
100% Max	1.973931
99%	1.812181
95%	1.578058
90%	1.339301
75% Q3	1.158331
50% Median	0.973857
25% Q1	0.827586
10%	0.732670
5%	0.667380
1%	0.571272
0% Min	0.496893

The SAS System

The UNIVARIATE Procedure
Fitted Normal Distribution for bvalue

Parameters for Normal Distribution		
Parameter	Symbol	Estimate
Mean	Mu	1.016175
Std Dev	Sigma	0.261292

Goodness-of-Fit Tests for Normal Distribution				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.06935659	Pr > D	<0.010
Cramer-von Mises	W-Sq	0.89441318	Pr > W-Sq	<0.005
Anderson-Darling	A-Sq	6.22017158	Pr > A-Sq	<0.005

Quantiles for Normal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	0.57127	0.40832
5.0	0.66738	0.58639
10.0	0.73267	0.68132
25.0	0.82759	0.83994
50.0	0.97386	1.01618
75.0	1.15833	1.19241
90.0	1.33930	1.35103
95.0	1.57806	1.44596
99.0	1.81218	1.62403

Statistical thinking models

Megan Sajiwan 11065177

STK795 Research Report

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

Supervisor: Dr. L. Fletcher

Department of Statistics, University of Pretoria

2 November 2016

Abstract

This research project covers an exploration of statistical thinking models with a focus on the paradigm shifts and perspectives surrounding the instruction and learning of statistics. Statistical thinking is not an innate ability yet it is an indispensable tool that is not confined to pedagogy but has applications in several other fields as well as everyday life. As such, it has become increasingly important to nurture and develop the way in which statistics is taught and learnt in introductory statistics courses.

In this essay, the literature covering the development and application of statistical thinking models will be reviewed in conjunction with developments in education research relating to the teaching and learning of statistics. The statistical thinking models central to the topic will be explored in some detail and then applied to an introductory statistics course. Specifically, the perceptions of students enrolled in an introductory statistics course at the University of Pretoria will be analysed using quantitative and qualitative approaches in order to ascertain if the students are able to think statistically upon completion of the course and if the course is adequate in fostering statistical thinking in the students.

Declaration

I, *Megan Sajiwan,* 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.

_____
Megan Sajiwan

_____
Dr. Lizelle Fletcher

_____
Date

Acknowledgements

The author would like to thank the Centre for Artificial Intelligence Research (CAIR) for financial support in the form of a postgraduate bursary.

Contents

1 Introduction **6**

2 Background theory **9**
2.1 Statistical thinking models 9
2.2 The teaching and learning of statistics 11
2.3 Applying statistical thinking models to teaching and learning 12

3 Application **14**
3.1 Description 14
3.2 Methodology 17
3.3 Results 17
3.4 Qualitative study 21

4 Conclusion and recommendations **23**

Appendix **26**

List of Figures

1 The 4-dimensional framework developed by Pfannkuch and Wild [11] 11
2 Regression procedure output for the analysis of variance 19
3 Regression procedure output for parameter estimates 20

List of Tables

1 Mean response to each statement 18

1 Introduction

In the world we live in, we are inundated with statistics, figures and charts reported in the media or presented by private or governmental organisations; the ubiquity and availability of data have increased substantially over the past years bringing to light the need for statistical literacy. The ability to make decisions given the data presented in everyday life has become an important skill to add to one's set. Should one find oneself in a career centered on statistics or a field that relies on statistical analysis, this ability needs to be sharpened. The way in which one may become statistically literate and furthermore, highly skilled, is through grasping statistical concepts and gathering a firm understanding thereof. Understanding statistics is based on being able to think statistically and being an effective thinker. Statistical thinking models are concerned with mapping the process of thinking that is used when solving statistical problems. Statistical thinking is something that must be taught in order for one to be able to think and reason in a way that will enable one to tackle problems. In introductory statistics courses, it is, therefore, paramount to instill in students a high level of statistical thinking and part of the agenda of statistical pedagogy to not only convey statistical concepts but convey meaning along with these concepts.

In this research project, statistical thinking is defined and the models that capture the process are examined; literature on statistical thinking models, statistical teaching and learning and the practice of statistics will henceforth be examined extensively. The application of these concepts and models in terms of teaching and learning statistics, and statistical thinking will be explored and then applied to an introductory statistics module to ascertain whether statistical thinking is being effectively encouraged by lecturers and instilled in the students. This will be achieved via an analysis incorporating both quantitative and qualitative methods. For the purpose of this report, an introductory statistics module offered at the University of Pretoria, formed the focus of the analysis. A survey of students enrolled in this course was taken with the aim of gauging the students' perceptions. The data from the survey was analysed with the aim of determining whether the students were able to engage in statistical thinking after completion of the module. In addition to this, interviews were conducted with a few willing students to further explore the perceptions the students have of the module. Critique of the teaching approach will be given along with recommendations to improve the module in terms of fostering the development of statistical thinking.

Introductory statistics courses form the foundation of statistical learning. Concepts learned in this environment need to be cemented in the minds of the student and, more importantly, the concepts must be understood and the student should be able to use these concepts to solve problems. The reason for this analysis is to discover how students learn statistics so as to be able to find a way to create understanding. Examining the way in which students think about statistics and learn these concepts is the first building block in improving the process to ensure that the student has learnt effectively.

Statistical literacy, reasoning, and thinking are all fundamental to a thorough understanding of statistics. The ability to think statistically must be taught and a deep impression of the thinking process should be left on all students of statistics regardless of the level. In [2], Ben-Zvi and Garfield defined statistical literacy, reasoning, and thinking as follows:

Definition 1. Statistical literacy: The fundamental skills required to understand and analyse results and information. These skills include being able to organise and represent the data in a sensible way and being able to work with this data. To be statistically literate is to understand the terminology, notation, and concepts.

Definition 2. Statistical reasoning: Statistical reasoning involves being able to understand, interpret, explain and statistically summarise the data. It also refers to the ability to link several concepts.

Definition 3. Statistical thinking: To think statistically means to be able to understand the reasons for statistical investigation and identify the ubiquity of the main concepts of statistics. Statistical thinking encompasses the process of deciding on an appropriate technique and, furthermore, when to apply the statistical concepts. The ability to synthesise statistical and contextual knowledge to find a solution to

a problem is imperative in statistical thinking and finally, to achieve a high level of statistical thinking, the ability to critically evaluate results is paramount.

The aforementioned definitions are not the result of the first attempt to give a concise structure to the thinking process. In [5], Mallows defined what he calls the zeroth problem that should precede Fisher's first problem (the problem of specification that involves choosing the mathematical form of the population). The zeroth problem is given as follows:

> Problem (0): "Considering the relevance of the observed data, and other data that might be observed, to the substantive problem."

Mallows claimed that statistical thinking is at the core of solving the zeroth problem and with that provided a definition for statistical thinking:

> "Statistical thinking concerns the relation of quantitative data to a real world problem, often in the presence of variability and uncertainty. It attempts to make precise and explicit what the data has to say about the problem of interest."

Having defined statistical thinking, the models that map the statistical thinking process must be examined. Pfannkuch and Wild [11] developed the 4-dimensional model of statistical thinking during data-based enquiry. The authors introduced the following dimensions in the framework which will be explicated in the following section:

- **Dimension 1**: The investigative cycle
- **Dimension 2**: Types of thinking: The five types of thinking are modelled and elaborated on by Pfannkuch and Wild in [8].
    1. Recognition of the need for data;
    2. Transnumeration;
    3. Consideration of variation;
    4. Reasoning with statistical models;
    5. Integrating the statistical and contextual.
- **Dimension 3**: Interrogative cycle
- **Dimension 4**: Dispositions

Given that the five types of thinking and the 4-dimensional framework have been modelled, it begs the question of how to approach statistical thinking from a pedagogical perspective.

Pfannkuch and Wild [8] traced the origins and progress of statistical thinking to gain insight into the stages of thinking that a student is required to undergo in order to develop statistical thinking completely as described by the five thinking types. The authors uncovered main factors on which statistical thinking is based: analysing data to gain knowledge; acknowledging that statistics can be used to map social behaviour; statistical models can be applied to various fields; new tools must be developed for analysis.

Statistics is not an isolated field and has applications across several disciplines; as such, statistical thinking and the models that describe it are used in these various fields. Pfannkuch and Wild [8] then considered the contributions to statistical thinking from fields as diverse as epidemiology, psychology, quality management, and contributions from statisticians themselves. Each of these fields uses statistical concepts as part of the process involved in analysing the real life situations on which judgements need to be made. In most fields, the five types of thinking are apparent in the approach to solving the problem. It has been seen that the perceptions of most people are built on experiences that can be limiting. Thus, in order to encourage sound judgement of a situation, statistical thinking must be taught. Contributions from statistics education

researchers revealed information about the way in which students learn and the challenges facing both students and instructors. The authors concluded their analysis of teaching and learning statistical thinking by identifying four major challenges in statistical thinking: raising awareness of statistical thinking; recognising statistical thinking in various situations; developing teaching strategies; and teaching and assessing students in a way that will encourage statistical thinking.

Other studies relating to the challenges faced by instructors of introductory statistics courses have also been carried out by Ben-Zvi and Garfield in [2] who suggested that instructors of introductory statistics courses face many challenges due to students viewing these courses and the coursework as difficult to engage with.

Much work has gone into finding solutions to the aforementioned challenges faced by both students and instructors. The way in which statistics is taught and learnt has constantly evolved and there is now a need for a focus on the concepts and thinking processes involved in finding solutions to real-world problems. There is much to be gained from analysing the methods of practising statisticians who work with real-world problems in a variety of contexts as finding a solution requires a high level of statistical thinking. Mimicking the approach in a classroom environment reveals possible approaches to enhancing statistical learning in students.

Chance [3] examined the promotion of statistical thinking in introductory statistics courses. The main aim of teaching, in this instance, is to make the student "an informed consumer of statistical information". Exposure to the types of thinking used by practising statisticians can enhance the learning of students. Constant repetition of statistical thinking habits will ensure that the student retains these habits and is able to apply it in other courses and fields. A similar approach to finding a solution was taken by Pfannkuch and Wild [7] who explored the way in which applied statisticians working in different backgrounds approach problems. Their aim was to investigate thinking processes and form an idea of how to enrich teaching with the information acquired.

Finding a solution to the challenges faced using a purely pedagogical approach is of utmost importance when faced with the task of instilling statistical thinking in students enrolled in introductory statistics courses. Rumsey [9] defined the ideal outcome for introductory statistics courses as preparing students for dealing with data that they would encounter on a daily basis and in their careers. To achieve this, Rumsey suggested that the student should become a good "statistical citizen" meaning one who can critically analyse data encountered and is essentially statistically literate. As a second goal, students should be imbued with research skills incorporating the scientific method. There is a need for statistical literacy, at any level of statistical knowledge and reason, in order to understand concepts and language used in everyday statistical reporting and in the media. The author explored misconceptions about understanding statistical ideas and gave suggestions to engender understanding.

On the issue of data and the zero$^{\text{th}}$ problem posed by Mallows [5], Rumsey suggested that allowing students to produce their own data and yield results based on this data will motivate the student. Students should also be able to communicate the results and findings. Here, tasks aimed at interpreting and effectively communicating results is a useful tool to create understanding. Ben-Zvi and Garfield [2] recommended incorporating more data and concepts instead of focusing on theory; using real data and not simply realistic data to develop the students' statistical literacy, reasoning, and thinking. However, when given existing data, as is often the case, Cobb and Moore [4] suggested that exploratory data analysis should be used as a starting point as basic methods and concepts can be used on existing data without the need to produce or collect the data. Students should be asked to engage in interpretation of results from the outset to build a foundation for interpreting problems stemming from concepts that are of an increased difficulty level.

Technology has resulted in statistics becoming operational and it has created an opportunity for students to use various resources to aid their studies. Moore [6] explored what makes a student an effective learner and how to enable effective learning. He made a case for the integration of technology into the curriculum in a capacity where the technology enhances teaching but does not replace it. Traditional teaching methods

should be synthesised with technology-based teaching methods such as the use of multimedia as a teaching aid and the use of computational technology that will allow for exposure to practical statistics. Statistical thinking encourages a focus on strategy in statistics. Snee [10] stated that the advancements in technology require the statistician to assume the role of a strategist in addition to the role of a problem solver using statistical tools.

Statistical thinking is a complex issue that can be approached from many angles to achieve the common goal of improving the understanding people have of statistics whether it be in an everyday situation or applied in one's career. As discussed above, there are numerous challenges to achieving this goal but they are not without solutions.

2 Background theory

From the literature, it appears that statistical thinking models and the goals of statistical teaching and learning run parallel. It is the goal of this essay to apply statistical thinking models to statistical pedagogy and to assess if the outcomes of teaching actually encourage statistical thinking. In this section, statistical thinking models will be discussed in depth along with ideal teaching and learning outcomes for introductory statistics modules.

2.1 Statistical thinking models

Pfannkuch and Wild have been the main contributors to the development of statistical thinking models as can be seen in the previous section. These models will be used in the analysis in the next section.

Pfannkuch and Wild [11] conducted interviews with students and professional statisticians in order to gain information and build a deeper understanding of the inner workings of statistical thinking. These interviews led to the construction of the 4-dimensional model which attempts to explain the process of statistical thinking during data-based enquiry. The thinker operates in all dimensions simultaneously albeit it with varying degrees of attention. Pfannkuch and Wild expand on each of the dimensions in the framework:

1. **The investigative cycle:** This was adapted from the model originally developed by McKay and Oldford in Pfannkuch and Wild [7]. This investigation cycle is designed to encourage other investigation cycles with the aim of solving a statistical problem that is part of a real-world problem.

2. **Types of thinking:** The 5 types of thinking - recognition of the need for data; transnumeration; consideration of variation; reasoning with statistical models, and integrating the statistical and contextual - the 5 types of thinking will be expanded on in the discussion below.

3. **The interrogative cycle:** This is the general thinking process followed during problem-solving and includes the following steps:

   (a) Generate: The generation of ideas via brainstorming in groups or individually. This can be applied to any problem.

   (b) Seek: Recalling existing knowledge or acquiring new knowledge relating to the problem.

   (c) Interpret: This phase involves making connections between new ideas and the existing information from the model and then incorporating these ideas into the model.

   (d) Criticise: To criticise here is to check the new information against internal reference points and validate the new information based on these points.

   (e) Judge: The judging phase is essentially the decision-making phase. The thinker decides what information is important and, conversely, what information is redundant and must hence be trimmed.

4. **Dispositions:** Personal characteristics of the thinker will have an effect on the thinking process. General characteristics include:

(a) Curiosity and awareness: Asking the questions "Why?" and "How?" is the basis of idea generation and discovery.

(b) Engagement: A deep-set interest in the problem will result in higher levels of observation and provides some motivation.

(c) Imagination: The ability to analyse the model or problem from different points of view requires an imaginative approach from the thinker.

(d) Scepticism: This characteristic allows one to view the problem critically and with the aim to evaluate the credibility of a solution.

(e) Being logical: The ability to think and argue logically is important in any problem and is the surest way to reach a reasonable conclusion.

(f) A propensity to seek deeper meaning: One should not merely accept the solutions presented. Instead, the desire to understand on a deeper level should be the dominant plan of approach.

From the 4-dimension framework, the second dimension, types of thinking, is the most useful at providing direction with regards to the paradigms one should become familiar with to be able to think statistically at a high level. Pfannkuch and Wild [8] built on the 5 types of thinking that are hierarchical in nature, with one thinking type preceding another that eventually leads to the individual having a holistic understanding of the statistical and contextual aspects of a problem. The 5 thinking types are stated and defined as follows:

1. **Recognition of the need for data**: In order to analyse real-world problems and make judgements that are sounds, identifying the need for data and the proper collection thereof is the foremost step that must be adhered to.

2. **Transnumeration**: The term "transnumeration", coined by Pfannkuch and Wild [11], refers to the stimulation of understanding through changing the representation of the data. Transnumeration occurs when graphically representing and summarising the data in order to gain knowledge from the data and to allow for communication of the situation and its analysis.

3. **Consideration of variation:** Understanding the theory behind variation and its presence and role in real life situations is critical in developing a thorough grasp of statistics and its applications. Being able to mitigate the sources of variability is also especially important. Snee [10] has stressed the importance of variation as a key concept in statistics and in other fields such as process control. Snee states that variation is an important concept in statistical thinking as it needs to be recognised that variation is present in processes whether or not data has been collected or analysed. The idea that variation is present can still be used in problem-solving if data cannot be collected as even though data is important to encourage statistical thinking; the thinking can be done without it if necessary.

4. **Reasoning with statistical models**: Models allow one to engage with the data on an aggregate basis and detect patterns in the data. The concept of variation comes through when analysing a model as variation can be detected more efficiently from a model. These models can then be used to reason with the data more effectively.

5. **Integrating the statistical and contextual:** The last type of thinking relates to the synthesis of the contextual knowledge, gained by applying the prior types of thinking, and statistical knowledge, and is a fundamental element of statistical thinking.

The 4-dimensional framework, illustrated in Figure 1, further explains each of the dimensions and their respective flows or components.

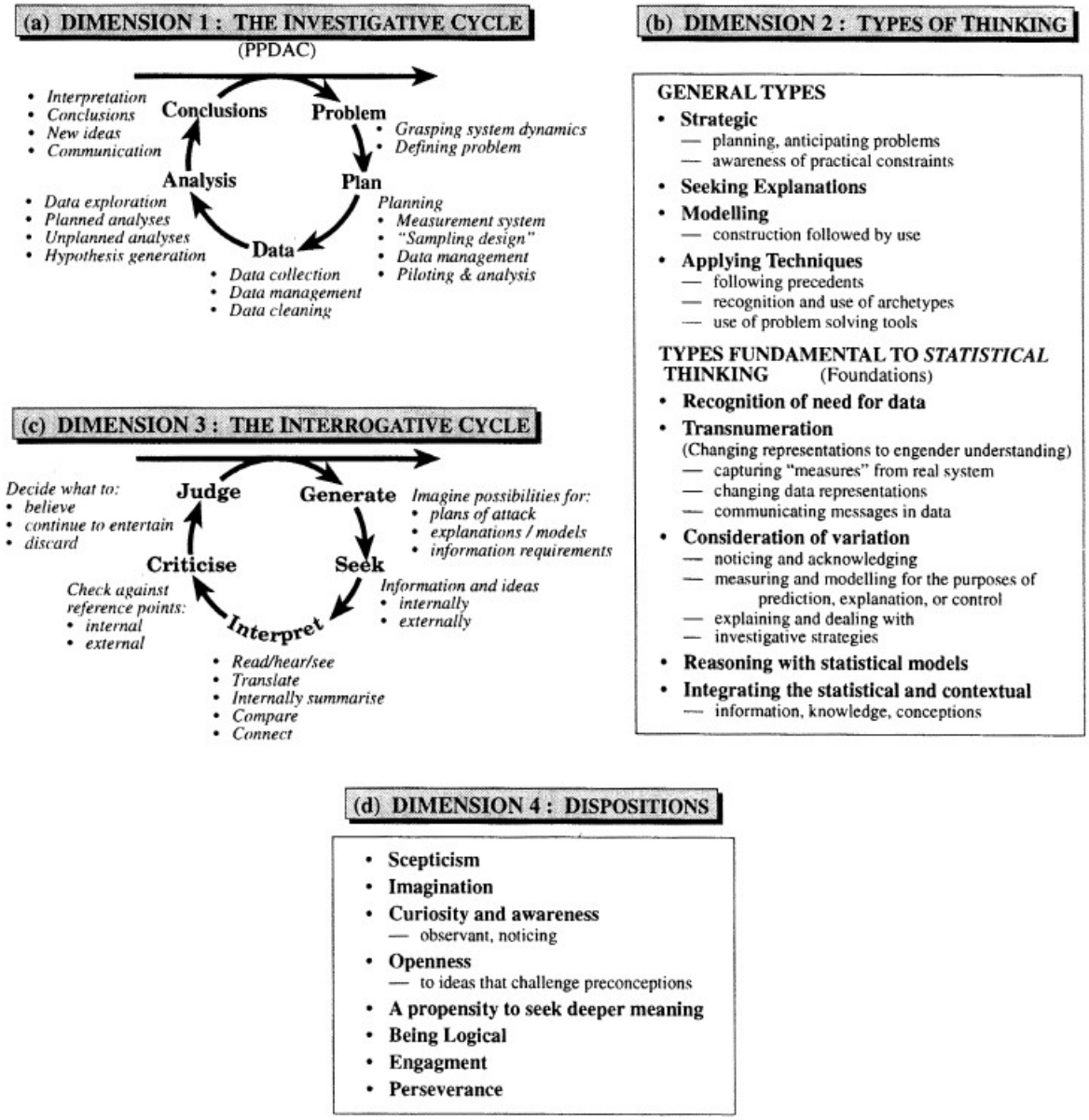


Figure 1: The 4-dimensional framework developed by Pfannkuch and Wild [11]

2.2 The teaching and learning of statistics

The way in which introductory statistics courses are taught is crucial but it is not simple and this fact is noted in [2] where Ben-Zvi and Garfield explored the many challenges faced by instructors, listed as follows:

1. The complexity of statistical concepts and the counter-intuitive nature thereof. Students are not easily motivated to learn the concepts.

2. Students experience difficulty with the mathematics which hampers their learning of the statistical concepts.

3. Students tend to rely on their personal experiences when analysing the context rather than following the statistical procedure to come to conclusions.

4. Students do not recognise the differences between mathematics and statistics and they are unfamiliar and uncomfortable with the spectrum of interpretations that is common in statistics but uncommon in mathematics.

From the previous section, Rumsey introduced the idea of a student being a good "statistical citizen" in [9]. The crux of the matter discussed by the author is that students should be critical of results and statistically literate. This literacy is impeded when emphasis is placed on areas that do not lead a student to understand the coursework. The author explored the misconceptions about statistical ideas:

1. Calculations demonstrate an understanding of statistical ideas: Emphasis should not be placed on calculations as the ability to do a calculation is not the same as understanding a calculation.

2. Formulas help students understand the statistical idea: Using a formula to convey a statistical idea is emphasising the mathematics and not the concept. Statistical ideas should be taught, at first, without a formula and the student should be encouraged to find or derive a formula that demonstrates that they understand the concept and not purely the mathematics. Students should also be able to explain how to do something in words that demonstrate the meaning of the measurement or statistic. This sentiment is echoed by Cobb and Moore [4] who discuss teaching statistics from two perspectives, that is, content and pedagogy. They suggest that introductory statistics courses should focus on statistical ideas instead of copious amounts of theory and formulas explained using a purely mathematical approach. Statistical tools common to statisticians should be used flexibly with mathematics to deal with problems; there should be an emphasis on an intellectual problem-solving method using reasoning to approach the problem. They then go on to suggest that introductory statistics courses should not contain lessons on formal probability theory. It is difficult to develop understanding with the basis in a concept such as probability theory with deep roots in mathematics. The recommendation is to teach informal probability instead because it is sufficient for the student to gain an understanding of inference. The derivation of distributions should be taught to students at a more advanced level. Distributions should be taught using data analysis tools instead of following mathematical rules to solve the probability problem.

3. Students who can explain things in statistical language demonstrate their understanding of a statistical idea: Students should be able to discuss a result using plain and meaningful language in addition to being able to give a statistical interpretation of the result. There should be an emphasis on why statistics is done and on what is the thing that must be achieved.

Rumsey goes on to provide suggestions to promote understanding and encourage statistical literacy:

1. Teaching tools and techniques of problem solving should be accompanied by reasons for its use and examples on how to use it.

2. Definitions used should be made easy to grasp.

3. Overarching ideas should be communicated rather than simply knowledge learned in a single context.

4. The language and technical terms used should be moderated and supplemented with terms or methods that convey the bigger picture.

2.3 Applying statistical thinking models to teaching and learning

After expanding on statistical thinking models and exploring the challenges of teaching introductory statistics courses, it is clear that there is some divergence in the way that statistics is currently taught and the ideal outcomes of teaching which is to create statistically literate students who can engage in statistical thinking. From the 4-dimensional framework and the 5 types of thinking, it is clear that statistical thinking would require a student to develop characteristics that encourage an enquiring and problem-solving mind, learn

how to approach problems and, furthermore, the student should also be equipped with tools and skills that would enable them to solve such problems. It is key here to note that the aforementioned attributes must be learnt and is not innate for most students. This would then put the onus on teachers to transfer such attributes to students and the best time would be when creating a foundation in introductory statistics modules.

Pfannkuch and Wild [8] considered contributions from statistics education researchers and integrated the 5 types of thinking with what they uncovered to ascertain how statistical thinking can be taught:

1. Integrating the statistical and contextual: Students need to assume the role of a "detective" and examine the data along with the context to discover the reasons for the presentation of the statistics. The data must be questioned by the student and the student should follow the process to reach a judgement or decision about the problem.

2. Transnumeration and context knowledge: Statistical thinking is propagated through the critique of graphs and patterns and not merely the acknowledgement of these representations. Collection and modelling of the data also employ the use of statistical thinking and can be valuable to the students' learning.

3. Recognition of the need for data: Students may assume that the data and analysis of the data will fit their perceptions and experiences of that or a similar situation. This reliance on their opinions introduce a bias and so a shift in thinking is needed for the student to identify a need for the data.

4. Statistical thinking and interacting with statistically based information: While the inclusion of the critical evaluation of statistically based reports in teaching curricula places the student in an investigative position and allows the student to employ high-level thinking; this alone is not sufficient. The teaching of the evaluation of such reports is required so the student may fully develop their statistical thinking to synthesise statistical knowledge into the context.

5. Probabilistic and deterministic thinking: These two paradigms in statistics need to be taught together as using both in models reveal information not available by analysis that treats these concepts as separate.

6. Variation as fundamental in statistical thinking: Variation is under-represented in curricula. Students need to develop a proper grasp of the concept of variation as this concept is foreign to the student at the outset.

Using statistical thinking models and considering the suggestions of statistics educators and researchers, the following would be required from students enrolled in introductory statistics courses and can be referred to as teaching outcomes to foster statistical thinking:

1. Students should complete practical projects and assignments that use real-world data or reported statistics and should be able to critique and provide insight into the problem and further follow steps to solve the problem using the tools and skills they have learnt in the course. This is the application of the 4-dimensional framework and will encourage statistical thinking and measures their level of statistical literacy.

2. Students should work with graphs and representations of data. They should be able to model the data and critique the data.

3. Students should be able to show how they can apply what they have learnt in real-life scenarios and not simply understand what they have learnt in a purely theoretical context.

4. Students should be able to understand what a formula does and how it works. Less emphasis should be placed on simply remembering and using formulae. It should be the focus for the student to be able to convey the idea behind the formula using plain language.

5. Students should be able to evaluate the results of their problems and calculations in context and they must be able to explain and make sense of this given the results.

6. Students should show that they understand the importance of variation in data and must be able to explain the concept of variation, as well as other statistical concepts, in plain language.

7. Students who struggle with mathematics tend to be impeded in their learning of statistics if the teaching approach is purely mathematical. The course should be structured in a way that allows the student to use statistical tools and mathematics together to help the student understand.

8. Students should be able to use available technology to enhance their learning, understanding and application of concepts.

9. Discussion and the use of multimedia should be used to enhance the students' experiences in the course.

If the students can fulfill the requirements, the students can then be said to be able to think statistically, based on the models discussed. Teaching outcomes should be directed at equipping the student to think statistically and to equip the student for dealing with problems outside the confines of the classroom – from statistics reported in the media to working with statistics in their later studies or careers.

In the next section, responses from students will be analysed in order to ascertain whether an introductory statistics course offered at the University of Pretoria, henceforth referred to as STAT 1, produces students that can think statistically and are statistically literate.

3 Application

STAT 1 is a first level statistics course offered at the University of Pretoria. The course is taken by students enrolled in a wide range of degrees and belonging to different faculties. Most students do not continue this course beyond first year level, therefore, STAT 1 serves as the only statistics instruction they receive in an academic setting. In terms of introductory statistics courses, it has been determined that this course is a good candidate for this analysis. The main goal of this analysis is to determine whether, after taking the course, the students are able to think statistically and if they gained meaning from the work covered. To achieve this end, the analysis was approached using both quantitative and qualitative methods. The perceptions of the students were the main focus of this analysis as these were used to determine whether students are able to engage in statistical thinking and if the course encourages statistical thinking.

3.1 Description

The STAT 1 study guide states the course outcome as follows: "The goal of this statistics course is to equip you with the basic knowledge and skills concerning the most important statistical techniques used daily in practice". From this, it is clear that the course is meant to teach students skills that are transferable to their daily lives. This implies an emphasis on statistical literacy and practicality. The student should be able to engage in statistical thinking after completing the course if the course content and structure lend itself to this. The students attend three theory classes, one tutorial, and one optional practical class per week where each class or session is 50 minutes long. The theory classes follow the usual lecture format where topics such as probability theory, descriptive statistics and statistical distributions and inference are covered. In the tutorial session, students are required to have completed an exercise based on current work beforehand so solutions can be discussed and problems can be addressed during the session. There is one practical session per week where attendance is not compulsory if students are able to complete the work on their own and do not require further assistance. The practical exercises must be completed using Microsoft Excel. The exercises are based on the theory covered in class. In addition to classes, students are required to complete graded Aplia assignments online before and after a block of classes. Aplia is an online assignment environment, covering various subjects, designed to link theoretical concepts and real-world examples in the form of assignments and exercises. Students are also required to make use of a clicker during classes and tests. A clicker is a device that is linked to the lecturer's computer and is used to measure the responses of questions. During class, the lecturer may display a question on the screen that should either be discussed in groups or pondered over individually thus creating an interactive environment. The student then enters their answer

and the correct answer, along with a graphical display of the answers given, is displayed on the screen. The students also use their clickers during tests and examinations, under normal examination conditions, where the device is used to record their answers. The students are subject to continuous assessment throughout the semester and they write a final examination at the end of the course. The semester assessment includes two semester tests weighing 25% each, a practical test weighing 10%, clicker tests weighing 10% in total, and Aplia and a test on hypothesis testing weighing 10% each. The final examination weighs 50% or 60% of the final mark and the remainder is the mark received for the semester assessment. A final mark of 50% is required in order to pass the course. The students are able to access resources from the library and are able to consult with their lecturers and tutors should they require assistance with the coursework.

From the course outline, it is seen that the students are exposed to theory, they practise using examples and apply the work using technologies such as Excel, and the students also engage with real-world examples using Aplia. The work is presented using a blended learning - specifically flipped classroom - approach and there are enough support resources available to them should they run into problems. Blended learning means that instruction is a combination of face-to-face teaching and online learning. Flipped classroom is a form of blended learning where the student is required to do online preparation before a face-to-face session. In STAT 1, students are required to do reading in the Aplia environment before a class and complete a pre-class Aplia assignment and a post-class Aplia assignment that forms part of online learning. From this point of view, the course seems to provide a holistic approach. However, the question is whether this approach has the right mix of components to develop statistical thinking. For the purpose of this analysis, a survey of a group of students enrolled in the course was taken with the questions designed to measure if the student's opinions on what they have learnt are in line with the ideal outcomes of teaching and learning to facilitate and develop statistical thinking. The survey was administered during a practical revision class in the last week of lectures for the semester. The students were asked to fill in a survey containing the following 11 statements:

1. What I have learnt thus far in STAT 1 is relevant to my life outside university.
2. I think more critically about reported statistics in the media.
3. I think the work covered is practical.
4. I can see how the distributions I have learnt can be applied to real-world scenarios.
5. When I study, I learn how to use the formula but I do not know what the formula means.
6. I understand the link between hypothesis testing and statistical inference.
7. I gain meaning about the problem context from the conclusions I draw from hypothesis testing.
8. Applying the work in Excel has helped me understand the work better.
9. I find the mathematical component of statistics difficult.
10. Clickers assisted me well to master STAT 1.
11. Aplia assignments assisted me well to master STAT 1.

Students were requested to mark a box that indicated to what extent they agreed with each statement, anchored from 1 - Strongly disagree to 10 - Strongly agree. The statements were designed to gauge information about the students' perceptions of the course content and more importantly, the approach of the course.

- Statement 1 was designed to measure if the student believes that what he/she has learnt in the course has any real-world application and if they think the skills they have acquired are transferable to daily life. Statement 1 also tests if the student can recognise the need for data - which is the first thinking type. If a student recognises the importance of data in their field of study or in real-world scenarios, then they are able to engage with the first thinking type.

- The response to statement 2 would indicate if the student has learned to critically analyse the data they come into contact with in media and statistical reports thus also measuring if the student is able to apply the second thinking type - transnumeration which involves being able to critically analyse patterns and graphical representations of data.
- Statement 3 would indicate if the student perceives the work as practical enough or if there is, in his/her opinion, too much emphasis on theory. This statement reveals information about whether the student can see the applications of the theory thus engaging with thinking type 5 - integrating the statistical and contextual.
- Statement 4 will allow for gauging if the student can directly see how something specific they have learnt can be used in real life. Similar to the case of statement 3, statement 4 also measures how well students engage with thinking type 5. In addition to this, statement 4 measures if the student is able to reason with statistical models which is the essence of thinking type 4 - reasoning with statistical models.
- Statement 5 is important in that it indicates if the student understands the calculations or if they simply follow recipes. The response to this statement is a very good indication of whether the student actually understands the work being presented and while this does not relate to any specific type of thinking, understanding is essential in statistical literacy and statistical thinking as a whole.
- Statements 6 and 7 are both an indication if the student is able to get contextual understanding from problems they are required to solve. Statements 6 and 7 relate to thinking type 5 - integrating the statistical and contextual which is a fundamental element of statistical thinking as the application of the other thinking types is also required.
- The students' responses to statement 8 will serve as an indication of whether the student can effectively use technologies to make sense of the data. This statement will give an indication of whether the student can engage with thinking types 1 and 2 which are recognition of the need for data and transnumeration.
- Statement 9 was designed in order to ascertain if the student finds that a mathematical approach would impede their progress. This statement does not relate to statistical thinking but can be used to explain the performance of a student and will provide insight into whether the course should be structured to incorporate a less mathematical approach.
- Statement 10 should provide information as to how the student responded to an interactive approach during lectures. Statement 10 does not link to a specific thinking type but rather gives information on whether using multimedia and interactive technologies enhance a student's performance.
- Finally, statement 11 should indicate whether students perceived the Aplia assignments as useful in assisting them to understand the work better. Statement 11 concerns Aplia which, as explained earlier, requires the student to answer questions posed in the form of real-world problems. The students' responses to this statement will indicate if they can integrate the statistical and contextual - thinking type 5 - and also, depending on the question can measure how well they engage with the other thinking types. The reason for this is that the questions may require students to work with data in Excel, make graphs, and answer theoretical questions.

The statements in the survey were designed to measure whether the student can engage with the 5 types of thinking explicated in the literature. The third type of thinking, consideration of variation, was not addressed in this survey but left to the qualitative study. The reason is that an explanation of variation would be better suited to the purpose of investigating the students' understanding of the concept. The survey also incorporates the suggestions and recommendations for the instruction of statistics so as to encourage understanding. The responses to these statements provide an idea as to whether STAT 1 is structured and taught in a way that encourages understanding, statistical thinking and literacy.

3.2 Methodology

From the surveys collected, there were 119 students who granted permission for their marks to be used for the purpose of this analysis and these surveys will be used for the statistical analysis. The goal of this approach was to determine if the marks the students have achieved are an indication of their levels of statistical thinking and further, if the course does develop statistical thinking in the students. The data was first summarised to obtain descriptive statistics. Correlations between the variables were analysed to cast more light on this issue and to assess if multicollinearity is present. Multiple regression was used to model the progress mark of the student as the dependent variable and their response to each statement as independent variables. The question of whether the student was repeating the module was also asked and this served as a qualitative independent variable.

The survey was based on a Likert scale with an underlying assumption of continuity. This combined with the fact that a 10-point scale was used means that the responses can be viewed as one would percentages. Descriptive statistics methods were thus applied to the data collected and the results obtained are explained below. The mean of each statement was calculated as well as the mean value of each statement for each group of students with similar marks. The marks referred to henceforth are the students' final semester marks for STAT 1. The groups were divided as follows: Students scoring a distinction, that is with a mark of 75% or higher, formed the first group; students in the second group are those that passed but did not achieve a distinction with marks ranging from 50% to 74%; students that fell into the last group scored marks below 50%, i.e. failed the module.

There were 5 missing values in total which occurred completely at random. In the cases where values were missing, the responses were sorted sequentially so as to obtain the most similar results across the statements. The missing values were then imputed as the average of the value of the response above and below the cell in question. The mean of each statement, including the imputed values, differed only very slightly in the third or fourth decimal if at all affected. Based on this, the dataset with the imputed missing values was the one used in the analyses conducted.

The analyses were conducted using SAS® software[1]. A bivariate correlation analysis was performed, followed by a multiple regression analysis, an analysis of partial correlations, and a stepwise regression analysis. The output and code can be found in the Appendix of this report.

3.3 Results

Having applied the methods described above, the results of the analyses are henceforth explained.

[1] The data analysis for this essay was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA.

Statement	Overall mean n = 119	Group 1 (75+) $n_1 = 26$	Group 2 (50 - 74) $n_2 = 69$	Group 3 (< 50) $n_3 = 24$
1	4.42	4.88	4.55	3.54
2	5.38	6.5	5.17	4.75
3	5.54	6.31	5.60	4.63
4	5.72	6.42	5.54	5.5
5	5.22	3.85	5.54	5.79
6	5.06	5.54	5	4.73
7	5.79	6.35	5.62	5.65
8	4.83	5.04	4.99	4.15
9	4.63	3.04	4.90	5.58
10	4.36	4.54	4.74	3.08
11	5.66	6.62	5.64	4.67

Table 1: Mean response to each statement

The mean responses to each statement are summarised in Table 1. Across all divisions, it is clear that there is no overwhelmingly high or low mean response that indicates extreme polarisation. Looking at the statements, a positive result will be for the responses to all statements to be as close to 10 as possible barring statement 5 and 9 where a response close to 1 would be ideal. Overall, students slightly agreed with statements 2, 3, 4, 5, 6, 7 and 11 and slightly disagreed with the remaining statements. None of the mean responses are very strong. When the statements are examined individually, across mark groups, a pattern emerges. The mean response for each statement decreases as the achievement group indicates that the students' marks are decreasing. The opposite is true for statement 5 and 9 where the mean response increases as the mark group indicates that the students received lower marks. This pattern is indicative that students with higher marks agreed more with the statements where a positive response would be ideal and they agreed less with statements where a negative response would be ideal. The converse is true for students with lower marks. It thus seems as if students who perform better gained more from the course in terms of the objectives of statistical thinking.

The mean responses to statement 1 indicate that no group strongly perceives the course as being relevant to real-life. The mean values indicate that students disagree that STAT 1 is relevant to their lives outside university. The responses to statement 2 show that groups 1 and 2 consider themselves to think more critically of reported statistics with group 1 agreeing more strongly. Group 3 students moderately disagree with this statement. Statement 3 required the student to determine if they thought that the work covered was practical and the responses have the same outcome and pattern as the responses to statement 2. Statement 4 received a positive mean response from all groups with the responses implying that the students in every group agreeing that they are able to see how the distributions they have learnt can be applied to real life. From statement 5, a response that indicates disagreement would be the desired outcome as statement 5 stated that the student knows how to use the formula but the student does not understand the formula. Only group 1 disagreed with statement 5 with groups 2 and 3 falling into the agreement category, however moderately. This indicates that the students probably perform the calculation mechanically without grasping the concept fully. The mean responses to statement 6 indicate moderate agreement from group 1 and moderate disagreement from groups 2 and 3 to the statement that the student is able to link hypothesis testing and statistical inference. There is agreement from all groups on statement 7 that the conclusions the student draws from hypothesis testing give meaning to the context of the problem. Statement 8 implied that by applying the work in Excel, the student would understand the work better. The mean responses indicate disagreement with this statement with only group 1 agreeing although very moderately. If the response to statement 9 were near 10 it would indicate that the student perceives the mathematical component of statistics to be difficult. The mean response from group 1 indicates that this group disagrees with this statement. Group 2's responses indicate that this group is only very slightly disagrees while the mean response from group 3 indicates that the students in this group found the mathematical component difficult on average. From the mean responses to statement 10, all groups disagree that clickers assisted them well to master the course with

group 3 disagreeing the most strongly on average. The last statement, statement 11, would be indicative of whether completing the Aplia assignments assisted the student to master the work. The mean responses from groups 1 and 2 indicate that these groups agree while group 3 disagrees albeit slightly.

A correlation analysis and regression analysis were performed on the same data; the statistical results can be found in the Appendix of this report. The results from the correlation analysis indicated that there is no presence of multicollinearity among the predictor variables. The highest correlation occurred between the variables for statements 6 and 7 with a Pearson correlation coefficient of 0.62053. There was also no strong correlation between the dependent variable and any of the other predictor variables. The highest Pearson correlation coefficient was 0.30141 for the dependent variable, Mark, and the independent variable representing statement 2 (which measures critical thinking). The point biserial correlation coefficient for the bivariate correlation between Mark and the dummy variable indicating whether or not a student is repeating the module was positive. Similarly, the Pearson correlation coefficients for the bivariate correlations between Mark and the statement variables were positive except for the variables associated with statement 5 and 9, as was expected.

From the regression analysis, the model is seen to be significant at a 5% level of significance with an F-value of 2.12 and a p-value of 0.0212 i.e. a statistically significant amount of variation in the dependent variable, Mark, can be explained by the combination of independent variables in the model. The R-squared value of the model was 0.1937, indicating that only 19.37% of the variation in the data can be explained by the model and the adjusted R-squared value drops to 0.1024 indicating the penalty for the number of predictors. The output from the analysis of variance in the regression analysis is provided in Figure 2.

STK 795
Megan Sajiwan, 11065177
Regression analysis

The REG Procedure
Model: MODEL1
Dependent Variable: Mark

Number of Observations Read	119
Number of Observations Used	119

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	12	4793.77405	399.48117	2.12	0.0212
Error	106	19961	188.31004		
Corrected Total	118	24755			

Root MSE	13.72261	R-Square	0.1937
Dependent Mean	62.16807	Adj R-Sq	0.1024
Coeff Var	22.07341		

Figure 2: Regression procedure output for the analysis of variance

As previously stated, all the statements were taken as independent variables as well as the independent variable indicating if the student is repeating the module. The output from the regression analysis with the details of parameter estimates can be found in Figure 3.

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	52.30437	7.66733	6.82	<.0001
rep	1	1.51861	4.81302	0.32	0.7530
S1	1	0.78102	0.73091	1.07	0.2877
S2	1	0.91669	0.77680	1.18	0.2406
S3	1	0.53125	0.76267	0.70	0.4876
S4	1	-1.08901	0.65818	-1.65	0.1010
S5	1	0.11550	0.48326	0.24	0.8116
S6	1	0.29102	0.66921	0.43	0.6645
S7	1	-0.00249	0.80095	-0.00	0.9975
S8	1	0.35163	0.58187	0.60	0.5469
S9	1	-0.99599	0.51304	-1.94	0.0549
S10	1	0.19195	0.53404	0.36	0.7200
S11	1	0.59598	0.53639	1.11	0.2690

Figure 3: Regression procedure output for parameter estimates

In the regression analysis, when looking at the predictors, the predictors representing statements 4 and 9 were the only predictors that are significant. According to the significance from p-values in Albright et. al [1], the predictor representing statement 9, s9, shows strong evidence of significance with a p-value of 0.0549. Looking to the regression coefficient, the parameter estimate for s9 of -0.99599, which indicates the impact of statement 9 controlling for the other predictors, has a negative sign. Given that the p-value shows strong evidence of significance, this indicates that, as the students' level of agreement with statement 9 increases, the students' marks decreased. Statement 9 was given as: *I find the mathematical component of statistics difficult.* This indicates that, controlling for all other variables, the more the student struggles with mathematics, the poorer their marks would be. The predictor representing statement 4, s4, was moderately significant with a p-value of 0.1010. The parameter estimate of s4 was -1.08901 which has a negative sign despite the bivariate correlation coefficient between Mark and s4 having a positive sign. Multicollinearity was investigated earlier and it was shown that there is no multicollinearity present to explain the different signs. To try and understand this anomaly, the partial correlations were calculated and the sign of the partial correlation was negative, matching that of the regression coefficient. The statement for s4 was: *I can see how the distributions I have learnt can be applied to real-world scenarios.* Due to the moderate significance of s4, this can be interpreted as the students' ability to apply the work increases, the students' mark decreases by more than 1 percentage point, holding the other variables constant. The predictor with the highest p-value (0.9975) was s7, indicating that it has the least impact on the dependent variable, Mark.

To better evaluate which statements impact on marks, if considering the statements one-by-one, a stepwise regression analysis was conducted. The stepwise regression was performed at a 5% level of significance. At this level, s2 entered the model in the first step. In the second step, s9 entered the model. These were the only two variables to meet the 0.05 significance level for entry into the model. The F-value for the test of model significance was 11.69 accompanied by a p-value of 0.0009 thus indicating that the model is significant. With a p-value of 0.0067, s2 is significant at a 1% level of significance while s9 is significant at a 5% level of significance (p-value = 0.018). The predictor s9 was significant in the original regression analysis, however, s2 was not. Statement 2 was given as: *I think more critically about reported statistics in the media.* From the stepwise regression results, as the students' agreement with the statement increases, the students' average mark increases by 1.48602 percentage points, controlling for s9. As in the original regression analysis, the sign of the regression coefficient for s9 is negative. In the stepwise regression, the parameter estimate for s9 is -1.06957 implying that holding s2 constant, as the students' level of agreement with the statement increases,

the statement being that they find the mathematics component of statistics difficult, the students' average mark decreases by 1.06957 percentage points.

These regression results indicate that the students' understanding of statistics plays little part in determining the marks they are awarded in the course. This is perhaps not unexpected since the bulk of the assessments, tests, and the examination, are designed to assess students' theoretical knowledge and not their intuition or their insight into the problem. However, it should also be noted that only the statement responses were used as independent variables in addition to the dummy variable indicating whether or not the student was repeating the module. There are many other factors that contribute to students' marks, for example, the performance of the student in grade 12 mathematics; whether or not the language of instruction is the student's home language. The model developed in this essay does not perform particularly well and these factors should be taken into consideration and the survey should be improved on.

3.4 Qualitative study

To get a firmer grasp on the perceptions of students, interviews were conducted with students with the purpose of determining their level of understanding after having completed the introductory statistics course, STAT 1. The interviews were conducted on a voluntary basis with seven students having participated. All seven students belong to the Faculty of Economics and Management Sciences and are studying finance and business related degrees. The interviews were conducted individually and each student was asked the same core questions and possible follow-up questions that are listed as follows:

1. Do you think what you have learnt thus far in STAT 1 is in any way relevant to your life outside of university?
    (a) Did the way you view reported statistics change in any manner?
2. Do you think the work covered is practical?
    (a) Can you give any examples or illustrate how you would apply any of the distributions you learnt to a real-world scenario?
3. What is statistical variation?
4. When you study, do you simply learn how to use to formula or would you say you understand what the formula means or represents?
    (a) Looking at the following formula $\frac{1}{n-1}\sum(x_i - \bar{x})^2$ can you explain what it means?
5. What is the link between hypothesis testing and statistical inference?
    (a) Do you feel you gain any meaning from the conclusions that you draw?
6. Has applying the work in the practical classes using Excel helped you with understanding the concepts?
7. Do you find the mathematical component of statistics difficult?
    (a) Do you feel that mathematics hampers you or helps you understand the work?
    (b) Can you make the distinction between mathematics and statistics?

The questions asked in the interview were linked to the application of the 5 types of thinking to the teaching and learning of statistics. The interview questions have a similarity to that of the statements in the survey. The responses to the questions were collected and compared and are summarised below.

None of the students indicated that what they have learnt in the module is relevant to their lives outside of university. They indicated that they were not able to see how it related to their fields of study or the degree

for which they were enrolled. When asked the follow-up question of if the way they viewed reported statistics changed in any manner, three students agreed that it did. The view of these students was that it matters how the statistics are reported as it can be manipulated; another student suggested that taking the course made him more sceptical of the reported statistics he comes across. One student linked the reported statistics to the field of economics where statistics is often employed in analyses and concluded that she was able to better engage with the statistics in that field after completing the module. The other students expressed disinterest in following current events and media in which statistical reporting is common such as politics and economics.

In response to question 2, all but two of the students agreed that the work covered is practical, meaning that they are given the opportunity to apply the theoretical concepts in examples. The two students that disagreed explained their response by suggesting that they did not work with real life examples. The follow-up question required of the student to give an example or illustrate how they would apply any of the distributions they learnt to a real-world scenario. In the case where they were not able to do so, the students were asked to give an example of how statistics can be applied in a real-world scenario. The students gave examples of hypothesis testing; using the normal distribution to check if marks obtained in a course were normally distributed; using descriptive statistics in market research; applying statistics in economics to calculate the GDP; keeping track of sales.

One of the 5 types of thinking is 'Consideration of variation'. Question 3 was asked with the aim of gauging how well students perform in this thinking type. None of the students claimed to understand variation or its role in the real-world. The students did have inklings of what variation could be. Two students suggested that it is because everything varies and one cannot say that one situation will be the same as before; this being a rather vague suggestion that samples will differ due to variation. Two other students answered with a formulaic approach describing variance as the distance from the mean. The students could not convey the concept of variation clearly nor state why variation is important in statistics.

The responses to question 4 should provide an idea of the students' conceptual understanding. As explained in the literature, students should be able to explain concepts and not simply rote learn formulae as this does not encourage understanding. Only three of the seven students suggested that they try to understand the formula but no students suggested that they were taught where a formula comes from or what it means or measures. The students who indicated that they try to understand the formula stated that they did so because it is important as if you understand then it makes studying the concepts easier. The responses imply that students look at a formula and learn how to use it because that is all that is required of them. To cast more light on the students' ability to understand a formula, the students were shown the formula for sample variance and asked to explain what it means or measures. The students were all able to identify the symbols and variables, for example: n is sample size. Two students recognised the formula as sample variance. From all the students, only one was able to give an adequate description of what the formula measures; two others implied that it measured the distance between the observations and the mean although not very clearly.

Question 5 and its follow-up question aimed to test if students are able to link the context to a problem and if they build understanding during the problem-solving process or if they simply follow the steps of the process. The students were asked if they understand the link between hypothesis testing and statistical inference. All the students were able to describe the link and further agreed that they gain meaning of the problem context after completing the hypothesis testing process. This is a good indication that they are able to analyse a problem in context and use statistical methods in their problem solving.

As part of the course, the students are expected to be able to use Microsoft Excel to apply the work and make calculations, create tables and charts, and use this to solve problems. The interviewees were asked if applying the work in Excel helped them with understanding the concepts. Three students agreed that applying the work in Excel helped them understand the concepts but the others all disagreed. The students that agreed suggested that applying the work in Excel was helpful as it brought clarity to the concepts and the calculations. Those who disagreed suggested that they felt that applying the work in Excel was more difficult.

Question 7 asked the students if they found the mathematical component of statistics difficult. None of the students stated that they found the mathematical component of statistics difficult and further, they all stated that mathematical calculations helped them with the work. All but two of the students could make the distinction between mathematics and statistics. This result is opposite to that generated in the quantitative analysis.

At the end of the interview, each student was asked if they have any comments about the course or any suggestions to improve the course. Most of the students requested that the course be made more relatable to their fields of study. Further, the students would like the relevance of the content to be covered in addition to the teaching of the content. One important suggestion was to assign the students a project where they are able to work with data from their field of study.

The responses from the interviews mostly support the results of the quantitative study. It should be noted that the number of students that participated in the interview was very small compared to the number of enrolled students. Further, the performance of the participating students was average or higher in comparison to the whole class as all students fell into group 2 meaning they all scored a mark between 50% and 75%. Whilst these responses provide more insight into the perceptions of the students, it is not an accurate representation of the overall perceptions of the course.

From the quantitative and qualitative analyses, albeit limited, it can be seen that the course is not very effective at conferring statistical thinking. The students' performance in the module is not a good indication of whether the students are able to think statistically. The module should be revised so that it teaches students statistical concepts but also conveys the relevance of the concepts as well as how these concepts can be applied to real-world scenarios and how technology can assist in solving statistical problems. Students should work with statistical models and not only learn how to develop and use the models but also how to use the tools learnt to gain valuable insight into problems thus allowing the students to engage in statistical thinking fully.

4 Conclusion and recommendations

This essay covered the literature pertaining to statistical thinking with specific emphasis on statistical thinking in pedagogy. It was established that developing statistical thinking is of vital importance in introductory statistics courses where the foundations of problem solving are developed. The 4-dimensional model which includes the 5 types of thinking was the central focus in the literature examined in this essay. These models were then applied to an introductory statistics module offered at the University of Pretoria where quantitative and qualitative studies were conducted with the main aim of ascertaining whether or not the course was successful at fostering statistical thinking in the students enrolled in the course.

It is important to note that both the quantitative and qualitative studies were limited. Regarding the quantitative study, the multiple regression model that was developed did not perform very well in predicting the students' final marks. To improve on this, variables that can play a large role in influencing a student's performance in the course could be taken into account, for example: the students' aptitude for mathematics and their performance in mathematics at a secondary education level; whether or not the language of instruction is the same as the students' home language; the educational resources a student has access to; and so on. However, the aim of this model was not to predict marks per se, but rather to evaluate the effect of the students' perceptions regarding the course as measured by the 11 statements in the survey. To this end, the model serves its purpose. Furthermore, the questions included in the survey should be revised for simplicity and clarity. Statements that were long and those that included terminology or phrases that the students might not be fully acquainted with may have had an effect on the response provided for that statement. The main shortfall of the qualitative study was the process of selection. It would have been ideal to interview students whose performance fell into the following categories: 'Top', where the students score 75% or more in the course; 'Middle' which would include students scoring in the range of 50% to 74%; and

'Bottom' which would include those students who scored lower than 50%. These categories match that of those used in the quantitative study. The students did not respond to requests for the interviews and so a volunteer-based approach was taken where interviews were conducted with students who were willing and thus it was not possible to do a comparison of responses to performance. It can probably be said that the students that volunteered were those that were enthusiastic about the course (given by their performance) and so the responses are not representative of the perceptions of the collective group. The qualitative study can be improved on by conducting interviews with more students and ensuring that the sample is demographically representative as well as unrestricted in terms of performance of the students.

The results from the quantitative and qualitative studies both indicate that the course does not perform well with regards to teaching statistical thinking. This is largely due to the traditional assessment structure of the course where a student is assessed on how well they use the statistical tools available to them but not necessarily if they understand why such a process is necessary in the larger context of the problem. The module was not completely unsuccessful in achieving this end, however, but there are clear areas for improvement. To convey the meaning of concepts, there should be a discussion of the relevance of each of the topics before the intricacies and methods are presented. This should ensure that the student is aware of why such a model is being developed or the reason for the calculations. To aid in the understanding of concepts, formulae should be taught in addition to what the formula is measuring and the conceptual development of the method or formula. The students should be able to explain important statistical concepts in their own words as well as in mathematical terms, for example, variation which is a key concept in statistics. The most important suggestion for improving the course is to include a project in the assessment structure. The students should go through the process of collecting or using real-world data that is relevant to their specific fields of study or in a field that interests them. They should be required to work with the data and generate descriptive statistics, represent the data graphically and perhaps model the data or carry out a hypothesis test. Students should be encouraged to make use of Excel and any other technological resource available to them. Finally, the students should be able to provide meaningful insights and conclusions to the problem they have been working on. By completing the project, the students would have had to engage with most of the processes described in the 4-dimensional model. Further, by using data that is relevant to them, the students will be able to relate to the problem and hopefully, gain meaning and understanding and ultimately, the student will be exposed to the process of statistical thinking. In addition to the project, another important suggestion is to include understanding as one of the main goals for the course.

Statistics is an incredibly broad field that can be applied in countless areas from medicine to marketing and is encountered in daily life and professional life. The students that enroll in introductory statistics courses must be exposed to the tools that will allow them to analyse data and make calculations but most importantly, the students must be able to solve problems and think critically - two important skills that cannot be acquired without understanding.

References

[1] S.C Albright, W.L. Winston, and C. Zappe. *Data analysis and decision making with Microsoft Excel*, page 503. Cengage Learning, 3rd edition, 2009.

[2] D. Ben-Zvi and J. Garfield. Statistical literacy, reasoning, and thinking: Goals, definitions, and challenges. In *The Challenge of Developing Statistical Literacy, Reasoning and Thinking*, pages 3–15. Springer, 2004.

[3] B.L. Chance. Components of statistical thinking and implications for instruction and assessment. *Journal of Statistics Education*, 10(3), 2002.

[4] G.W. Cobb and D.S. Moore. Mathematics, statistics, and teaching. *The American Mathematical Monthly*, 104(9):801–823, 1997.

[5] C. Mallows. The zeroth problem. *The American Statistician*, 52(1):1–9, 1998.

[6] D.S. Moore. New pedagogy and new content: The case of statistics. *International Statistical Review*, 65(2):123–137, 1997.

[7] M. Pfannkuch and C.J Wild. Statistical thinking and statistical practice: Themes gleaned from professional statisticians. *Statistical Science*, pages 132–152, 2000.

[8] M. Pfannkuch and C.J. Wild. Towards an understanding of statistical thinking. In *The Challenge of Developing Statistical Literacy, Reasoning and Thinking*, pages 17–46. Springer, 2004.

[9] D.J. Rumsey. Statistical literacy as a goal for introductory statistics courses. *Journal of Statistics Education*, 10(3):6–13, 2002.

[10] R.D. Snee. Discussion: Development and use of statistical thinking: A new era. *International Statistical Review*, 67(3):255–258, 1999.

[11] C.J. Wild and M. Pfannkuch. Statistical thinking in empirical enquiry. *International Statistical Review*, 67(3):223–248, 1999.

Appendix

Code

The regression analysis and correlation analysis was carried out using SAS® software. The program used is given below.

Correlation analysis and regression analysis:

```
options nodate ps = 5000 pageno = 1;
title1 "STK 795";
title2 "Megan Sajiwan, 11065177";

data project;
set sasuser.meg;
if repeating = 'No' then rep = 1;
if repeating = 'Yes' then rep = 0;
run;


title3 "Correlation analysis";
proc corr data = project;
var mark rep s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11;
run;


title3 "Regression analysis";
proc reg data= project;
model mark = rep s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11;
run;
```

Partial correlation analysis:

```
title3 "Partial correlations";

title4 " ";
title5 "repeating";
proc corr data=project;
var mark;
with rep; partial s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11;;
run;

title5 "s1";
proc corr data=project;
var mark;
with s1; partial rep s2 s3 s4 s5 s6 s7 s8 s9 s10 s11;
run;

title5 "s2";
proc corr data=project;
var mark;
with s2; partial rep s1 s3 s4 s5 s6 s7 s8 s9 s10 s11;
run;

title5 "s3";
proc corr data=project;
var mark;
with s3; partial rep s1 s2 s4 s5 s6 s7 s8 s9 s10 s11;
run;

title5 "s4";
proc corr data=project;
var mark;
with s4; partial rep s1 s2 s3 s5 s6 s7 s8 s9 s10 s11;
run;
```

```
title5 "s5";
proc corr data=project;
var mark;
with s5; partial rep s1 s2 s3 s4 s6 s7 s8 s9 s10 s11;
run;

title5 "s6";
proc corr data=project;
var mark;
with s6; partial rep s1 s2 s3 s4 s5 s7 s8 s9 s10 s11;
run;

title5 "s7";
proc corr data=project;
var mark;
with s7; partial rep s1 s2 s3 s4 s5 s6 s8 s9 s10 s11;
run;

title5 "s8";
proc corr data=project;
var mark;
with s8; partial rep s1 s2 s3 s4 s5 s6 s7 s9 s10 s11;
run;

title5 "s9";
proc corr data=project;
var mark;
with s9; partial rep s1 s2 s3 s4 s5 s6 s7 s8 s10 s11;
run;

title5 "s10";
proc corr data=project;
var mark;
with s10; partial rep s1 s2 s3 s4 s5 s6 s7 s8 s9 s11;
run;
```

Stepwise regression analysis:

```
title5 "s11";
proc corr data=project;
var mark;
with s11; partial rep s1 s2 s3 s4 s5 s6 s7 s8 s9 s10;
run;

title3 "Stepwise Regression analysis";
title4 " ";
title5 "alpha=0.05";
proc reg data= project;
model mark = rep s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 /selection=stepwise
sle=0.05;
run;
```

Output

The output from the program is given below.
Descriptive statistics:

STK 795
Megan Sajiwan, 11065177
Correlation analysis

The CORR Procedure

13 Variables:	Mark rep S1 S2 S3 S4 S5 S6 S7 S8 S9 S10 S11

Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
Mark	119	62.16807	14.48396	7398	29.00000	93.00000
rep	119	0.91597	0.27861	109.00000	0	1.00000
S1	119	4.42017	2.39171	526.00000	1.00000	10.00000
S2	119	5.37815	2.40406	640.00000	1.00000	10.00000
S3	119	5.53782	2.35707	659.00000	1.00000	10.00000
S4	119	5.72269	2.60353	681.00000	1.00000	10.00000
S5	119	5.21849	3.05919	621.00000	1.00000	10.00000
S6	119	5.04202	2.51570	600.00000	1.00000	10.00000
S7	119	5.73950	2.33057	683.00000	1.00000	10.00000
S8	119	4.91597	2.55310	585.00000	1.00000	10.00000
S9	119	4.63025	2.90489	551.00000	1.00000	10.00000
S10	119	4.36134	2.78230	519.00000	1.00000	10.00000
S11	119	5.65546	2.88901	673.00000	1.00000	10.00000

Correlation analysis results:

Pearson Correlation Coefficients, N = 119
Prob > |r| under H0: Rho=0

	Mark	rep	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Mark	1.00000	0.08333 0.3676	0.29298 0.0012	0.30141 0.0009	0.24308 0.0077	0.12058 0.1915	-0.16991 0.0647	0.16517 0.0726	0.21998 0.0162	0.17822 0.0525	-0.27748 0.0022	0.19048 0.0380	0.27379 0.0026
rep	0.08333 0.3676	1.00000	0.06615 0.4747	0.00989 0.9150	0.18555 0.0434	0.14285 0.1212	-0.14731 0.1099	0.07763 0.4014	0.10957 0.2356	0.04956 0.5925	-0.12248 0.1845	0.23629 0.0097	0.16377 0.0751
S1	0.29298 0.0012	0.06615 0.4747	1.00000	0.61917 <.0001	0.44363 <.0001	0.44894 <.0001	-0.19566 0.0330	0.34494 0.0001	0.42574 <.0001	0.30838 0.0006	-0.17139 0.0624	0.37306 <.0001	0.31794 0.0004
S2	0.30141 0.0009	0.00989 0.9150	0.61917 <.0001	1.00000	0.53959 <.0001	0.50974 <.0001	-0.28788 0.0015	0.25658 0.0049	0.44578 <.0001	0.34764 0.0001	-0.25527 0.0051	0.24166 0.0081	0.29468 0.0011
S3	0.24308 0.0077	0.18555 0.0434	0.44363 <.0001	0.53959 <.0001	1.00000	0.61418 <.0001	-0.29497 0.0011	0.16909 0.0660	0.24170 0.0081	0.29908 0.0010	-0.27395 0.0026	0.30222 0.0008	0.39084 <.0001
S4	0.12058 0.1915	0.14285 0.1212	0.44894 <.0001	0.50974 <.0001	0.61418 <.0001	1.00000	-0.26791 0.0032	0.25281 0.0055	0.33018 0.0002	0.25400 0.0053	-0.26243 0.0039	0.20581 0.0247	0.35111 <.0001
S5	-0.16991 0.0647	-0.14731 0.1099	-0.19566 0.0330	-0.28788 0.0015	-0.29497 0.0011	-0.26791 0.0032	1.00000	-0.08930 0.3342	-0.17857 0.0520	-0.22874 0.0123	0.41446 <.0001	-0.11091 0.2298	-0.35674 <.0001
S6	0.16517 0.0726	0.07763 0.4014	0.34494 0.0001	0.25658 0.0049	0.16909 0.0660	0.25281 0.0055	-0.08930 0.3342	1.00000	0.62053 <.0001	0.13646 0.1389	-0.13354 0.1477	0.30777 0.0007	0.16525 0.0725
S7	0.21998 0.0162	0.10957 0.2356	0.42574 <.0001	0.44578 <.0001	0.24170 0.0081	0.33018 0.0002	-0.17857 0.0520	0.62053 <.0001	1.00000	0.38226 <.0001	-0.21338 0.0198	0.26557 0.0035	0.29493 0.0011
S8	0.17822 0.0525	0.04956 0.5925	0.30838 0.0006	0.34764 0.0001	0.29908 0.0010	0.25400 0.0053	-0.22874 0.0123	0.13646 0.1389	0.38226 <.0001	1.00000	-0.01565 0.8658	0.18804 0.0406	0.29247 0.0012
S9	-0.27748 0.0022	-0.12248 0.1845	-0.17139 0.0624	-0.25527 0.0051	-0.27395 0.0026	-0.26243 0.0039	0.41446 <.0001	-0.13354 0.1477	-0.21338 0.0198	-0.01565 0.8658	1.00000	-0.05882 0.5251	-0.34653 0.0001
S10	0.19048 0.0380	0.23629 0.0097	0.37306 <.0001	0.24166 0.0081	0.30222 0.0008	0.20581 0.0247	-0.11091 0.2298	0.30777 0.0007	0.26557 0.0035	0.18804 0.0406	-0.05882 0.5251	1.00000	0.33507 0.0002
S11	0.27379 0.0026	0.16377 0.0751	0.31794 0.0004	0.29468 0.0011	0.39084 <.0001	0.35111 <.0001	-0.35674 <.0001	0.16525 0.0725	0.29493 0.0011	0.29247 0.0012	-0.34653 0.0001	0.33507 0.0002	1.00000

Regression analysis results - ANOVA:

STK 795
Megan Sajiwan, 11065177
Regression analysis

The REG Procedure
Model: MODEL1
Dependent Variable: Mark

Number of Observations Read	119
Number of Observations Used	119

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	12	4793.77405	399.48117	2.12	0.0212
Error	106	19961	188.31004		
Corrected Total	118	24755			

Root MSE	13.72261	R-Square	0.1937
Dependent Mean	62.16807	Adj R-Sq	0.1024
Coeff Var	22.07341		

Regression analysis results - Parameter estimates:

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	52.30437	7.66733	6.82	<.0001
rep	1	1.51861	4.81302	0.32	0.7530
S1	1	0.78102	0.73091	1.07	0.2877
S2	1	0.91669	0.77680	1.18	0.2406
S3	1	0.53125	0.76267	0.70	0.4876
S4	1	-1.08901	0.65818	-1.65	0.1010
S5	1	0.11550	0.48326	0.24	0.8116
S6	1	0.29102	0.66921	0.43	0.6645
S7	1	-0.00249	0.80095	-0.00	0.9975
S8	1	0.35163	0.58187	0.60	0.5469
S9	1	-0.99599	0.51304	-1.94	0.0549
S10	1	0.19195	0.53404	0.36	0.7200
S11	1	0.59598	0.53639	1.11	0.2690

Partial correlation analysis results:

STK 795
Megan Sajiwan, 11065177
Partial correlations

repeating

The CORR Procedure

11 Partial Variables:	S1 S2 S3 S4 S5 S6 S7 S8 S9 S10 S11
1 With Variables:	rep
1 Variables:	Mark

Pearson Partial Correlation Coefficients, N = 119 Prob > |r| under H0: Partial Rho=0	
	Mark
rep	0.03063 0.7530

STK 795
Megan Sajiwan, 11065177
Partial correlations

s1

The CORR Procedure

11 Partial Variables:	rep S2 S3 S4 S5 S6 S7 S8 S9 S10 S11
1 With Variables:	S1
1 Variables:	Mark

Pearson Partial Correlation Coefficients, N = 119
Prob > |r| under H0: Partial Rho=0

	Mark
S1	0.10323 0.2877

STK 795
Megan Sajiwan, 11065177
Partial correlations

s2

The CORR Procedure

11 Partial Variables:	rep S1 S3 S4 S5 S6 S7 S8 S9 S10 S11
1 With Variables:	S2
1 Variables:	Mark

Pearson Partial Correlation Coefficients, N = 119
Prob > |r| under H0: Partial Rho=0

	Mark
S2	0.11388 0.2406

STK 795
Megan Sajiwan, 11065177
Partial correlations

s3

The CORR Procedure

11 Partial Variables:	rep S1 S2 S4 S5 S6 S7 S8 S9 S10 S11
1 With Variables:	S3
1 Variables:	Mark

Pearson Partial Correlation Coefficients, N = 119
Prob > |r| under H0: Partial Rho=0

	Mark
S3	0.06750 0.4876

STK 795
Megan Sajiwan, 11065177
Partial correlations

s4

The CORR Procedure

11 Partial Variables:	rep S1 S2 S3 S5 S6 S7 S8 S9 S10 S11
1 With Variables:	S4
1 Variables:	Mark

Pearson Partial Correlation Coefficients, N = 119
Prob > |r| under H0: Partial Rho=0

	Mark
S4	-0.15867 0.1010

STK 795
Megan Sajiwan, 11065177
Partial correlations

s5

The CORR Procedure

11 Partial Variables:	rep S1 S2 S3 S4 S6 S7 S8 S9 S10 S11
1 With Variables:	S5
1 Variables:	Mark

Pearson Partial Correlation Coefficients, N = 119
Prob > |r| under H0: Partial Rho=0

	Mark
S5	0.02321 0.8116

STK 795
Megan Sajiwan, 11065177
Partial correlations

s6

The CORR Procedure

11 Partial Variables:	rep S1 S2 S3 S4 S5 S7 S8 S9 S10 S11
1 With Variables:	S6
1 Variables:	Mark

Pearson Partial Correlation Coefficients, N = 119
Prob > |r| under H0: Partial Rho=0

	Mark
S6	0.04220 0.6645

STK 795
Megan Sajiwan, 11065177
Partial correlations

s7

The CORR Procedure

11 Partial Variables:	rep S1 S2 S3 S4 S5 S6 S8 S9 S10 S11
1 With Variables:	S7
1 Variables:	Mark

Pearson Partial Correlation Coefficients, N = 119
Prob > |r| under H0: Partial Rho=0

	Mark
S7	-0.00030 0.9975

STK 795
Megan Sajiwan, 11065177
Partial correlations

s8

The CORR Procedure

11 Partial Variables:	rep S1 S2 S3 S4 S5 S6 S7 S9 S10 S11
1 With Variables:	S8
1 Variables:	Mark

Pearson Partial Correlation Coefficients, N = 119
Prob > |r| under H0: Partial Rho=0

	Mark
S8	0.05860 0.5469

STK 795
Megan Sajiwan, 11065177
Partial correlations

s9

The CORR Procedure

11 Partial Variables:	rep S1 S2 S3 S4 S5 S6 S7 S8 S10 S11
1 With Variables:	S9
1 Variables:	Mark

Pearson Partial Correlation Coefficients, N = 119 **Prob > |r| under H0: Partial Rho=0**	
	Mark
S9	-0.18529 0.0549

STK 795
Megan Sajiwan, 11065177
Partial correlations

s10

The CORR Procedure

11 Partial Variables:	rep S1 S2 S3 S4 S5 S6 S7 S8 S9 S11
1 With Variables:	S10
1 Variables:	Mark

Pearson Partial Correlation Coefficients, N = 119 **Prob > |r| under H0: Partial Rho=0**	
	Mark
S10	0.03489 0.7200

STK 795
Megan Sajiwan, 11065177
Partial correlations

s11

The CORR Procedure

11 Partial Variables:	rep S1 S2 S3 S4 S5 S6 S7 S8 S9 S10
1 With Variables:	S11
1 Variables:	Mark

Pearson Partial Correlation Coefficients, N = 119 **Prob > |r| under H0: Partial Rho=0**	
	Mark
S11	0.10730 0.2690

Stepwise regression results:

STK 795
Megan Sajiwan, 11065177
Stepwise Regression analysis

alpha=0.05

The REG Procedure
Model: MODEL1
Dependent Variable: Mark

Number of Observations Read	119
Number of Observations Used	119

Stepwise Selection: Step 1

Variable S2 Entered: R-Square = 0.0908 and C(p) = 4.5142

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	2248.92073	2248.92073	11.69	0.0009
Error	117	22506	192.35656		
Corrected Total	118	24755			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	52.40170	3.12646	54037	280.92	<.0001
S2	1.81593	0.53109	2248.92073	11.69	0.0009

Bounds on condition number: 1, 1

Stepwise Selection: Step 2

Variable S9 Entered: R-Square = 0.1339 and C(p) = 0.8593

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	3313.78708	1656.89354	8.96	0.0002
Error	116	21441	184.83493		
Corrected Total	118	24755			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	59.12840	4.15290	37469	202.72	<.0001
S2	1.48602	0.53844	1407.85518	7.62	0.0067
S9	-1.06957	0.44561	1064.86635	5.76	0.0180

Bounds on condition number: 1.0697, 4.2788

All variables left in the model are significant at the 0.1500 level.

No other variable met the 0.0500 significance level for entry into the model.

Summary of Stepwise Selection								
Step	Variable Entered	Variable Removed	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	S2		1	0.0908	0.0908	4.5142	11.69	0.0009
2	S9		2	0.0430	0.1339	0.8593	5.76	0.0180

The geometric Poisson distribution applied to traffic accidents

Zola Mary-Jean Sibanda 12104002

WST795 Research Report

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

Supervisor(s): Dr. R. Ehlers

Department of Statistics, University of Pretoria

2 November 2016

Abstract

In this paper we focus on the geometric Poisson distribution (also called the Pólya-Aeppli distribution) which is as a unique case of the compound Poisson distribution. Our main aim in this study is to show how an explicit expression for the probability function of the Pólya-Aeppli distribution can be derived, to derive some of the properties of distribution and to demonstrate the practical relevance of the distribution by fitting it to a traffic accident database as an example.

Declaration

I, *Zola Mary-Jean Sibanda,* 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.

_ _

Zola Mary-Jean Sibanda

_ _

Dr. R. Ehlers

_ _

Date

Contents

1 Introduction **5**

2 Preliminary results **6**

3 The geometric Poisson distribution **9**
3.1 Definition and probability mass function 9
3.2 The properties of the geometric Poisson distribution 10
3.3 Derivation of the GPD 13

4 Application **16**
4.1 Algorithm 16
4.2 Graphical displays 16
4.3 Fitting the distribution to traffic accident data 21

5 Conclusion **24**

Appendix **26**

List of Figures

1 Probability mass functions of the GPD when $E[Y] = 10$ and θ increases. 16
2 Graph of $FI[Y]$ against θ when $E[Y] = 10$. 17
3 Probability mass functions of the GPD when $E[Y] = 5$ and λ increases. 18
4 Graph of $FI[Y]$ against λ when $E[Y] = 5$. 19
5 Probability mass functions of the GPD when $\lambda = 2$ and $E[Y]$ is increased 19
6 Probability mass functions of the GPD when $\theta = 0.5$ and $E[Y]$ is increased. 20
7 Graph of $FI[Y]$ against $E[Y]$ for constant (i) $\lambda = 2$ and (ii) $\theta = 0.5$. 21
8 The p.m.f. of the total number of accident fatalities explained by the GPD with parameters $\hat{\lambda} = 9.833$ and $\hat{\theta} = 0.65753$. 23

List of Tables

1 Summary statistics of distribution displayed in Figure 1. 17
2 Summary statistics of distribution displayed in Figure 3. 18
3 Summary statistics of distribution displayed in Figure 5. 20
4 Summary statistics of distribution displayed in Figure 6. 20
5 Total accidents on a given Sunday, n_i and the corresponding total number of fatalities, y_i are recorded for each month during the period $1997 - 2004$ in the Groningen region. 22
6 The geometric distribution fit of the observed frequency for the number of fatalities. 23

1 Introduction

With the increase of technological advancements, the world is in an era where there is a growth in the reliance and usage of motor vehicles. As a result, the increase in the number of vehicles has led to a rise in traffic accidents and fatalities. Our main objective is to study a special case of the compound Poisson distribution (CPD) known as the geometric Poisson distribution. We would then like to study the geometric Poisson distribution (also known as the Pólya-Aeppli distribution) and show that this distribution can be applied to traffic accident data which was previously seen in the article presented by Özel and Inal [14]. This distribution is defined by Özel and Inal [14] as follows:

Definition Let N be a discrete random variable that is Poisson distributed with parameter $\lambda > 0$ (i.e. $N \sim POI(\lambda)$) and let X_i, $i = 1, 2, 3 \ldots$, be independent identically distributed (i.i.d.) random variables from a geometric distribution with parameter θ (i.e. $X_i \sim GEO(\theta)$) and independent of N. Then Y is defined as

$$Y = \sum_{i=1}^{N} X_i.$$

that has a geometric Poisson distribution (GPD) denoted as $Y \sim GEOPOI(\lambda, \theta)$.

This distribution was first introduced in 1930 by Pólya [15] with a reference to a thesis by Aeppli [2] in 1924. The GPD has been studied and applied further in several real-world situations and has been seen to be of statistical significance as it has practical relevance. For example, in 1992 Johnson et al. [8] developed a linear formula so that the calculations of probabilities of the CPD could be simply illustrated in the case of the GPD. Then in 1995, Randolph and Sahinoglu [16] applied the distribution to the controlling of software defects. Then the geometric Poisson CUSUM control scheme was developed for process control by Chen et al. [6] in 2005. The following year the geometric Poisson distribution was used for the biological process of modeling DNA substitutions by Rosychuk et al. [18] whilst the distribution was also modeled for the overlapping of word occurrences in 2007 by Robin et al. [17].

In other studies, Nuel [13] used Kummer's confluent geometric function derived the geometric Poisson distribution recurrence relation in 2008. Subsequently in 2010, Özel and Inal [14] derived the explicit probability function of the distribution and set up an algorithm to compute these probabilities. Thereafter Ata and Ozel [4] derived the compound Poisson distribution for the survival functions in 2012.

The article presented by Anwar and Ahmed [3] prove some of the properties of the geometric Poisson distribution looking particularly at its infinite divisibility, log-concavity and unimodality. The survival function is also obtained, the first-order negative moment developed and lastly, the computation and proof of the above mentioned properties is characterized by the use of the recursive formula. Minkova and Balakrishnan [11] focused on the compound weighted Poisson distribution. In their article the variability of the different models measured by the Fisher index of dispersion is discussed and the factorial moment of mean measure is introduced.

The calculation of the exact probabilities remains tedious as there is some difficulty in terms of computational memory and time, however, Özel and Inal [14] give a forthright derivation and proof of the explicit probability function of the GPD. They also derive an algorithm to illustrate the usefulness of the distribution by applying it to the traffic accident data presented by Meintanis [10]. Furthermore, in the article presented by Leiter and Hamdan [9], two bivariate models similar to this distribution are studied where the number of accidents and the number of fatalities or fatal accidents were investigated.

Our primary aim in this study will entail deriving an explicit probability function of the geometric Poisson distribution and to derive the expected value and variance of the distribution. We will apply the distribution in a similar way as was done by Özel and Inal [14]. In Section 2, the necessary and preliminary results that will be used throughout the study will be given. In Section 3, we will present the derivation of the probability function of the GPD with the use of integer partitions and compute the expected value and variance of the distribution by using the probability generating function. The algorithm proposed by Özel and Inal [14] will be discussed and then both the mass function and cumulative distribution function of the distribution for

different values of the parameters will be illustrated graphically in Section 4. The distribution will also be applied to a numerical example from traffic accident data. Lastly, in Section 5 the conclusion will be given.

2 Preliminary results

The following statistical results given will be used throughout the study.

The following definitions given below are as given in Bain and Engelhardt [5].

Definition 1 The random variable X with a geometric distribution with parameter θ (i.e. $X \sim GEO(\theta)$), has a probability mass function (p.m.f.) given by

$$\begin{aligned} p_j &= P(X = j) \\ &= \theta(1-\theta)^{j-1}, \ j = 1, 2, 3, \ldots. \end{aligned} \tag{1}$$

Definition 2 The probability mass function of N, a discrete random variable that is Poisson distributed with parameter $\lambda > 0$ (i.e. $N \sim POI(\lambda)$) is given by

$$P(N = n) = e^{-\lambda}\frac{\lambda^n}{n!}. \tag{2}$$

Definition 3 Consider the discrete random variable X taking nonnegative integer values $\{0, 1, 2, \ldots\}$ with probability $P(X = j) = p_j$. The probability generating function (p.g.f.) of X is defined as

$$\begin{aligned} g_X(z) &= E\left[z^X\right] \\ &= \sum_{j=0}^{\infty} p_j z^j. \end{aligned} \tag{3}$$

where $0 \leq z \leq 1$.

The probability generating function is a power series that can be duly expanded as well as differentiated to unveil the individual probabilities.

Theorem 1 Let X be a discrete random variable. Differentiating the p.g.f. $g_X(z)$ will give the probabilities

$$\begin{aligned} p_j &= P(X = j) \\ &= \frac{1}{j!}\frac{\partial^j}{\partial z^j}\left(g_X(z)\right)|_{z=0}. \end{aligned} \tag{4}$$

where $j = 0, 1, 2, \ldots$

Proof. We have that

$$p_0 = P(X = 0) = g_X(0) \cdot$$

The first derivative is

$$\begin{aligned}\frac{\partial}{\partial z}\left(g_X\left(z\right)\right) &= g_X^{(1)}\left(z\right)\\ &= p_1 + 2p_2 z + 3p_3 z^2 + 4p_4 z^3 + \ldots\end{aligned}$$

Substituting $z = 0$ gives $g_X^{(1)}\left(0\right) = p_1 = P\left(X = 1\right)$.

From the second derivative:

$$\begin{aligned}\frac{\partial^2}{\partial z^2}\left(g_X\left(z\right)\right) &= g_X^{(2)}\left(z\right)\\ &= \left(2\right)\left(1\right)p_2 + \left(3\right)\left(2\right)p_3 z + \left(4\right)\left(3\right)p_4 z^2 + \ldots\\ &= 2p_2 + 6p_3 z + 12p_4 z^2 + \ldots\end{aligned}$$

It follows that $g_X^{(2)}\left(0\right) = 2p_2$ and that $\frac{1}{2}g_X^{(2)}\left(0\right) = p_2 = P\left(X = 2\right) = \frac{1}{2!}g_X^{(2)}\left(0\right)$.

From the third derivative:

$$\begin{aligned}\frac{\partial^3}{\partial z^3}\left(g_X\left(z\right)\right) &= g_X^{(3)}\left(z\right)\\ &= \left(3\right)\left(2\right)\left(1\right)p_3 + \left(4\right)\left(3\right)\left(2\right)p_4 z + \ldots\\ &= 6p_3 + 24p_4 z + \ldots\end{aligned}$$

It follows that $g_X^{(3)}\left(0\right) = 6p_3$ and that $\frac{1}{6}g_X^{(3)}\left(0\right) = p_3 = P\left(X = 3\right) = \frac{1}{3!}g_X^{(3)}\left(0\right)$.

$$\vdots$$

Continuing in a similar manner, we get the result in (4). ■

Theorem 2 Suppose $X \sim GEO(\theta)$ distributed with p.m.f. $P\left(X = j\right) = p_j = \theta\left(1-\theta\right)^{j-1},\ j = 1, 2, 3, \ldots$ Then the p.g.f. is given by

$$\begin{aligned}E\left[z^X\right] &= g_X\left(z\right)\\ &= \sum_{j=1}^{\infty} p_j z^j\\ &= \frac{\theta z}{1-\left(1-\theta\right)z}.\end{aligned} \tag{5}$$

Proof. From (1) and (3) we have that

$$\begin{aligned}g_X\left(z\right) &= p_1 z + p_2 z^2 + p_3 z^3 + \ldots\\ &= \theta\left(1-\theta\right)^0 z + \theta\left(1-\theta\right)z^2 + \theta\left(1-\theta\right)^2 z^3 + \ldots \qquad (A)\end{aligned}$$

letting

$$g_X\left(z\right)\left(1-\theta\right)z = \theta\left(1-\theta\right)z^2 + \theta\left(1-\theta\right)^2 z^3 + \theta\left(1-\theta\right)^3 z^4 + \ldots (B)$$

and calculating $(A) - (B)$ gives

$$\begin{aligned} g_X(z) - g_Y(z)(1-\theta)z &= \theta z. \\ g_X(z)[1-(1-\theta)z] &= \theta z. \\ g_X(z) &= \frac{\theta z}{1-(1-\theta)z}. \end{aligned}$$

■

The following results on compound distributions are as given in Sundt and Vernic [19].

Definition 4 Let N be a discrete random variable and X_i, $i = 1, 2, 3 \ldots$, i.i.d. random variables independent of N. Then

$$Y = \sum_{i=1}^{N} X_i. \tag{6}$$

is referred to as a random variable having a compound distribution.

Theorem 3 The p.m.f. of a random variable Y having a compound distribution is given by

$$\begin{aligned} p_Y(k) &= P(Y=k) \\ &= \sum_{n=0}^{\infty} P(X_1 + X_2 + \ldots + X_n = k | N = n) P(N = n). \end{aligned} \tag{7}$$

Proof. The p.m.f. of $Y = \sum_{i=1}^{N} X_i$ is as follows

$$\begin{aligned} p_Y(k) &= P(Y=k) \\ &= P\left(\sum_{i=1}^{N} X_i = k\right) \\ &= P\left(\sum_{i=1}^{N} X_i = k \; and \; N = 0\right) + P\left(\sum_{i=1}^{N} X_i = k \; and \; N = 1\right) + P\left(\sum_{i=1}^{N} X_i = k \; and \; N = 2\right) + \ldots \\ &= \sum_{n=0}^{\infty} P\left(\sum_{i=1}^{N} X_i = k \; and \; N = n\right). \end{aligned}$$

It follows from the definition of a conditional probability $P(A|B) = \frac{P(A \bigcap B)}{P(B)}$ *i.e.* $P(A \bigcap B) = P(A|B) P(B)$ that

$$p_Y(k) = \sum_{n=0}^{\infty} P\left(\sum_{i=1}^{N} X_i = k | N = n\right) P(N = n).$$

■

We should note that obtaining an explicit formula for the probability function of $Y = \sum_{i=1}^{N} X_i$ from (7) is hardly a simple matter and as a result serves as a hindrance in the complete usage of the compound Poisson distribution (CPD) in [13] and [8] as k is increased.

Theorem 4 The expected value of a random variable having a compound distribution that is $Y = \sum_{i=1}^{N} X_i$ where $i = 1, 2, 3, \ldots$, is given by

$$E[Y] = E_N[N] E[X]. \tag{8}$$

Proof. It follows from the definition that

$$\begin{aligned} E_Y[Y] &= E_N\left[E_{Y|N}[Y]\right] \\ &= E_N[NE(X)] \\ &= E_N[N] E[X]. \end{aligned}$$

■

Theorem 5 The variance of a random variable having a compound distribution that is $Y = \sum_{i=1}^{N} X_i$ where $i = 1, 2, 3, \ldots$, is given by

$$Var[Y] = E_N[N] Var[X] + (E[X])^2 Var_N[N]. \tag{9}$$

Proof. It follows from the definition that

$$\begin{aligned} Var[Y] &= E_N\left[Var_{Y|N}[Y]\right] + Var_N\left[E_{Y|N}[Y]\right] \\ &= E_N[N] Var[X] + [E(X)]^2 Var_N[N]. \end{aligned}$$

■

3 The geometric Poisson distribution

The geometric Poisson distribution is defined in this section and some properties are derived.

3.1 Definition and probability mass function

The GPD is defined by Özel and Inal [14] as follows

Definition 5 Let N be a discrete random variable that is Poisson distributed with parameter $\lambda > 0$ (i.e. $N \sim POI(\lambda)$) and let X_i, $i = 1, 2, 3 \ldots$, be i.i.d. random variables from a geometric distribution with parameter θ (i.e. $X_i \sim GEO(\theta)$) and independent of N. Then Y, defined as

$$Y = \sum_{i=1}^{N} X_i. \tag{10}$$

has a geometric Poisson distribution denoted as $Y \sim GEOPOI(\lambda, \theta)$.

The following Theorem given below is as defined by Johnson *et al.* [8] of the geometric Poisson distribution.

Theorem 6 Suppose $Y \sim GEOPOI(\lambda, \theta)$. Then the probability mass function of Y is given by

$$p_Y(k) = \sum_{n=1}^{k} e^{-\lambda} \frac{\lambda^n}{n!} \left(\begin{array}{c} k-1 \\ n-1 \end{array} \right) \theta^n (1-\theta)^{k-n}, \; k = 1, 2, 3, \ldots \tag{11}$$

where $\lambda > 0$ and $0 < \theta < 1$.

Due to the difficulties that arise in deriving probabilities obtained from (11) for greater values of k, Nuel [13] devised a new way of rewriting equation (11) in a much simpler way. This made use of Kummer's confluent hypergeometric function by expressing it as a recurrence relation. The result is given in Theorem 7 below.

Theorem 7 Suppose $Y \sim GEOPOI(\lambda, \theta)$ as defined in (10). Then it follows that

$$\begin{aligned} P(Y = 0) &= e^{-\lambda}. \\ P(Y = 1) &= e^{-\lambda}(1-\theta)s. \end{aligned} \tag{12}$$

and $\forall\, k \epsilon \mathbb{N},\; k \geq 2$

$$P(Y = k) = \frac{(2k-2+s)}{k}(1-\theta) P(Y = k-1) + \frac{(2-k)}{k}(1-\theta)^2 P(Y = k-2). \tag{13}$$

where $s = \frac{\lambda\theta}{1-\theta}$.

It should be noted that (13) is a recursive formula and that its computation requires the results of previous probabilities for $k = 0, 1, 2, \ldots, k-1$ in order to solve $P(Y = k)$.

3.2 The properties of the geometric Poisson distribution

Theorem 8 If $Y \sim GEOPOI(\lambda, \theta)$ with p.m.f. given by (11) then the p.g.f. for Y is given as follows

$$\begin{aligned} g_Y(z) &= \sum_{n=0}^{\infty} e^{-\lambda} \frac{\lambda^n}{n!} [g_X(z)]^n \\ &= e^{\lambda[g_X(z)-1]}. \end{aligned} \tag{14}$$

Proof. The proof is given by

$$\begin{aligned} g_Y(z) &= \sum_{n=0}^{\infty} e^{-\lambda} \frac{\lambda^n}{n!} [g_X(z)]^n \\ &= e^{-\lambda} \sum_{n=0}^{\infty} \frac{[\lambda g_X(z)]^n}{n!} \\ &= e^{-\lambda} e^{\lambda g_X(z)} \\ &= e^{\lambda[g_X(z)-1]}. \end{aligned}$$

By substituting (5) into (14), then the probability generating function of $g_Y(z)$ is as follows

$$\begin{aligned} g_Y(z) &= e^{-\lambda} e^{\lambda\left[p_1 z + p_2 z^2 + p_3 z^3 + \ldots\right]} \\ &= e^{-\lambda} e^{\lambda\left[\frac{\theta z}{1-(1-\theta)z}\right]} \\ &= e^{\lambda\left[\frac{\theta z}{1-(1-\theta)z} - 1\right]} \\ &= e^{\lambda\left[\frac{z-1}{1-(1-\theta)z}\right]}. \end{aligned}$$

■

Recall from (8) and (9) of a compound distribution. The expected value and variance of the GPD are calculated in the theorems below.

Theorem 9 The expected value of a random variable having a compound distribution that is $Y = \sum_{i=1}^{N} X_i$ where the $X_i \sim GEO(\theta)$, $i = 1, 2, 3, \ldots$, independent of $N \sim POI(\lambda)$ is given by

$$E[Y] = \frac{\lambda}{\theta}. \tag{15}$$

Proof. Let $X_i \sim GEO(\theta)$ then it follows that $E[X_i] = \frac{1}{\theta}$ and $Var[X_i] = E\left[X_i^2\right] - [E(X_i)]^2 = \frac{1-\theta}{\theta^2}$. Also given that $N \sim POI(\lambda)$, we have that $E_N[N] = Var_N[N] = \lambda$.
Then from (8) it follows that the expected value of the GPD is given by

$$\begin{aligned} E_Y[Y] &= E_N[N] E[X] \\ &= \lambda E[X] \\ &= \frac{\lambda}{\theta.} \end{aligned}$$

■

Theorem 10 The variance of a random variable having a compound distribution that is $Y = \sum_{i=1}^{N} X_i$ where the $X_i \sim GEO(\theta)$, $i = 1, 2, 3, \ldots$, independent of $N \sim POI(\lambda)$ is given by

$$Var[Y] = \frac{\lambda(2-\theta)}{\theta^2}. \tag{16}$$

Proof. It follows from (9), the variance of a compound Poisson distribution is calculated as follows:
Since (8) holds and $E_N[N] = Var_N[N] = \lambda$ because $N \sim POI(\lambda)$ then it follows that

$$\begin{aligned} Var[Y] &= E_N[N] Var[X] + [E(X)]^2 Var_N[N] \\ &= E[N]\left(Var[X] + [E(X)]^2\right) \\ &= E[N]\left(E\left[X^2\right]\right) \\ &= \lambda E\left[X^2\right]. \end{aligned}$$

It then follows from above and the proof in (15) that the variance of the GPD is given by

$$\begin{aligned} Var[Y] &= \lambda E\left[X^2\right] \\ &= \frac{\lambda(2-\theta)}{\theta^2}. \end{aligned}$$

■

Next we would like perform a variability measure that is commonly used of the random variable $Y \sim GEOPOI(\lambda, \theta)$. This is known as the Fisher index of dispersion. It is defined by Fisher [7] as follows

Definition 6 The Fisher index of dispersion for the random variable Y is defined as

$$Fisher\, index = FI(Y) = \frac{Var[Y]}{E[Y]}. \tag{17}$$

It is stated by Minkova and Balakrishnan [11] that a distribution is said to be equi-dispersed if $FI(Y) = 1$, under-dispersed if $FI(Y) < 1$ and over-dispersed if $FI(Y) > 1$.

Theorem 11 The Fisher index of dispersion of the random variable $Y \sim GEOPOI(\lambda, \theta)$ is

$$\begin{aligned} FI(Y) &= \frac{2}{\theta} - 1 \\ &= 1 + \frac{2\rho}{1-\rho}. \end{aligned} \tag{18}$$

where $\rho = 1 - \theta$.

Proof. Recall that $Y \sim GEOPOI(\lambda, \theta)$, then from (15) and (16) we have $E[Y] = \frac{\lambda}{\theta}$ and $Var[Y] = \frac{\lambda(2-\theta)}{\theta^2}$. It then follows that the Fisher index of Y is given by

$$\begin{aligned} FI(Y) &= \frac{Var[Y]}{E[Y]} \\ &= \frac{\left[\frac{\lambda(2-\theta)}{\theta^2}\right]}{\left[\frac{\lambda}{\theta}\right]} \\ &= \frac{(2-\theta)}{\theta} = \frac{2}{\theta} - 1. \end{aligned}$$

Letting $\theta = 1 - \rho$, it follows that

$$\begin{aligned} FI(Y) &= \frac{2-(1-\rho)}{1-\rho} \\ &= \frac{1+\rho}{1-\rho} \\ &= 1 + \frac{2\rho}{1-\rho} > 1. \end{aligned}$$

■

From (18) it is clear that the random variable $Y \sim GEOPOI(\lambda, \theta)$ is over-dispersed with respect to the Poisson distribution which has a Fisher index of 1. It should be noted that the particular feature makes the Pólya-Aeppli distribution better suited for insurance and financial data as mentioned by Minkova and Balakrishnan [12].

3.3 Derivation of the GPD

Theorem 12 given below is the explicit formula for the probability generating function of the geometric Poisson distribution derived by Özel and Inal [14] for the case where $j = 1, 2$ i.e. $m = 2$.

Theorem 12 Let $Y = \sum_{i=1}^{N} X_i$ where $N \sim POI(\lambda)$ independent of $X_i \sim GEO(\theta)$, $i = 1, 2, 3 \ldots$, i.i.d. with $p_j = P(X_i = j) = \theta(1-\theta)^{j-1}$ and let $\lambda_j = \lambda p_j$, $j = 1, 2, \ldots, m$. Then for $m = 2$, the explicit formula for the probability function of Y is given by

$$\begin{aligned} p_Y(k) &= P(Y = k) \\ &= \begin{cases} e^{-\lambda}, & k = 0 \\ e^{-\lambda} \sum_{i=0}^{[k/2]} \frac{\lambda_1^{k-2i} \lambda_2^i}{(k-2i)!\,i!}, & k = 1, 2, 3, \ldots \end{cases} \end{aligned} \tag{19}$$

where the integer part of the number in the brackets is denoted by $[\,]$.

Proof. Refer to [14] for the proof. ■

The next theorem derives the result in Theorem 12 for any m and then the case where $m = 3$.

Theorem 13 Let $Y = \sum_{i=1}^{N} X_i$ where $N \sim POI(\lambda)$ independent of $X_i \sim GEO(\theta)$, $i = 1, 2, 3 \ldots$, i.i.d. with $p_j = P(X_i = j) = \theta(1-\theta)^{j-1}$ and let $\lambda_j = \lambda p_j$, $j = 1, 2, \ldots, m$. Then a general form of the probability functions of Y given any m are

$$\begin{aligned} P(Y=0) &= e^{-\lambda}, \\ P(Y=1) &= e^{-\lambda} \frac{\lambda_1}{1!}, \\ P(Y=2) &= e^{-\lambda} \left[\frac{\lambda_1^2}{2!} + \frac{\lambda_2}{1!} \right], \\ P(Y=3) &= e^{-\lambda} \left[\frac{\lambda_1^3}{3!} + \frac{\lambda_1 \lambda_2}{1!1!} + \frac{\lambda_3}{1!} \right], \\ P(Y=4) &= e^{-\lambda} \left[\frac{\lambda_1^4}{4!} + \frac{\lambda_1^2 \lambda_2}{2!1!} + \frac{\lambda_2^2}{2!} + \frac{\lambda_1 \lambda_3}{1!1!} + \frac{\lambda_4}{1!} \right], \\ P(Y=5) &= e^{-\lambda} \left[\frac{\lambda_1^5}{5!} + \frac{\lambda_1^3 \lambda_2}{3!1!} + \frac{\lambda_1 \lambda_2^2}{1!2!} + \frac{\lambda_1^2 \lambda_3}{2!1!} + \frac{\lambda_1 \lambda_4}{1!1!} + \frac{\lambda_2 \lambda_3}{1!1!} + \frac{\lambda_5}{1!} \right], \\ &\vdots \end{aligned} \tag{20}$$

Proof. We start by deriving the p.g.f. of Y for $m = 3$ by using (5) and (14) and the fact that $\lambda_j = \lambda p_j$ where $j = 1, 2, 3$. This gives

$$\begin{aligned} g_Y(z) &= e^{-\lambda} e^{\lambda \left[p_1 z + p_2 z^2 + p_3 z^3 \right]} \\ &= e^{-\lambda} e^{\lambda p_1 z + \lambda p_2 z^2 + \lambda p_3 z^3} \\ &= e^{-\lambda} e^{\lambda_1 z + \lambda_2 z^2 + \lambda_3 z^3} \end{aligned} \tag{21}$$

By using (4) and (21), the individual probabilities of the probability mass function of Y can be found by calculating the kth derivative of $g_Y(z)$, given that $z = 0$ and are

$$\begin{aligned} P(Y=0) &= g_Y(0) \text{ and} \\ P(Y=k) &= \frac{\partial^k / \partial z^k \left(g_Y(z)\right)|_{z=0}}{k!}, \qquad k=1,2,3,\ldots \end{aligned} \tag{22}$$

Given below are the corresponding probabilities $P(Y=k)$, $k=0,1,2,3,\ldots$, of the probability generating function for $m=3$. These are obtained by differentiating the p.g.f. in (21) as in (22) as follows

$$\begin{aligned} P(Y=0) &= e^{-\lambda}, \\ P(Y=1) &= e^{-\lambda}\frac{\lambda_1}{1!}, \\ P(Y=2) &= e^{-\lambda}\left[\frac{\lambda_1^2}{2!}+\frac{\lambda_2}{1!}\right], \\ P(Y=3) &= e^{-\lambda}\left[\frac{\lambda_1^3}{3!}+\frac{\lambda_1\lambda_2}{1!1!}+\frac{\lambda_3}{1!}\right], \\ P(Y=4) &= e^{-\lambda}\left[\frac{\lambda_1^4}{4!}+\frac{\lambda_1^2\lambda_2}{2!1!}+\frac{\lambda_2^2}{2!}+\frac{\lambda_1\lambda_3}{1!1!}\right], \\ P(Y=5) &= e^{-\lambda}\left[\frac{\lambda_1^5}{5!}+\frac{\lambda_1^3\lambda_2}{3!1!}+\frac{\lambda_1\lambda_2^2}{1!2!}+\frac{\lambda_1^2\lambda_3}{2!1!}+\frac{\lambda_2\lambda_3}{1!1!}\right], \\ &\vdots \end{aligned} \tag{23}$$

It can be seen from (23) that the nonnegative integers can be expressed in several ways as a sum of these integers.

So as a result, a general form of the probability functions of Y given that $m>3$ can be derived in a similar manner as we have done above. The result is given as follows

$$\begin{aligned} P(Y=0) &= e^{-\lambda}, \\ P(Y=1) &= e^{-\lambda}\frac{\lambda_1}{1!}, \\ P(Y=2) &= e^{-\lambda}\left[\frac{\lambda_1^2}{2!}+\frac{\lambda_2}{1!}\right], \\ P(Y=3) &= e^{-\lambda}\left[\frac{\lambda_1^3}{3!}+\frac{\lambda_1\lambda_2}{1!1!}+\frac{\lambda_3}{1!}\right], \\ P(Y=4) &= e^{-\lambda}\left[\frac{\lambda_1^4}{4!}+\frac{\lambda_1^2\lambda_2}{2!1!}+\frac{\lambda_2^2}{2!}+\frac{\lambda_1\lambda_3}{1!1!}+\frac{\lambda_4}{1!}\right], \\ P(Y=5) &= e^{-\lambda}\left[\frac{\lambda_1^5}{5!}+\frac{\lambda_1^3\lambda_2}{3!1!}+\frac{\lambda_1\lambda_2^2}{1!2!}+\frac{\lambda_1^2\lambda_3}{2!1!}+\frac{\lambda_1\lambda_4}{1!1!}+\frac{\lambda_2\lambda_3}{1!1!}+\frac{\lambda_5}{1!}\right], \\ &\vdots \end{aligned}$$

This is the result given in (20).

■

Note that the individual probabilities $P(Y=k)$, $k=1,2,3,\ldots$, can be determined depending on how k is partitioned for any given integer value m. If we consider $k=5$ and $m\to\infty$ as an example then there are 7

types of partitions which are$\{1,1,1,1,1\}$, $\{2,1,1,1\}$, $\{3,1,1\}$, $\{4,1\}$,$\{2,2,1\}$, $\{3,2\}$,$\{5\}$. Similarly if $k=6$ and $m\to\infty$ there are 11 types of partitions which are $\{1,1,1,1,1,1\}$, $\{2,1,1,1,1\}$, $\{3,1,1,1\}$, $\{4,1,1\}$, $\{5,1\}$, $\{2,2,1,1\}$, $\{2,2,2\}$, $\{3,2,1\}$, $\{3,3\}$, $\{4,2\}$, $\{6\}$. We further note that as performed in (23) where $m=3$, the partitions are also obtained in a similar fashion although tend to have fewer partitions for every $k>m$. For example, consider $k=5$ again and $m=3$ as before then we now have 5 types of partitions which are $\{1,1,1,1,1\}$, $\{2,1,1,1\}$, $\{3,1,1\}$, $\{2,2,1\}$, $\{3,2\}$. Since (20) is not recursive, the probabilities $P(Y=k)$, $k=1,2,3,\ldots$, can be computed directly.

Given the probabilities derived in (20) and (23), we further go on to prove that they satisfy the following conditions.

Theorem 14. The probability mass function given in (20) satisfies the following conditions

$$p_Y(k)=P(Y=k)\geq 0\;\forall\, k\to\infty \quad \textit{and} \quad \sum_{k=0}^{\infty}p_Y(k)=1. \tag{24}$$

Proof. Since for any event A, $0\leq P(A)\leq 1$ we have that

$$0\leq p_Y(k)\leq 1,\ k=0,1,2,\ldots$$

Then it follows that rewriting (23),

$$\begin{aligned}
\sum_{k=0}^{\infty}p_Y(k) &= P(Y=0)+P(Y=1)+P(Y=3)+P(Y=4)+\ldots \\
&= e^{-\lambda}\left\{1+\left[\frac{\lambda_1}{1!}\right]+\left[\frac{\lambda_1^2}{2!}+\frac{\lambda_2}{1!}\right]+\left[\frac{\lambda_1^3}{3!}+\frac{\lambda_1\lambda_2}{1!1!}+\frac{\lambda_3}{1!}\right]+\left[\frac{\lambda_1^4}{4!}+\frac{\lambda_1^2\lambda_2}{2!1!}+\frac{\lambda_2^2}{2!}+\frac{\lambda_1\lambda_3}{1!1!}\right]+\cdots\right\} \\
&= e^{-\lambda}\left\{1+\frac{\lambda_1}{1!}+\frac{\lambda_1^2}{2!}+\frac{\lambda_2}{1!}+\frac{\lambda_1^3}{3!}+\frac{\lambda_1\lambda_2}{1!1!}+\frac{\lambda_3}{1!}+\frac{\lambda_1^4}{4!}+\frac{\lambda_1^2\lambda_2}{2!1!}+\frac{\lambda_2^2}{2!}+\frac{\lambda_1\lambda_3}{1!1!}+\cdots\right\} \\
&= e^{-\lambda}\left\{\left[1+\tfrac{\lambda_1}{1!}+\tfrac{\lambda_1^2}{2!}+\cdots\right]+\tfrac{\lambda_2}{1!}\left[1+\tfrac{\lambda_1}{1!}+\tfrac{\lambda_1^2}{2!}+\cdots\right]+\tfrac{\lambda_3}{1!}\left[1+\tfrac{\lambda_1}{1!}+\tfrac{\lambda_1^2}{2!}+\cdots\right]+\quad\cdots\right. \\
&\quad \left.+\frac{\lambda_2^2}{2!}\left[1+\frac{\lambda_1}{1!}+\cdots\right]+\cdots\right\}
\end{aligned}$$

Since we know that $e^z=\sum_{k=0}^{\infty}\frac{z^k}{k!}$ and given that $\lambda_j=\lambda p_j$, $j=1,2,3$, it continues from above that

$$\begin{aligned}
\sum_{k=0}^{\infty}p_Y(k) &= e^{-\lambda}\left\{e^{\lambda p_1}+\frac{\lambda_2}{1!}e^{\lambda p_1}+\frac{\lambda_3}{1!}e^{\lambda p_1}+\frac{\lambda_2^2}{2!}e^{\lambda p_1}+\cdots\right\} \\
&= e^{-\lambda}e^{\lambda p_1}\left\{1+\frac{\lambda_2}{1!}+\frac{\lambda_2^2}{2!}+\cdots+\frac{\lambda_3}{1!}+\cdots\right\} \\
&= e^{-\lambda}e^{\lambda p_1}e^{\lambda p_2}e^{\lambda p_3} \\
&= e^{-\lambda}e^{\lambda(p_1+p_2+p_3)} \\
&= e^{-\lambda}e^{\lambda} \\
&= 1.
\end{aligned}$$

■

From (23) where $j=1,2,3$, we have proved that $\sum_{k=0}^{\infty}p_Y(k)=1$. It then follows that the conditions must also satisfy the generalized form (20) since the p.g.f. derived in (23) is a distinct case of (20) where $m>3$.

4 Application

4.1 Algorithm

An algorithm is given by Özel and Inal [14] using (19) to draw the probability mass functions of the GPD. We have used the exact probabilities formula for the GPD given by (11) to draw the graphs illustrated in the next subsection. We are of the opinion that the complexity of the algorithm proposed by Özel and Inal [14] does not justify its use in drawing the graphs, because of the short computational time in SAS Software [1] and the simplicity of the the calculations (i.e. not recursive).

4.2 Graphical displays

Our investigation continues as we compare by illustration the probability mass functions (pmfs) of the GPD produced from SAS Software [1] graphing capabilities. This is done in two parts:

(i) by keeping the expected value constant and varying the parameters λ and θ;

(ii) secondly by keeping one of the parameters constant and varying the expected value which in turns varies the other parameter.

Using the expression of the p.m.f. of the geometric Poisson distribution given in (11), the computation time when using SAS Software [1] is about 0.09s in real time and 0.06s in (Central Processing Unit) CPU time to run the programs given in the Appendix, which draws the graphs illustrated below.

Suppose $Y \sim GEOPOI(\lambda, \theta)$.

Part 1: Keeping $E[Y]$ constant and varying the parameters λ and θ.

Figure 1 shows the pmfs of the GPD when $E[Y] = \frac{\lambda}{\theta} = 10$ and θ increases from 0.25, 0.5 to 0.75. The summary statistics are given in Table 1.

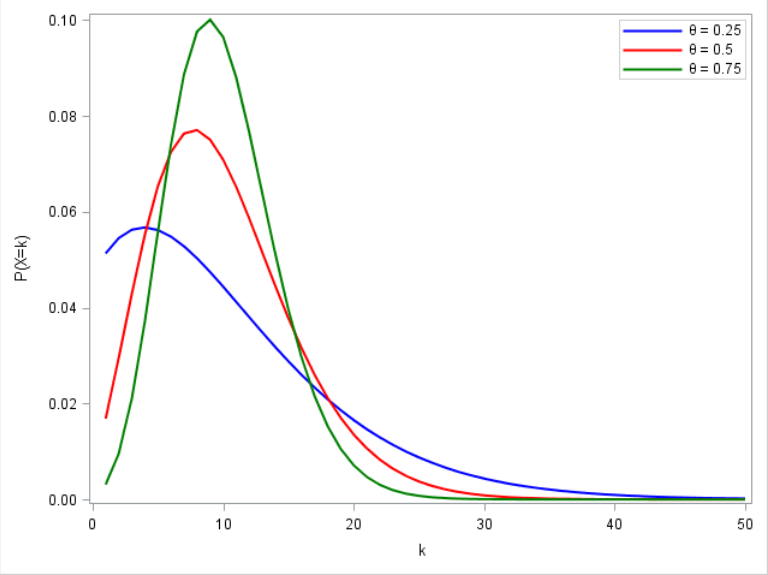


Figure 1: Probability mass functions of the GPD when $E[Y] = 10$ and θ increases.

θ	λ	$E[Y]$	$Var[Y]$	$FI[Y]$
0.25	2.5	10	70	7
0.5	5.0	10	30	3
0.75	7.5	10	16.67	1.667

Table 1: Summary statistics of distribution displayed in Figure 1.

From Figure 1 and Table 1, given a constant central location, increasing θ results in a smaller variance and smaller dispersion index, $FI[Y]$. The latter is also illustrated in Figure 2 which displays the relationship between the Fisher index of dispersion and θ when $E[Y] = 10$. The GPD is over-dispersed relative to the Poisson distribution which has a Fisher index of 1, but as θ increases whilst $E[Y]$ remains constant the Fisher index of the GPD tends to 1.

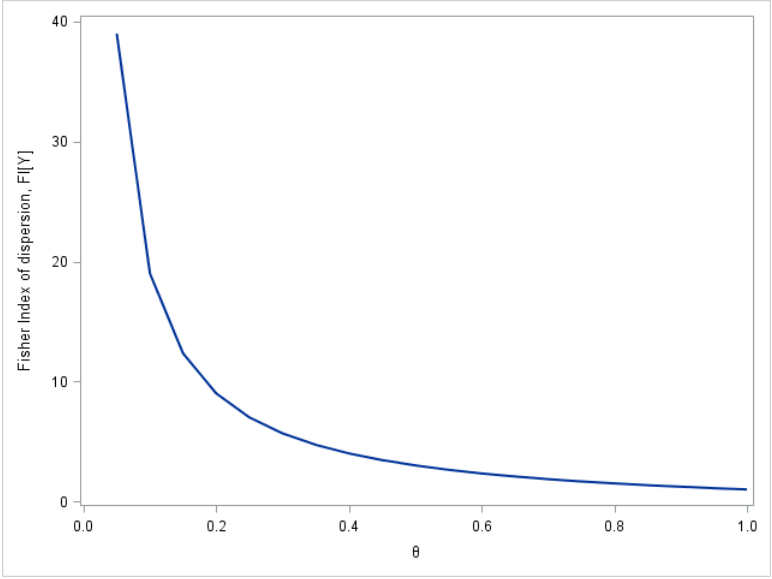


Figure 2: Graph of $FI[Y]$ against θ when $E[Y] = 10$.

Similarly, Figure 3 shows the pmfs of the GPD when $E[Y] = \frac{\lambda}{\theta} = 5$ and λ increases from 0.75, 2.5 to 4.5. The summary statistics are given in Table 2.

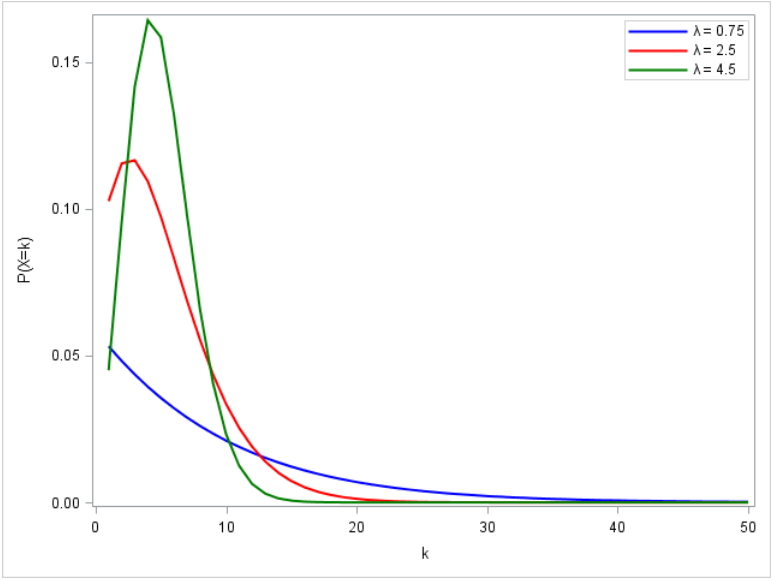


Figure 3: Probability mass functions of the GPD when $E[Y] = 5$ and λ increases.

θ	λ	$E[Y]$	$Var[Y]$	$FI[Y]$
0.15	0.75	5	61.667	12.33
0.5	2.5	5	15	3
0.9	4.5	5	6.111	1.222

Table 2: Summary statistics of distribution displayed in Figure 3.

From Figure 3 and Table 2, given a constant central location, increasing λ results in a smaller variance and smaller dispersion index, $FI[Y]$. The latter is also illustrated in Figure 4 which displays the relationship between the Fisher index of dispersion and λ when $E[Y] = 5$. The GPD is over-dispersed relative to the Poisson distribution which has a Fisher index of 1, but as λ increases whilst $E[Y]$ remains constant the Fisher index of the GPD tends to 1.

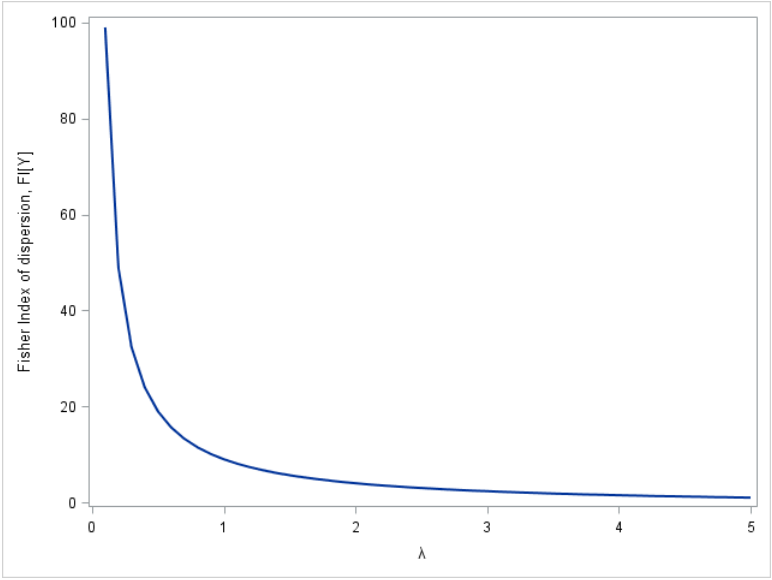


Figure 4: Graph of $FI[Y]$ against λ when $E[Y] = 5$.

Part 2: Keeping one of the parameters constant and varying the expected value.

Figure 5 shows the pmfs of the GPD when the parameter $\lambda = 2$ constant and $E[Y] = \frac{\lambda}{\theta}$ is varied from 5, 10 to 20. The summary statistics are given in Table 3.

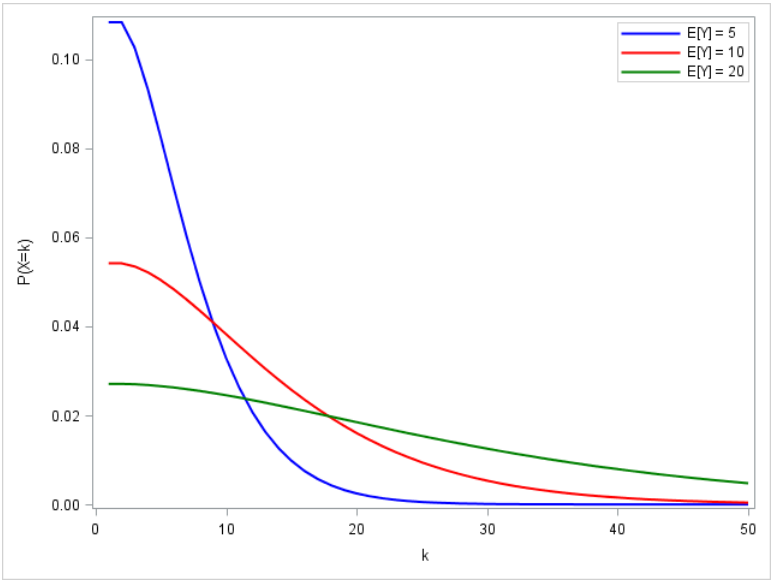


Figure 5: Probability mass functions of the GPD when $\lambda = 2$ and $E[Y]$ is increased .

θ	λ	$E[Y]$	$Var[Y]$	$FI[Y]$
0.4	2	5	20	4
0.2	2	10	90	9
0.1	2	20	380	19

Table 3: Summary statistics of distribution displayed in Figure 5.

From Figure 5 and Table 3, given we keep the parameter λ constant, increasing the central location results in a larger variance and a larger dispersion index. The latter is also illustrated in Figure 7 when $\lambda = 2$ which displays the relationship between the Fisher index of dispersion and $E[Y]$. As $E[Y]$ increases whilst $\lambda = 2$ constant, θ decreases. We then have that from (18) and since $0 < \theta < 1$, the Fisher index increases in a linear function. The GPD is therefore always over-dispersed relative to the Poisson distribution which has a Fisher index of 1.

Figure 6 shows the pmfs of the GPD when the parameter $\theta = 0.5$ constant and $E[Y] = \frac{\lambda}{\theta}$ is varied from 5, 10 to 20. The summary statistics are given in Table 4.

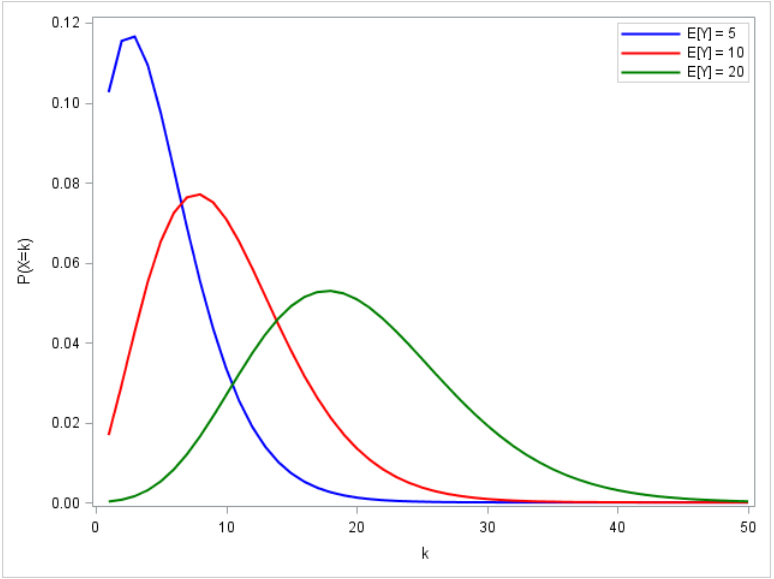


Figure 6: Probability mass functions of the GPD when $\theta = 0.5$ and $E[Y]$ is increased.

θ	λ	$E[Y]$	$Var[Y]$	$FI[Y]$
0.5	2.5	5	15	3
0.5	5	10	30	3
0.5	10	20	60	3

Table 4: Summary statistics of distribution displayed in Figure 6.

From Figure 6 and Table 4, given we keep the parameter θ constant, increasing the central location results in a larger variance but a constant dispersion index greater than one. The latter can also be seen in (18) and is illustrated in Figure 7 which displays the relationship between the Fisher index of dispersion and $E[Y]$ for constant θ. Note that in Figure 6, increasing the central location results in a shift of the pmfs to the right becoming more symmetric in shape.

Graphs showing the relationship between the dispersion index and $E[Y]$ for parameters chosen to be $\lambda = 2$ and $\theta = 0.5$ respectively are given in displayed in Figure 7 below.

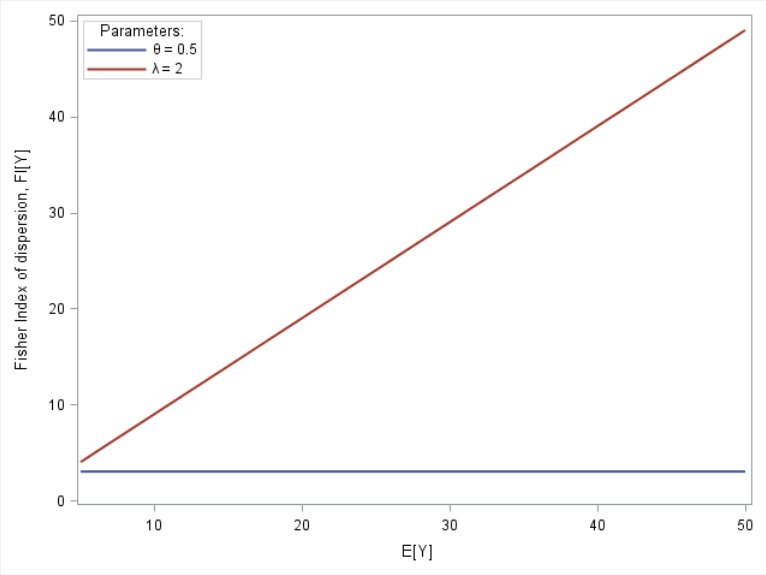


Figure 7: Graph of $FI[Y]$ against $E[Y]$ for constant (i) $\lambda = 2$ and (ii) $\theta = 0.5$.

4.3 Fitting the distribution to traffic accident data

In this part the GPD is fitted to traffic accident data by following the procedure described by Özel and Inal [14]. The data presented by Meintanis [10], given in Table 5 is used and gives the total accidents on a given Sunday and the corresponding number of fatalities that occurred during the years of $1997 - 2004$ in the Groningen region. The data was taken from BRON database recorded by the Transport Ministry in the Netherlands. The GPD is fitted to $Y = \sum_{i=1}^{N} X_i$ where

1. N represents the total number of accidents on Sundays that occur during the years of $1997 - 2004$ in the Groningen region,
2. X_i, $i = 1, 2, 3, \ldots$, represents the number of fatalities of the ith accident and
3. $Y = \sum_{i=1}^{N} X_i$ represents the total number of traffic accident fatalities.

	1997		1998		1999		2000		2001		2002		2003		2004	
MONTH	n_1	y_1	n_2	y_2	n_3	y_3	n_4	y_4	n_5	y_5	n_6	y_6	n_7	y_7	n_8	y_8
JANUARY	6	0	6	0	13	1	11	0	8	0	8	0	11	4	2	0
FEBRUARY	10	0	10	1	7	0	4	0	8	1	8	0	9	0	2	0
MARCH	7	0	13	4	8	0	10	0	6	0	12	0	9	0	3	0
APRIL	11	0	5	0	14	1	15	1	9	0	10	1	7	1	1	1
MAY	12	0	17	2	13	0	18	0	13	2	11	0	12	1	5	0
JUNE	21	1	19	0	14	0	21	1	12	3	12	1	13	0	7	2
JULY	15	0	10	0	14	0	11	1	10	2	4	0	8	0	1	0
AUGUST	11	1	11	1	10	0	8	0	9	0	14	1	6	0	5	0
SEPTEMBER	7	0	11	0	7	0	9	0	22	1	16	1	7	0	8	1
OCTOBER	11	2	13	1	16	1	14	0	15	1	8	1	6	1	2	0
NOVEMBER	15	1	17	1	13	0	13	0	6	0	9	1	11	1	1	0
DECEMBER	5	0	7	0	10	1	10	0	10	0	8	0	5	0	2	0

Table 5: Total accidents on a given Sunday, n_i and the corresponding total number of fatalities, y_i are recorded for each month during the period $1997-2004$ in the Groningen region.

The total number of fatalities can be modeled by the GPD if the following conditions hold:

1. The distribution of N is $POI(\lambda)$.
The hypothesis for this condition is

$$\begin{aligned} H_0 &: \textit{The total number of accidents, } N\textit{, have a Poisson distribution.} \\ H_A &: \textit{The total number of accidents, } N\textit{, is not Poisson distributed.} \end{aligned}$$

We use historical data that the total number of accidents on a given Sunday are generally Poisson distributed over the time interval such that $t > 0$ [9, 10]. A Poisson distribution was fitted to the data using the PROC GENMOD procedure in SAS [1] and from this the parameter is estimated to be $\hat{\lambda} = 9.833$. From the Pearson goodness-of-fit test the null hypothesis is rejected which means the data does not fit a Poisson distribution. However, to rework the results in the article presented by Özel and Inal [14] where they state that the Poisson distribution fits the data well, the Poisson distribution was used further as in the example. We use the fact that the total number of accidents, N is Poisson distributed with estimated parameter as given above.

2. The distribution of the random variables X_i, $i = 1, 2, 3, \ldots$, are $GEO(\theta)$.
The hypothesis for this condition is

$$\begin{aligned} H_0 &: \textit{The number of fatalities, } X_i,\ i = 1, 2, 3, \ldots,\textit{ random variables have a geometric distribution.} \\ H_A &: \textit{The number of fatalities, } X_i,\ i = 1, 2, 3, \ldots,\textit{ do not have a geometric distribution.} \end{aligned}$$

The PROC GENMOD procedure in SAS [1] was also used to fit a geometric distribution to the traffic accident fatalities. This gave $\hat{\theta} = 0.65753$ and from Table 6 the Pearson goodness-of-fit statistic is 5.974 which is less than corresponding critical value at a 5% significance level $\chi^2_\alpha(k-p-1) = \chi^2_{0.05}(3) = 7.815$. As a result H_0 cannot be rejected and the geometric distribution provides a good fit to the number of fatalities, X_i, $i = 1, 2, 3, \ldots$, random variables.
The results of the performed the goodness fit test on the number of fatalities on a given Sunday are given in the following table.

Number of fatalities, X_i	Observed frequency	Expected Frequency
		Geometric
0	59	63.12
1	29	21.62
2	5	7.40
3	1	2.54
4	2	0.87
Total	96	95.55
$\chi^2 - statistic$		5.974
degrees of freedom, d.o.f.		3

Table 6: The geometric distribution fit of the observed frequency for the number of fatalities.

3. The random variables N and X_i, $i = 1, 2, 3, \ldots$, are independent.
The hypothesis of this condition is

$$\begin{aligned} H_0 &: \rho_s = 0 \\ H_A &: \rho_s \neq 0. \end{aligned}$$

The Spearman's correlation coefficient was calculated as $\rho_s = 0.23128$ with corresponding $p - value = 1510$. Therefore H_0 cannot be rejected at a 5% significance level. Thus the number of fatalities, X_i, $i = 1, 2, 3, \ldots$ and the total number of accidents N are independent.

We have that all the conditions above hold. Therefore we assume $N \sim POI\left(\hat{\lambda}\right)$ represents the total number of accidents on Sundays that occur during the years of $1997 - 2004$ in the Groningen region and the random variables $X_i \sim GEO\left(\hat{\theta}\right)$, $i = 1, 2, 3, \ldots$, represents the number of fatalities of the ith accident. As a result assuming the valid use of the Poisson distribution, the total number of fatalities $Y = \sum_{i=1}^{N} X_i$ on a given Sunday are represented by $Y \sim GEOPOI(\lambda, \theta)$ where such that for the traffic accidents can be explained by the GPD.
The figure below follows from the probabilities computed in (11) and the parameters estimated from the traffic accident data $\left(i.e.\ \theta = 0.65753\ and\ \hat{\lambda} = 9.833\right)$. We have

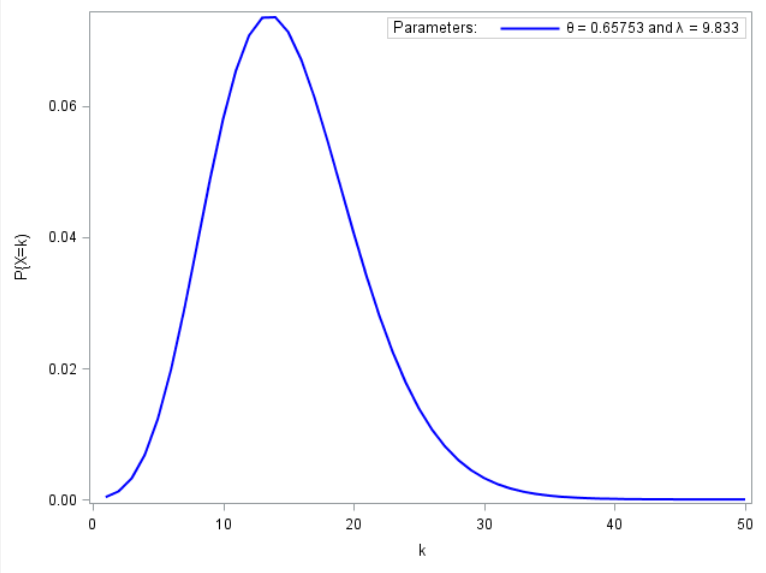


Figure 8: The p.m.f. of the total number of accident fatalities explained by the GPD with parameters $\hat{\lambda} = 9.833$ and $\hat{\theta} = 0.65753$.

5 Conclusion

In this paper some of the properties of the geometric Poisson distribution (also called the P*ó*lya-Aeppli distribution) were studied. The GPD is a unique instance of the compound Poisson distribution. The aim of the study was to show how an explicit probability function of the P*ó*lya-Aeppli distribution could be derived, to derive some properties of the distribution and then conclude by demonstrating the distributions practical relevance by fitting the distribution to a traffic accident database as an example. It is shown that the probability function is easily derived, although an algorithm similar to that of *Ö*zel and *İ*nal [14] was not computed due to its complexity and rather discussed. The exact formula in (11) was used to create some of the graphs. The relationships between the parameters and the Fisher index of dispersion were also analyzed in depth. The paper is concluded with an application on traffic accident data. In our analysis we found that the Poisson distribution was not a good fit for the total number of accidents. This can be the focus of further studies to determine a more appropriate compound distribution for the data given. In Actuarial Statistics, compound Poisson distributions are commonly fitted to financial and insurance data as seen in Sundt and Vernic [19]. It would be of interest to investigate how well the GPD could be fitted to financial data in future.

References

[1] *SAS Software, Version 9.4 of SAS System for Windows, Copyright 2016, SAS Institute Inc. Cary, NC, USA.*

[2] A. Aeppli. *Zur Theorie verketteter Wahrscheinlichkeiten: Markoffsche Ketten höherer Ordnung.* PhD thesis, University of Zurich, 1924.

[3] M. Anwar and M. Ahmad. On some properties of the geometric Poisson distribution. *Pakistan Journal of Statistics*, 30(2):233–244, 2014.

[4] N. Ata and G. Özel. Survival functions for the frailty models based on the discrete compound Poisson process. *Journal of Statistical Computation and Simulation*, 83(11):2105–2116, 2013.

[5] L.J. Bain and M. Engelhardt. *Introduction to Probability and Mathematical Statistics.* Brooks/Cole, 2nd edition, 1992.

[6] C. Chen, P.H. Randolph, and T. Liou. Using CUSUM control schemes for monitoring quality levels in compound Poisson production environment: the geometric Poisson process. *Quality Engineering*, 17(2):207–217, 2005.

[7] R.A. Fisher. The effect of methods of ascertainment upon the estimation of frequencies. *Annals of Eugenics*, 6(1):13–25, 1934.

[8] N.L. Johnson, S. Kotz, and A.W. Kemp. *Univariate Discrete Distributions.* Wiley, Third edition, 1992.

[9] R.E. Leiter and M.A. Hamdan. Some bivariate probability models applicable to traffic accidents and fatalities. *International Statistical Review*, 41(1):87–100, 1973.

[10] S.G. Meintanis. A new goodness-of-fit test for certain bivariate distributions applicable to traffic accidents. *Statistical Methodology*, 4(1):22–34, 2007.

[11] L.D. Minkova and N. Balakrishnan. Compound weighted Poisson distributions. *Metrika*, 76(4):543–558, 2013.

[12] L.D. Minkova and N. Balakrishnan. On a bivariate Pólya-Aeppli distribution. *Commiunications in Statistics - Theory and Methods*, 43(23):5026–5038, 2014.

[13] G. Nuel. Cumulative distribution function of a geometric Poisson distribution. *Journal of Statistical Computation and Simulation*, 78(3):385–394, 2008.

[14] G. Özel and C. Inal. The probability function of a geometric Poisson distribution. *Journal of Statistical Computation and Simulation*, 80(5):479–487, 2010.

[15] G. Pólya. Sur quelques points de la théorie des probabilités. In *Annales de l'institut Henri Poincaré*, volume 1, pages 117–161, 1930.

[16] P. Randolph and M. Sahinoglu. A stopping rule for a compound Poisson random variable. *Applied Stochastic Models and Data Analysis*, 11(2):135–143, 1995.

[17] S. Robin, S. Schbath, and V. Vandewalle. Statistical test to compare motif count exceptionalities. *Bioinformatics*, 8(84):1–20, 2007.

[18] R.J. Rosychuk, C. Huston, and N.G.N. Prasad. Spatial event cluster detection using a compound Poisson distribution. *Biometrics*, 62(2):465–470, 2006.

[19] B. Sundt and R. Vernic. *Recursions for Convolutions and Compound Distributions with Insurance Applications.* Springer Science & Business Media, 2009.

Appendix

The code files listed below use SAS Software [1] to produce the figures throughout the document.

```
*SAS code used to produce Figure 1;
ODS GRAPHCS ON;
DATA FIGURE1;
MU = 10;
THETA = 0.25;
THETA1 = 0.5;
THETA2 = 0.75;
LAMBDA = MU*THETA;
LAMBDA1 = MU*THETA1;
LAMBDA2 = MU*THETA2;
CUM = 0;
CUM1 = 0;
CUM2 = 0;
DO k = 1 TO 50;
PROB = 0;
PROB1 = 0;
PROB2 = 0;
DO n = 1 TO k;
TERM = EXP(-LAMBDA)*(LAMBDA**n)/FACT(n)*COMB(k-1,n-1)*(THETA**n)*((1-THETA)**(k-n));
PROB = PROB+TERM;
TERM1 = EXP(-LAMBDA1)*(LAMBDA1**n)/FACT(n)*COMB(k-1,n-1)*(THETA1**n)*((1-THETA1)**(k-n));
PROB1 = PROB1+TERM1;
TERM2 = EXP(-LAMBDA2)*(LAMBDA2**n)/FACT(n)*COMB(k-1,n-1)*(THETA2**n)*((1-THETA2)**(k-n));
PROB2 = PROB2+TERM2;
END;
CUM = CUM+PROB;
CUM1 = CUM1+PROB1;
CUM2 = CUM2+PROB2;
OUTPUT;
END;
PROC PRINT DATA = FIGURE1;
VAR PROB CUM;
VAR PROB1 CUM1;
VAR PROB2 CUM2;
GOPTIONS RESET = ALL;
ODS ESCAPECHAR = "^";
PROC SGPLOT DATA = FIGURE1;
SERIES X=k Y=PROB /LINEATTRS = (COLOR = BLUE THICKNESS = 2) LEGENDLABEL = "^{UNICODE THETA} = 0.25";
SERIES X=k Y=PROB1 /LINEATTRS = (COLOR = RED THICKNESS = 2) LEGENDLABEL = "^{UNICODE THETA} = 0.5";
SERIES X=k Y=PROB2 /LINEATTRS = (COLOR = GREEN THICKNESS = 2) LEGENDLABEL = "^{UNICODE THETA} = 0.75";
XAXIS LABEL = "k";
YAXIS LABEL =  "P(X=k)";
KEYLEGEND / ACROSS = 1 BORDER LOCATION = INSIDE POSITION = TOPRIGHT;
RUN;
ODS GRAPHICS OFF;
```

```
*SAS code used to produce Figure 2;
ODS GRAPHICS ON;
DATA FI_GEO;
MU = 10;
```

```
DO THETA = 0 TO 1 BY 0.05;
LAMBDA = MU*THETA;
SIGMA_SQ = LAMBDA*(2-THETA)/THETA**2;
FI = SIGMA_SQ/MU;
OUTPUT;
END;
RUN;
ODS ESCAPECHAR = "^";
PROC PRINT DATA = FI_GEO LABEL;
LABEL THETA = "^{UNICODE theta}";
LABEL MU = "^{UNICODE mu}";
LABEL LAMBDA = "^{UNICODE lambda}";
LABEL SIGMA_SQ = "VAR[Y]";
LABEL FI = "FI[Y]";
GOPTIONS RESET = ALL;
ODS ESCAPECHAR = "^";
PROC SGPLOT DATA = FI_GEO;
SERIES X = THETA Y = FI / LINEATTRS = (THICKNESS = 2);
XAXIS LABEL = "^{UNICODE THETA}";
YAXIS LABEL =  "Fisher Index of dispersion, FI[Y]";
RUN;

ODS GRAPHICS OFF;

*SAS code used to produce Figure 3;
DATA FIGURE3;
MU = 5;
LAMBDA = 0.75;
LAMBDA1 = 2.5;
LAMBDA2 = 4.5;
THETA = LAMBDA/MU;
THETA1 = LAMBDA1/MU;
THETA2 = LAMBDA2/MU;
CUM = 0;
CUM1 = 0;
CUM2 = 0;
DO k = 1 TO 50;
PROB = 0;
PROB1 = 0;
PROB2 = 0;
DO n = 1 TO k;
TERM = EXP(-LAMBDA)*(LAMBDA**n)/FACT(n)*COMB(k-1,n-1)*(THETA**n)*((1-THETA)**(k-n));
PROB = PROB+TERM;
TERM1 = EXP(-LAMBDA1)*(LAMBDA1**n)/FACT(n)*COMB(k-1,n-1)*(THETA1**n)*((1-THETA1)**(k-n));
PROB1 = PROB1+TERM1;
TERM2 = EXP(-LAMBDA2)*(LAMBDA2**n)/FACT(n)*COMB(k-1,n-1)*(THETA2**n)*((1-THETA2)**(k-n));
PROB2 = PROB2+TERM2;
END;
CUM = CUM+PROB;
CUM1 = CUM1+PROB1;
CUM2 = CUM2+PROB2;
OUTPUT;
END;
```

```
PROC PRINT DATA = FIGURE3;
VAR PROB CUM;
VAR PROB1 CUM1;
VAR PROB2 CUM2;
GOPTIONS RESET = ALL;
ODS ESCAPECHAR = "^";
PROC SGPLOT DATA = FIGURE3;
SERIES X=k Y=PROB /LINEATTRS = (COLOR = BLUE THICKNESS = 2) LEGENDLABEL = "^{UNICODE LAMBDA} = 0.75";
SERIES X=k Y=PROB1 /LINEATTRS = (COLOR = RED THICKNESS = 2) LEGENDLABEL = "^{UNICODE LAMBDA} = 2.5";
SERIES X=k Y=PROB2 /LINEATTRS = (COLOR = GREEN THICKNESS = 2) LEGENDLABEL = "^{UNICODE LAMBDA} = 4.5";
XAXIS LABEL = "k";
YAXIS LABEL =  "P(X=k)";
KEYLEGEND / ACROSS = 1 BORDER LOCATION = INSIDE POSITION = TOPRIGHT;
RUN;
ODS GRAPHICS OFF;

*SAS code used to produce Figure 4;
ODS GRAPHICS ON;
DATA FI_POISSON;
MU = 5;
DO LAMBDA = 0 TO 5 BY 0.1;
THETA = LAMBDA/MU;
SIGMA_SQ = LAMBDA*(2-THETA)/THETA**2;
FI = SIGMA_SQ/MU;
OUTPUT;
END;
RUN;
ODS ESCAPECHAR = "^";
PROC PRINT DATA = FI_POISSON LABEL;
LABEL THETA = "^{UNICODE theta}";
LABEL LAMBDA = "^{UNICODE lambda}";
LABEL MU = "^{UNICODE mu}";
LABEL SIGMA_SQ = "VAR[Y]";
LABEL FI = "FI[Y]";
GOPTIONS RESET = ALL;
ODS ESCAPECHAR = "^";
PROC SGPLOT DATA = FI_POISSON;
SERIES X = LAMBDA Y = FI / LINEATTRS = (THICKNESS = 2);
XAXIS LABEL = "^{UNICODE LAMBDA}";
YAXIS LABEL =  "Fisher Index of dispersion, FI[Y]";
RUN;

ODS GRAPHICS OFF;

*SAS code used to produce Figure 5;
DATA FIGURE5;
LAMBDA = 2;
THETA = LAMBDA/5;
THETA1 = LAMBDA/10;
THETA2 = LAMBDA/20;
CUM = 0;
CUM1 = 0;
CUM2 = 0;
DO k = 1 TO 50;
```

```
PROB = 0;
PROB1 = 0;
PROB2 = 0;
DO n = 1 TO k;
TERM = EXP(-LAMBDA)*(LAMBDA**n)/FACT(n)*COMB(k-1,n-1)*(THETA**n)*((1-THETA)**(k-n));
PROB = PROB+TERM;
TERM1 = EXP(-LAMBDA)*(LAMBDA**n)/FACT(n)*COMB(k-1,n-1)*(THETA1**n)*((1-THETA1)**(k-n));
PROB1 = PROB1+TERM1;
TERM2 = EXP(-LAMBDA)*(LAMBDA**n)/FACT(n)*COMB(k-1,n-1)*(THETA2**n)*((1-THETA2)**(k-n));
PROB2 = PROB2+TERM2;
END;
CUM = CUM+PROB;
CUM1 = CUM1+PROB1;
CUM2 = CUM2+PROB2;
OUTPUT;
END;
PROC PRINT DATA = FIGURE5;
VAR PROB CUM;
VAR PROB1 CUM1;
VAR PROB2 CUM2;
GOPTIONS RESET = ALL;
ODS ESCAPECHAR = "^";
PROC SGPLOT DATA = FIGURE5;
SERIES X = k Y = PROB / LINEATTRS = (COLOR = BLUE THICKNESS = 2) LEGENDLABEL = "E[Y] = 5";
SERIES X = k Y = PROB1 / LINEATTRS = (COLOR = RED THICKNESS = 2) LEGENDLABEL = "^E[Y] = 10";
SERIES X = k Y = PROB2 / LINEATTRS = (COLOR = GREEN THICKNESS = 2) LEGENDLABEL = "E[Y] = 20";
XAXIS LABEL = "k";
YAXIS LABEL =  "P(X=k)";
KEYLEGEND / ACROSS = 1 BORDER LOCATION = INSIDE POSITION = TOPRIGHT;
RUN;
ODS GRAPHICS OFF;

*SAS code used to produce Figure 6;
DATA FIGURE6;
THETA = 0.5;
LAMBDA = THETA*5;
LAMBDA1 = THETA*10;
LAMBDA2 = THETA*20;
CUM = 0;
CUM1 = 0;
CUM2 = 0;
DO k = 1 TO 50;
PROB = 0;
PROB1 = 0;
PROB2 = 0;
DO n = 1 TO k;
TERM = EXP(-LAMBDA)*(LAMBDA**n)/FACT(n)*COMB(k-1,n-1)*(THETA**n)*((1-THETA)**(k-n));
PROB = PROB+TERM;
TERM1 = EXP(-LAMBDA1)*(LAMBDA1**n)/FACT(n)*COMB(k-1,n-1)*(THETA**n)*((1-THETA)**(k-n));
PROB1 = PROB1+TERM1;
TERM2 = EXP(-LAMBDA2)*(LAMBDA2**n)/FACT(n)*COMB(k-1,n-1)*(THETA**n)*((1-THETA)**(k-n));
PROB2 = PROB2+TERM2;
END;
```

```
CUM = CUM+PROB;
CUM1 = CUM1+PROB1;
CUM2 = CUM2+PROB2;
OUTPUT;
END;
PROC PRINT DATA = FIGURE6;
VAR PROB CUM;
VAR PROB1 CUM1;
VAR PROB2 CUM2;
GOPTIONS RESET = ALL;
ODS ESCAPECHAR = "^";
PROC SGPLOT DATA = FIGURE6;
SERIES X = k Y = PROB / LINEATTRS = (COLOR = BLUE THICKNESS = 2) LEGENDLABEL = "E[Y] = 5";
SERIES X = k Y = PROB1 / LINEATTRS = (COLOR = RED THICKNESS = 2) LEGENDLABEL = "^E[Y] = 10";
SERIES X = k Y = PROB2 / LINEATTRS = (COLOR = GREEN THICKNESS = 2) LEGENDLABEL = "E[Y] = 20";
XAXIS LABEL = "k";
YAXIS LABEL =  "P(X=k)";
KEYLEGEND / ACROSS = 1 BORDER LOCATION = INSIDE POSITION = TOPRIGHT;
RUN;
ODS GRAPHICS OFF;

*SAS code used to produce Figure 7;
ODS GRAPHICS ON;

DATA THETA_LAMBDA_MU;
THETA = 0.5;
LAMBDA = 2;
INPUT MU;
DATALINES;
5
10
20
30
50
;
RUN;
DATA FI_IND;
SET THETA_LAMBDA_MU;
LAMBDA1 = MU*THETA;
SIGMA_SQ1 = LAMBDA1*(2-THETA)/THETA**2;
FI_1 = SIGMA_SQ1/MU;
THETA1 = LAMBDA/MU;
SIGMA_SQ2 = LAMBDA*(2-THETA1)/THETA1**2;
FI_2 = SIGMA_SQ2/MU;
CHISQ = QUANTILE('CHISQ',0.05,2);
PROC PRINT DATA = FI_IND LABEL;
LABEL THETA = "^{UNICODE theta}";
LABEL MU = "^{UNICODE mu}";
LABEL LAMBDA = "^{UNICODE lambda}";
LABEL SIGMA_SQ = "VAR[Y]";
LABEL FI = "FI[Y]";
GOPTIONS RESET = ALL;
ODS ESCAPECHAR = "^";
```

```
PROC SGPLOT DATA = FI_IND;
SERIES X = MU Y = FI_1 / LINEATTRS = (THICKNESS = 2) LEGENDLABEL = "^{UNICODE THETA} = 0.5";
SERIES X = MU Y = FI_2 / LINEATTRS = (THICKNESS = 2) LEGENDLABEL = "^{UNICODE LAMBDA} = 2";
XAXIS LABEL = "E[Y]";
YAXIS LABEL =  "Fisher Index of dispersion, FI[Y]";
KEYLEGEND / ACROSS = 1 BORDER LOCATION = INSIDE POSITION = TOPLEFT TITLE = "Parameters:";
RUN;

ODS GRAPHICS OFF;

*SAS code used on the traffic accident application of GPD;
ODS GRAPHICS ON;

DATA NUMBER;
INPUT MONTH $ YEAR N Y;
LABEL   N = 'Total Sunday accidents in year.'
        Y = 'Number of fatalities in corresponding year.';
DATALINES;
JANUARY  1997   6   0
JANUARY  1998   6   0
JANUARY  1999   13  1
JANUARY  2000   11  0
JANUARY  2001   8   0
JANUARY  2002   8   0
JANUARY  2003   11  4
JANUARY  2004   2   0
FEBRUARY 1997   10  0
FEBRUARY 1998   10  1
FEBRUARY 1999   7   0
FEBRUARY 2000   4   0
FEBRUARY 2001   8   1
FEBRUARY 2002   8   0
FEBRUARY 2003   9   0
FEBRUARY 2004   2   0
MARCH    1997   7   0
MARCH    1998   13  4
MARCH    1999   8   0
MARCH    2000   10  0
MARCH    2001   6   0
MARCH    2002   12  0
MARCH    2003   9   0
MARCH    2004   3   0
APRIL    1997   11  0
APRIL    1998   5   0
APRIL    1999   14  1
APRIL    2000   15  1
APRIL    2001   9   0
APRIL    2002   10  1
APRIL    2003   7   1
APRIL    2004   1   1
MAY      1997  12   0
MAY      1998  17   2
MAY      1999  13   0
```

```
MAY       2000  18   0
MAY       2001  13   2
MAY       2002  11   0
MAY       2003  12   1
MAY       2004  5    0
JUNE      1997  21   1
JUNE      1998  19   0
JUNE      1999  14   0
JUNE      2000  21   1
JUNE      2001  12   3
JUNE      2002  12   1
JUNE      2003  13   0
JUNE      2004  7    2
JULY      1997  15   0
JULY      1998  10   0
JULY      1999  14   0
JULY      2000  11   1
JULY      2001  10   2
JULY      2002  4    0
JULY      2003  8    0
JULY      2004  1    0
AUGUST    1997  11   1
AUGUST    1998  11   1
AUGUST    1999  10   0
AUGUST    2000  8    0
AUGUST    2001  9    0
AUGUST    2002  14   1
AUGUST    2003  6    0
AUGUST    2004  5    0
SEPTEMBER 1997 7    0
SEPTEMBER 1998 11   0
SEPTEMBER 1999 7    0
SEPTEMBER 2000 9    0
SEPTEMBER 2001 22   1
SEPTEMBER 2002 16   1
SEPTEMBER 2003 7    0
SEPTEMBER 2004 8    1
OCTOBER   1997 11   2
OCTOBER   1998 13   1
OCTOBER   1999 16   1
OCTOBER   2000 14   0
OCTOBER   2001 15   1
OCTOBER   2002 8    1
OCTOBER   2003 6    1
OCTOBER   2004 2    0
NOVEMBER  1997 15   1
NOVEMBER  1998 17   1
NOVEMBER  1999 13   0
NOVEMBER  2000 13   0
NOVEMBER  2001 6    0
NOVEMBER  2002 9    1
NOVEMBER  2003 11   1
NOVEMBER  2004 1    0
```

```
DECEMBER  1997 5    0
DECEMBER  1998 7    0
DECEMBER  1999 10   1
DECEMBER  2000 10   0
DECEMBER  2001 10   0
DECEMBER  2001 8    0
DECEMBER  2003 5    0
DECEMBER  2004 2    0
;
PROC PRINT DATA = NUMBER;
RUN;

PROC FREQ DATA = NUMBER;
TABLES Y / CHISQ EXPECTED OUT = NUMBER1;
RUN;
PROC PRINT DATA = NUMBER1;
RUN;

*2x2 Contigency Table between N and Y;
PROC FREQ DATA = NUMBER;
TABLES N*Y / CHISQ EXPECTED NOCOL NOROW NOPERCENT OUT = NUMBER2;
RUN;

*Spearman Correlation Coefficient Test;
PROC CORR DATA = NUMBER2 SPEARMAN;
VAR N;
WITH Y;
RUN;

*Estimating the Poisson parameter;
PROC GENMOD DATA = NUMBER;
MODEL N =  / DIST = POISSON;
ODS OUTPUT PARAMETERESTIMATES = POI_ESTIMATE;
RUN;
DATA POI_ESTIMATE;
SET;
IF _N_ = 1;
LAMBDA = EXP(ESTIMATE);
LOWER = EXP(LOWERWALDCL);
UPPER = EXP(UPPERWALDCL);
RUN;
PROC PRINT DATA = POI_ESTIMATE;
VAR LAMBDA LOWER UPPER;
RUN;

*Estimatiing the geometric parameter;
PROC GENMOD DATA = NUMBER;
MODEL Y = /DIST = NEGBIN SCALE = 1 NOSCALE;
ODS OUTPUT PARAMETERESTIMATES = GEO_ESTIMATE;
RUN;
DATA GEO_ESTIMATE;
SET;
IF _N_=1;
```

```
THETA = 1/(1+EXP(ESTIMATE));
LOWER_P = 1/(1+EXP(LOWERWALDCL));
UPPER_P = 1/(1+EXP(UPPERWALDCL));
RUN;
PROC PRINT DATA = GEO_ESTIMATE;
VAR THETA LOWER_P UPPER_P;
RUN;

*Figure 8;
DATA FIGURE8;
SET GEO_ESTIMATE;
SET POI_ESTIMATE;
CUM = 0;
DO k = 1 TO 50;
PROB = 0;
DO m = 1 TO k;
TERM = EXP(-LAMBDA)*(LAMBDA**m)/FACT(m)*COMB(k-1,m-1)*(THETA**m)*((1-THETA)**(k-m));
PROB = PROB+TERM;
END;
CUM = CUM+PROB;
OUTPUT;
END;
PROC PRINT DATA = FIGURE8;
VAR PROB CUM;
GOPTIONS RESET = ALL;
ODS ESCAPECHAR = "~";
PROC SGPLOT DATA = FIGURE8;
SERIES X = k Y = PROB / LINEATTRS = (COLOR = BLUE THICKNESS = 2)
LEGENDLABEL = "~{UNICODE THETA} = 0.65753 and ~{UNICODE LAMBDA}  = 9.833" ;
XAXIS LABEL = "k";
YAXIS LABEL =  "P{X=k)";
KEYLEGEND / ACROSS = 1 BORDER LOCATION = INSIDE POSITION = TOPRIGHT TITLE = "Parameters:";
RUN;

ODS GRAPHICS OFF;
```

Approximate Bayesian Computation

Louisa Somerville 11032074

STK 795 Research Report

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

Supervisors: Mrs J van Niekerk; Mr H Masoumi Karakani

Department of Statistics, University of Pretoria

02 November 2016

Abstract

There is an increasing demand for modern statistical analysis that brings about an issue of complex; indeterminable likelihood functions of the models. Approximate Bayesian Computation (ABC) methods address this problem by bypassing the evaluation of the likelihood function in order to widen the scope of application for statistical inference. This report aims at laying out the evolution and extensions of the original ABC algorithm by applying it to theoretical and real life applications.

Declaration

I, *Louisa Mary Somerville,* 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.

_ _
Louisa Somerville

_ _
Janet van Niekerk

_ _
Hossein Masoumi Karakani

_ _
Date

Acknowledgments

The author would like to thank the Center for Artificial Intelligence Research (CAIR) for financial support in the form of a postgraduate bursary.

Contents

1 Introduction **6**

2 Literature review **6**

3 ABC algorithms (Methodology) **7**
3.1 Theoretical background . . . 7
3.2 ABC algorithms . . . 7

4 Application **8**
4.1 Practical . . . 8
4.2 Example . . . 9
4.3 Simulation . . . 9
4.4 Real data set: Human demographic history . . . 10
4.4.1 Background . . . 10
4.4.2 Demographic models . . . 11
4.4.3 Model selection . . . 11
4.4.4 Goodness-of-fit test . . . 11
4.4.5 Posterior predictive checks . . . 12
4.4.6 Cross-validation . . . 13
4.4.7 Parameter inference . . . 13
4.5 Model comparison and application . . . 14
4.6 Improvements . . . 15

5 Conclusion **15**

6 Appendix **16**
6.1 Simulation . . . 16
6.1.1 Table 1 . . . 16
6.1.2 Figure 1 . . . 16
6.2 Real data set: Human demographic history . . . 17
6.2.1 Figure 2 . . . 17
6.2.2 Figure 3 . . . 18
6.2.3 Figure 4 . . . 18
6.2.4 Figure 5 . . . 18
6.2.5 Figure 6 . . . 18
6.2.6 Figure 7 . . . 18

List of Figures

1 **Simulated data: MCMC posterior distribution** . . . 10
2 **Misclassification proportions for the three models** . . . 11
3 **Histogram of the test statistic for goodness of fit assuming a bottleneck model.** . . . 12
4 **Posterior predictive checks for the European data under the bottleneck model** . . . 12
5 **Cross-validation for parameter inference.** . . . 13
6 **Histogram of posterior sample of Na** . . . 14
7 **ABC regression diagnostics for the estimation of the posterior distribution of Na** . . . 14

List of Tables

1 **Simulated data: Parameter estimation** . . . 10

1 Introduction

Approximate Bayesian computation (ABC) methods are comprised of several computational methods [6]. The likelihood function of a model is the probability of the observed data being within a statistical model [6]. Although the likelihood function plays a major role in statistical inference, ABC methods operate by bypassing the evaluation of the likelihood function in order to widen the scope of application for statistical inference. This allows for models of all variation to be analyzed and eliminates the challenge of parameter estimation and model selection [6].

One of the most prominent uses of ABC today lies in the biological field. During the formulation phase of ABC, the method was applied to test the genealogy of DNA. It was used to address the problem of determining the posterior distribution of the time of the most recent common ancestor of individuals [6]. It is also used to make inferences about evolutionary genetics; population genetics; melanoma cell research; cosmology and biodiversity numbers in tropical rain forests.

Other areas of application of ABC include forecasting insurance loss payments [2]; smoothing data; goodness-of-fit statistics; summary statistic weights; linear regression; Gaussian processes and differential equations.

This paper aims at laying out the evolution and extensions of the original ABC algorithm by applying it to theoretical and real life applications. Firstly, we will use the ABC algorithm against a simulated normally distributed data set and compare the ABC parameter estimations against the theoretical normal estimates. The Monte-Carlo algorithm will be tested against the theoretical normal data set to compare the ABC approach to theoretical values. Secondly, we will also apply the ABC method to a real life data set and test how accurate the algorithm is on real time data; as well as analyze any pitfalls and remedies that can be developed to improve the algorithm in real life applications.

2 Literature review

ABC thinking began in the 1980's by a man called Donald Rubin [6]. Rubin was determined to develop a concept that would allow statisticians to break free from the limitations of only working with analytically controllable models. The issue in modern day statistics is that, to a large part, the models analyzed are analytically uncontrollable. Rubin foresaw this issue and created a computational method which measures the posterior distribution of interest [4].

In 1984, Peter Diggle and Richard Gratton extended the research of approximating the likelihoods of uncontrollable models [9]. They based their approach on conducting various simulations of likelihoods within a parameter grid [9]. Therefore, Diggle and Gratton managed to extend Rubin's concept by introducing simulations, however they only approximate the likelihood and not the posterior distribution as seen in today's form of ABC [9].

The preliminary work of considering ABC algorithms for posterior prediction was seen in an article written by Tavare et al. [10]. They generated a sample from the posterior model parameters by trial and error in comparing the errors of simulated data versus real data [9]. Jonathan Pritchard refined the sample models and ABC methods were becoming more practically relevant [8].

Rubin's initial approach took on many forms and variations with several algorithms adapting the method to real time data [9]. The Monte Carlo algorithm is one of the most commonly used algorithms in modern day statistics [4]. Once the ABC method was developed there was no longer an obstacle to analyze and estimate previously uncontrollable models. This allowed for a wider spread of models that could be analyzed and is used in several applications today.

The concept of ABC is introduced by looking at the beginnings of ABC and how it has adapted over time [9, 6]. The basic ABC method and the primary ABC rejection algorithm are understood before the development of more efficient inference algorithms for real-time constraints is covered [6]. Research concerning real time inference is studied by addressing a survey that outlines Bayesian network inference algorithms [4].

The final algorithm, the likelihood-free population Monte Carlo sampler, is a commonly used algorithm in modern day statistical inference [6]. This algorithm is based on a sequence of simulated samples and importance weights.

A thorough timeline of the evolution of the ABC method is discussed. This outlines the relationship that exists between these algorithms [11]. Several pressing problems brought about by real-time Bayesian network inference is outlined to evaluate the practicality of this method [2].

Once the theoretical understanding of the workings of ABC is covered, it is applied and compared to other models to determine the accuracy of the method. One of the approaches is to use rejection-sampling [3]. A brief account of the pitfalls and remedies of this method are discussed which include the approximation, and not exact, use of the posterior distribution [9].

3 ABC algorithms (Methodology)

3.1 Theoretical background

In Bayesian statistics the Bayes rule is defined as [9]:

$$\mathrm{P}(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} \tag{1}$$

Where θ denotes the observed parameter values; D denotes the data set; P(D|θ) denotes the likelihood function; P(θ) denotes the prior; P($\theta|D$) denotes the posterior and P(D) denotes the marginal likelihood.

In common practice, prior distributions are chosen in order to make further anslysis of the prior as simple as possible [9]. This is sometimes done by factorizing the joint distributions of the observations of the parameter with regards to a combination of their conditional probabilities [9]. For many practical applications, the likelihood function, P(D|θ), is unavailable [6] or very costly to evaluate [9]. Hence, in order to overcome this problem ABC methods will be considered.

The way in which ABC methods avoid the issue of intractable likelihood functions is by approximating the likelihood functions through simulations [9]. The basic method is achieved through the following steps:

1. Generate a set of parameter observations, θ^*, by sampling from the prior distribution [5].
2. A new data set , D*, is simulated with regards to the parameters in step 1 [5].
3. If the new data set differs too much from the original data set the parameter observation is discarded.

The general decision rule is based on evaluating the difference between the summary statistics of the original data set, θ, and the summary statistics of the simulated data set, θ^* [5].

The final result provides a sample of parameter values that are approximately distributed in line with the posterior distribution without evaluating the likelihood function [9]. The above outlined steps portray a general Bayesian analysis [3]: Define a model, fit the model to data and improve the model by checking the goodness-of-fit with the posterior distribution [3].

From the first ABC applications the method and algorithms used to apply ABC have evolved. By identifying areas of fault the algorithm has been remodeled to be applicable to the current demand for ABC application. A brief overview of the evolution of the ABC algorithm will be discussed in the following section.

3.2 ABC algorithms

The initial ABC algorithm by Rubin [9] is called the likelihood-free rejection sampler and takes on the form of an accept-reject method [6]. A new set of parameters, θ^*, is sampled from the prior, $P(\theta)$, and depending on how similar the simulated value is to the sample value the parameter may be accepted or rejected [6]. Rubin outlined that this method would not suit cases where the likelihood is unattainable, however the outcome gives a better understanding of the posterior distributions [6].

Likelihood-free rejection sampler (Algorithm 1) [6]:

Step 1: Generate a set of parameter observations, θ^*, from the prior distribution.
Step 2: Generate a new data set from the parameter estimates in Step 1. Let D* denote the new data set.
Step 3: Repeat until D*=y, where y is 'true' if the simulated sample is almost identical to the observed sample.
Step 4: Set $\theta_i = \theta^*$.
Step 5: Repeat steps 1-4 N times.

Pritchard extended Rubin's algorithm [9] by adapting the first algorithm to continuous cases. This algorithm is known as the likelihood-free rejection sampler 2. This algorithm was more applicable than Rubin's algorithm but still had some shortcomings that needed to be addressed. In practice, the algorithm was not adapting well to cases where the prior distribution was non informative.

Likelihood-free rejection sampler 2 (Algorithm 2) [6]:

Step 1 & 2 are the same as Algorithm 1.
Step 3: Repeat until the distance between the two samples is below a given tolerance level.
Step 4: Set $\theta_i = \theta^*$.
Step 5: Repeat steps 1-4 N times.

Marjoram et al. formulated the likelihood-free MCMC sampler to address the problems faced in Algorithm 2. The probability to accept in this algorithm is not subject to calculations of the likelihood. Therefore, the likelihood-free MCMC sampler was the first algorithm to operate true to ABC requirements. Majoram et al. developed an effective ABC algorithm that managed to achieve the aim of ABC by bypassing likelihood calculations however the efficiency of the algorithm could be improved [7].

Likelihood-free MCMC sampler (Algorithm 3) [6]:

Steps 1 & 2 are the same as algorithm 1.
Step 3: Acceptance probability is calculated with various distance and tolerance levels and excludes any calculation of the likelihood.
Step 4: Set $\theta_i = \theta^*$.
Step 5: Repeat steps 1-4 N times.

In order to improve the efficiency of Algorithm 3 the likelihood-free population Monte Carlo sampler was developed. The algorithm operates in terms of importance sampling by using sequential techniques that increase the efficiency of the ABC algorithm by adapting to the target population. From the improvements of the original algorithm the Monte Carlo sampler is one of the most commonly used algorithms in modern day statistics.

Likelihood-free Monte Carlo sampler (Algorithm 4) [6]:

Steps 1 & 2 are the same as Algorithm 1.
Step 3: Acceptance probability is calculated using random walk scale and decreasing tolerance thresholds.
Step 4: Set $\theta_i = \theta^*$.
Step 5: Repeat steps 1-4 N times.

4 Application

4.1 Practical

ABC algorithms are available in several statistical software [9]. Choosing the appropriate software and package(s) depends on the individuals type of application and the algorithms that are intended to be used [9]. The most applicable for statistical inference, and for the practical component of this research is R [3]. The package in R is called 'abc' and requires the user to provide information regarding summary statistics (both observed and simulated) [3]. The ABC package works in conjunction with the abc.data package which is a real life example data set [1].

The package defines the summary statistics and calculates the distances between the corresponding observed and simulated summary statistics [3]. If the distance is lower than a given value, the corresponding parameter value

is accepted [3]. This method follows the general approach of the ABC algorithms discussed above. The area of flexibility with this package is setting the threshold value for acceptance. The user is required to provide a tolerance rate (ratio of accepted simulations) which then sets the corrected threshold value for the simulation [3]. Setting the threshold value allows the ABC package to use the MCMC and Monte-Carlo algorithms with more complex acceptance probability calculations.

The package 'abc' makes use of three ABC algorithms:

1. A rejection method [3]
2. Regression method using local linear regression [3]
3. Regression method using neural networks [3]

Rejection method:
The rejection method algorithm is called by selecting the "rejection" option [3]. This method classifies the accepted parameter values as a sample from the posterior distribution [3]. This method aligns with the workings of Algorithm 1 & 2.

Regression method:
The regression method algorithm using local linear regression or neural networks is called by selecting the "loclinear" or "neralnet" options respectively [3]. These methods correct for the variability in differences between the observed and simulated summary statistics [3]. These two methods work according to Algorithm 3 & 4 and are more thorough in the acceptance probability calculations.

4.2 Example

In order to portray the ABC method more clearly, a time series example is analyzed. Time series data is characterized by tedious likelihoods which dictate the need for ABC methods [9]. When working with a large data set such as time series data it is often beneficial to use summary statistics to reduce the burden of a large data set [9]. In this example the summary statistic illustrates the switches between two states, A and B [9]. Conclusions regarding posterior parameters can be made through the following 5 steps:

Step 1: Data is assumed to be in the form: AAAABAABBAAAAAABAAAA, which portrays a summary statistic, the number of switches between the states in the data [9].

Step 2: A uniform distribution will be assumed as the prior in the interval (0,1) [9]. Several parameters are drawn from the prior in order to build the model [9].

Step 3: The summary statistic is calculated for each combination of sequence data [9].

Step 4: In order to extract the accurate sample statistics the distance between the real and simulated data is calculated [9].

Step 5: The accurate parameters predict the posterior distribution [9].

4.3 Simulation

In order to test the ABC package first we test the algorithms on a simulated data set. The data set is simulated for three different sample sizes: n=30, n=100 and n=1000. Each data set follows a normal distribution with $\mu = 5.3$ and $\sigma^2 = 2.7$.

In regular statistical analysis there are several methods to estimate population parameters. Due to the normal distribution having a likelihood function that is workable we will compare parameter estimates calculated by different methods to analyze the accuracy of the ABC method. Estimation methods for comparison include:

1. ***Sample value:*** The mean and variance of the sample will be calculated and used as an estimate for the population parameters.
2. ***Method of moments:*** The mean and variance are calculated in terms of moments.
3. ***Bayes MCMC:*** The MCMC algorithm is tested against the theoretical methods above.

The results of each method for each sample size are summarized in Table 1. It is evident that the Bayes parameter estimates are larger than the sample and MME estimates for both parameter values. The larger the sample size gets the more accurate the sample and MME methods are at estimating the true population values however the Bayes MCMC method is most accurate in samples of size 30 and 100.

	n=30		**n=100**		**n=1000**	
Estimation method	$\hat{\mu}$	$\hat{\sigma^2}$	$\hat{\mu}$	$\hat{\sigma^2}$	$\hat{\mu}$	$\hat{\sigma^2}$
Population value	5.3	2.7	5.3	2.7	5.3	2.7
Sample value	5.12	2.34	5.18	2.72	5.23	2.77
MME	5.12	2.34	5.18	2.72	5.23	2.77
Bayes MCMC	5.27	2.89	5.12	3.12	5.779	3.79

Table 1: Simulated data: Parameter estimation

Looking at the MCMC method more closely we can see how well this method estimates the population parameters by looking at the density plots for each parameter. In figure 1 you will see the density plots for each parameter, μ and σ^2, done for all sample sizes.

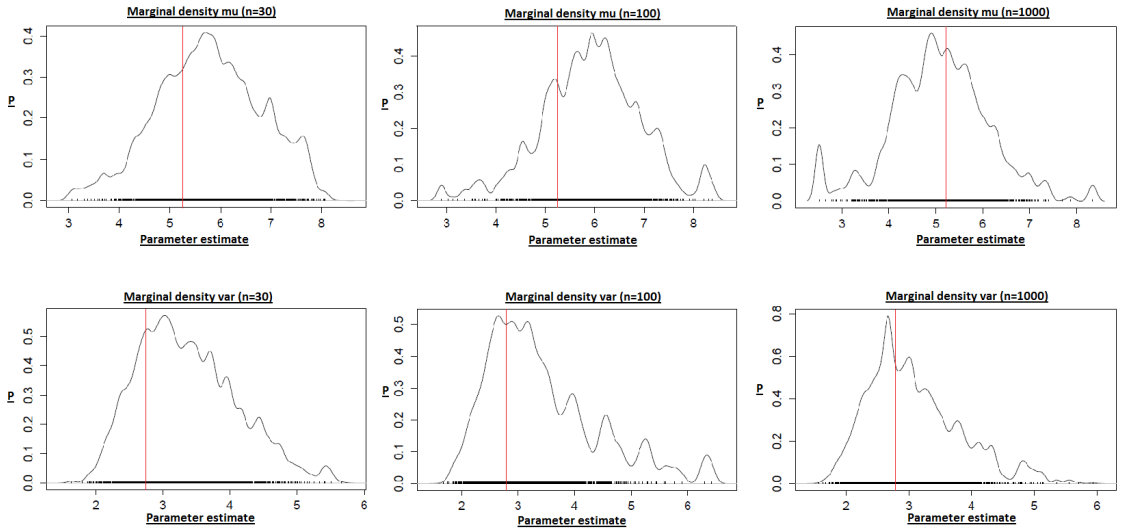


Figure 1: **Simulated data: MCMC posterior distribution**

As can be seen in figure 1, each density plot more or less follows a normal distribution. The red vertical lines indicate the true population value. As is evident by the clustering of the parameter points around the red line, the larger the sample size the more accurate the MCMC estimation is.
The graphical representation shows the overall accuracy of the MCMC method. Even though the tabulated information suggests that the MCMC method estimates more accurately for n=30 and n=100; the overall estimation for each sample value is more accurate for n=1000 due to the increased number of clustered data points around the true population value.

4.4 Real data set: Human demographic history

4.4.1 Background

The biological field is one of the more influential areas of study regarding ABC analysis. For this reason the real data set that will be analyzed is the Human demographic history data set built into the abc.data package. The human data set is a more realistic application as this data set contains real data. The goal is to estimate the human ancestral population size and differentiate between different demographic models [1]. The data set background and the parameters to be analyzed will be discussed further.

The human demographic history data set contains 50 independent autosomal non-coding regions from a Hausa (Africa), a Chinese (Asia), and an Italian (Europe) population [1]. Many population and genetic studies have discovered that African populations continue to expand while other populations have experienced fluctuations between bottleneck patterns and expansion patterns. This anomaly will be analyzed by looking at the human data set in the abc.data package and applying ABC principals on it.

The data set is comprised of three summary statistics: the average nucleotide diversity, $\bar{\Pi}$, and the mean and the variance of Tajima's D. Both Tajima's D and $\bar{\Pi}$ are used to detect historical changes in population size [1]. A negative Tajima's D represents an expansion in the population size where as a positive Tajima's D represents a population bottleneck [1]. When Tajima's D is equal to zero this indicates a constant population size. The data set is also comprised of objects: stat.voight and stat.3pops.sim which contains the simulated summary statistics. These objects can be used to predict posterior probabilities within different demographic models in Africa; Asia and Europe.

4.4.2 Demographic models

Choosing a demographic model to estimate ancestral population size depends on which model is best supported by the given data. The available models are: constant population size, exponential growth and population bottleneck [1]. All three models are characterized by several parameters including population size and rate of changes in population size [1].

4.4.3 Model selection

Before running an ABC analysis on the data we need to determine if ABC can distinguish between the three models. Figure 2 represents a confusion matrix. The confusion matrix shows how the ABC algorithm classified each model. If the models were classified correctly every time, each bar would be a different color.

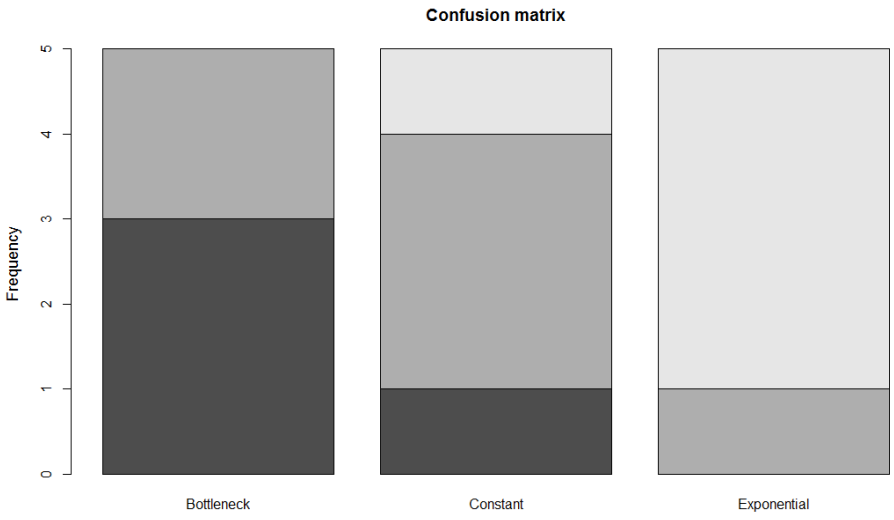


Figure 2: **Misclassification proportions for the three models**

The confusion matrix confirms that ABC can distinguish between the three demographic models. It also indicates that the exponential population expansion model is classified the most correctly with 4 correct classifications out of 5. In the following analysis we only look at the the European data set.

4.4.4 Goodness-of-fit test

Before running analysis on the European data, a goodness-of-fit test must be done to confirm the correct demographic model for the data. The European data seems to follow a bottleneck demographic model. This is confirmed

by testing this claim under a bottleneck distribution and plotting the densities in Figure 3.

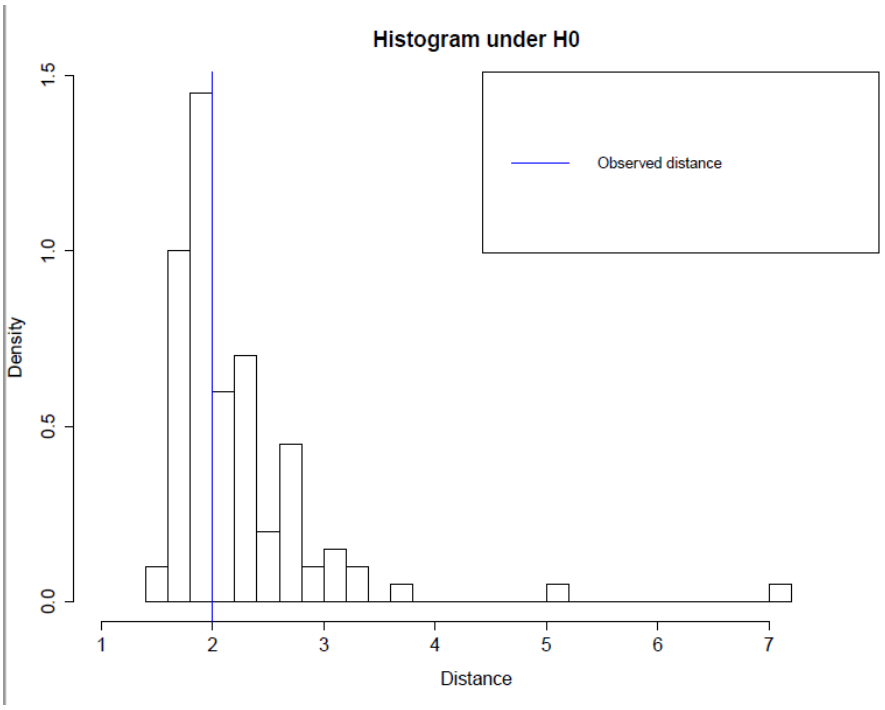


Figure 3: **Histogram of the test statistic for goodness of fit assuming a bottleneck model.**

4.4.5 Posterior predictive checks

Another test to confirm the correct demographic model for the European data is to run posterior predictive checks. Figure 4 shows the posterior distributions of the three parameters plotted using the ABC package. This is done by estimating the posterior distributions; obtaining a sample of parameters from the posterior distribution and then simulating new parameter estimates from this sample. The posterior checks use the summary statistics twice, once for sampling from the posterior distribution and once for predicting the new parameter values. This repetitive process may take long periods of time when working with larger data sets.

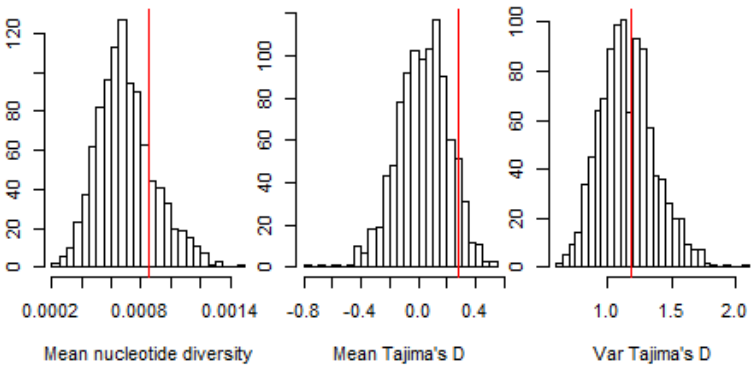


Figure 4: **Posterior predictive checks for the European data under the bottleneck model**

Figure 4 validates that the bottleneck demographic model is best suited for the European data set as the posterior distributions fit the data under the model well.

4.4.6 Cross-validation

Now that we have identified the European data set as following a bottleneck population growth pattern we can start making inferences about the ancestral population size in the European population. In order to make these inferences the data set containing the simulated summary statistics on the European population must be accessed. This data set contains: the ancestral population size (Na), the ratio of the population sizes before and during the bottleneck (a), the duration of the bottleneck (duration), and the time since the beginning of the bottleneck (start) [1].

Testing to see if ABC can estimate the parameter Na, a cross-validation test is performed. Different tolerance rates are run and the results are given in figure 5. The rejection and local linear regression methods of ABC are used to estimate the value of Na under various tolerance rates. Figure 5 compares the two methods ("rejection" and "loclinear") under three different tolerance rates [1]. Figure 5 indicates the posterior distribution medians of Na for each cross-validation sample. The points are scattered around the identity line which indicates that Na can we well estimated using the three summary statistics and ABC [1]. It is important to note that the various tolerance rates (light to dark colors indicate increasing tolerance rates) do not affect the accuracy of the estimates of Na.

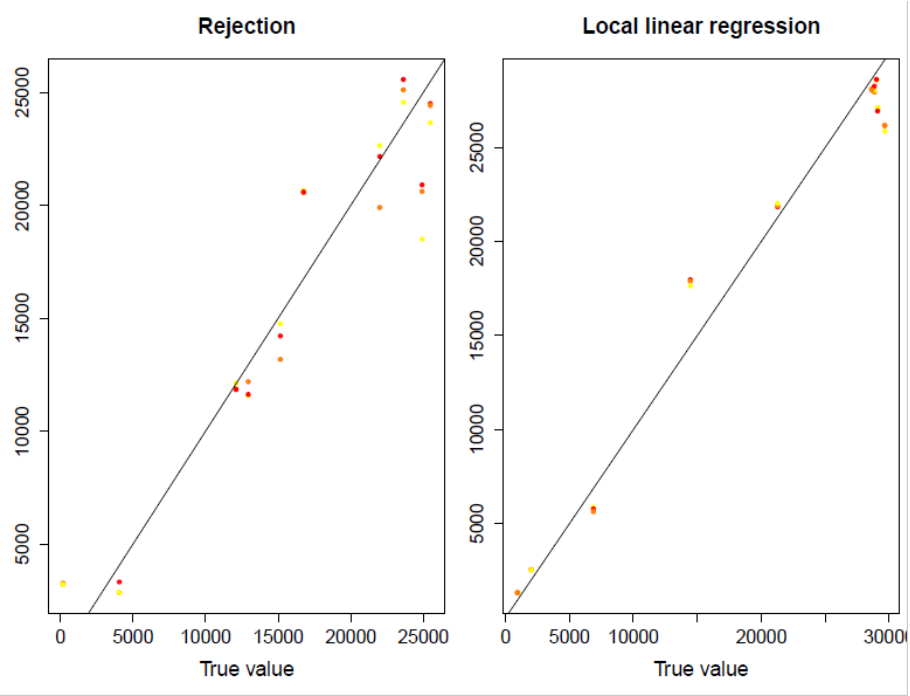


Figure 5: **Cross-validation for parameter inference.**

4.4.7 Parameter inference

Now that all the checks have been approved we can estimate the posterior distribution of Na using the ABC package. Figure 6 shows the posterior distribution of the variable, Na, from the European data set under a bottleneck demographic model.

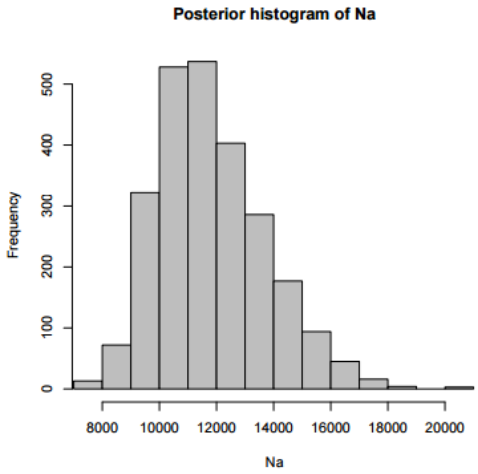


Figure 6: **Histogram of posterior sample of Na**

Figure 7 shows ABC regression diagnostics for the estimation of the posterior distribution of Na. The following three plots are generated: a density plot of the prior distribution (left), a scatter plot of the Euclidean distances as a function of the parameter values (middle) and a density plot of the posterior distribution (right). The apparent difference in the prior and posterior distribution plots convey that the three summary statistics convey information about the ancestral population size. The middle panel of Figure 7 shows the distance between the simulated and observed summary statistics as a function of the prior values of Na [1]. This confirms once again that the summary statistics convey information about Na since the distances for the accepted values (red) are clustered together [1].

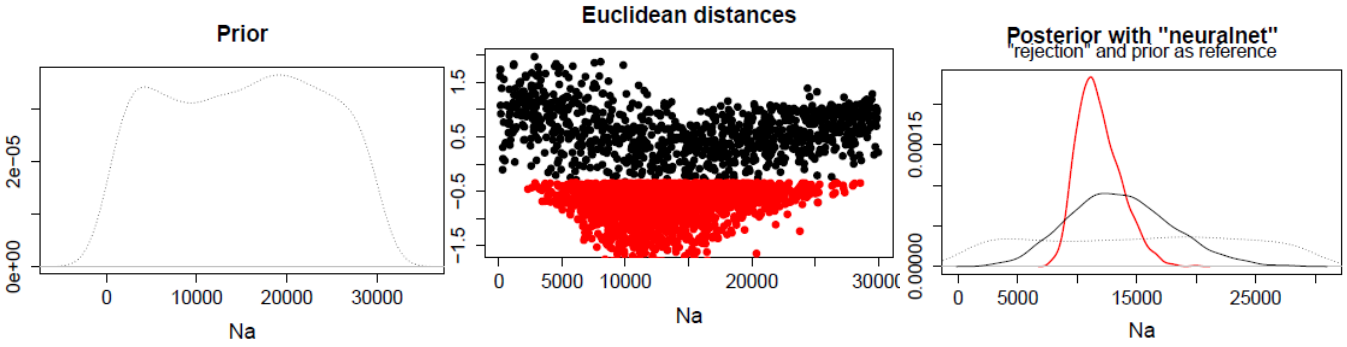


Figure 7: **ABC regression diagnostics for the estimation of the posterior distribution of Na**

4.5 Model comparison and application

ABC methods be used as a tool for estimating parameters for deriving the posterior probabilities of various models [9]. These estimations have numerous practical uses in varying industries. Biologically related areas are one of the more prominent sectors that ABC has influenced. Molecular data can now be analyzed and statistical methods are providing more insight into fields such as natural populations and evolutionary genetics [2]. The Markov chain Monte Carlo technique is the more commonly used method in this area of study [2]. Owing to the exponentially growing size of data within DNA advancements, ABC methods are used to bypass likelihood calculations to improve efficiency [2].

Another sector which largely benefits from ABC methods is the insurance sector. Bayesian inference is where actuarial credibility theory began so the ABC application in the insurance sector is valid and well established. Other areas of use of ABC methods, according to Worldwide Science include:

1. Functional statistics
2. Goodness-of-fit statistics

3. Population genetics
4. Diffusion filtration
5. Forward modeling in cosmology

4.6 Improvements

Although ABC has enabled previously unexplainable models to be evaluated it still contains problem areas and room for improvements. The risk of using ABC methods will be highlighted here with a focus on the areas where ABC does not operate well.

Prior distribution and parameter ranges [9]:

In ABC methods the prior distribution and parameter range must be specified. In some cases these values have been estimated by the user which bring in a space of error and biasedness. Although in some cases it is possible to estimate these values in accordance with the known properties, there are cases where these estimates are unattainable. Therefore, in cases where these values are to be estimated by the user, an element of error is introduced to the ABC method.

Small number of models [9]:

Model-based studies operate within a small model space. There have been several criticisms on the small number of models studied in the model-based studies. The risk taken here is that the small number of models gives a limited insight which may cause the conclusions to be biased.

Large data sets [9] :

On the other hand, large data sets can also bring with it some problems. For model-based studies, large data sets may cause a computational bottleneck. This may cause parts of a data set to be excluded from the analysis which introduces another element of biasedness.

Curse of dimensionality [9] :

High dimension parameter sets and data sets demand high numbers of parameter simulations. This simply increases the operating costs and in severe cases my make the analysis intractable.

Although there are many pitfalls with the ABC method, the overall advantages of using a method like this are immense.

5 Conclusion

In conclusion, ABC consists of a group of efficient methods for statistical inference. However, when applying ABC methods additional caution must be taken owing to the level of biasedness introduced with the approximations [9]. At this point in time ABC is very well suited for problems that involve individual parameter inference. In order to practically use ABC in problems with a multitude of parameters more work and adaptation to the current ABC methods is required. The ease of using ABC to bypass complex likelihoods should not be blindsided by the fact that these complex likelihoods may not allow for accurate prediction by ABC methods.

6 Appendix

6.1 Simulation

6.1.1 Table 1

```
#TITLE: MME and MLE for n=30;100;1000

data_30 =  rnorm(30, mean =5.3,sd = 2.7)
MMEm_30=mean(data_30)
frame_30=data.frame(data_30, MMEm_30,data_30-MMEm_30, (data_30-MMEm_30)^2)
MMEv_30=sum(frame_30$X.data_30...MMEm_30..2)/30
MEAN_30=mean(data_30)
V_30=sd(data_30)^2

data_100 =  rnorm(100, mean =5.3,sd = 2.7)
MMEm_100=mean(data_100)
frame_100=data.frame(data_100, MMEm_100,data_100-MMEm_100, (data_100-MMEm_100)^2)
MMEv_100=sum(frame_100$X.data_100...MMEm_100..2)/100
MEAN_100=mean(data_100)
V_100=sd(data_100)^2

data_1000 =  rnorm(1000, mean =5.3,sd = 2.7)
MMEm_1000=mean(data_1000)
frame_1000=data.frame(data_1000, MMEm_1000,data_1000-MMEm_1000, (data_1000-MMEm_1000)^2)
MMEv_1000=sum(frame_1000$X.data_1000...MMEm_1000..2)/1000
MEAN_1000=mean(data_1000)
V_1000=sd(data_1000)^2
```

6.1.2 Figure 1

```
# TITLE: A simple Approximate Bayesian Computation MCMC (ABC-MCMC)

library(coda)

# assuming the data are 10 samples of a normal distribution
# with mean 0 and sd 1
mcmc_30=data.frame(data_30)
mcmc_100=data.frame(data_100)
mcmc_1000=data.frame(data_1000)

# we want to use ABC to infer the parameters that were used.
# we sample from the same model and use mean and variance
# as summary statstitics. We return true for ABC acceptance when
# the difference to the data is smaller than a certain threshold

mean_30 <- mean(data_30)
sd_30 <- sd(data_30)
mean_100 <- mean(data_100)
sd_100 <- sd(data_100)
mean_1000 <- mean(data_1000)
sd_1000 <- sd(data_1000)


ABC_acceptance <- function(par,meandata,standarddeviationdata){

  # prior to avoid negative standard deviation
  if (par[2] <= 0) return(F)
```

```
  # stochastic model generates a sample for given par
  samples <- rnorm(10, mean =par[1], sd = par[2])

  # comparison with the observed summary statistics
  diffmean <- abs(mean(samples) - meandata)
  diffsd <- abs(sd(samples) - standarddeviationdata)
  if((diffmean < 0.1) & (diffsd < 0.2)) return(T) else return(F)
}


# we plug this in in a standard metropolis hastings MCMC,
# with the metropolis acceptance exchanged for the ABC acceptance

run_MCMC_ABC <- function(startvalue, iterations, meandata,standarddeviationdata){

  chain = array(dim = c(iterations+1,2))
  chain[1,] = startvalue

  for (i in 1:iterations){

    # proposalfunction
    proposal = rnorm(2,mean = chain[i,], sd= c(0.7,0.7))

    if(ABC_acceptance(proposal,meandata,standarddeviationdata)){
      chain[i+1,] = proposal
    }else{
      chain[i+1,] = chain[i,]
    }
  }
  return(mcmc(chain))
}

posterior_30 <- run_MCMC_ABC(c(4,2.3),300000,mean_30,sd_30)
plot(posterior_30)
summary(posterior_30)

posterior_100 <- run_MCMC_ABC(c(4,2.3),300000,mean_100,sd_100)
plot(posterior_100)
summary(posterior_100)

posterior_1000 <- run_MCMC_ABC(c(4,2.3),300000,mean_1000,sd_1000)
plot(posterior_1000)
summary(posterior_1000)
```

6.2 Real data set: Human demographic history

6.2.1 Figure 2

```
require(abc.data)
data(human)
stat.voight

#Demographic models
par(mfcol = c(1,3), mar=c(5,3,4,.5))

#Model selection
cv.modsel <- cv4postpr(models, stat.3pops.sim, nval=5, tol=.01, method="mnlogistic")
```

```
s <- summary(cv.modsel)

plot(cv.modsel, names.arg=c("Bottleneck", "Constant", "Exponential"))
```

6.2.2 Figure 3

```
modsel.ha<-postpr(stat.voight["hausa",], models, stat.3pops.sim, tol=.05, method="mnlogistic")
modsel.it<-postpr(stat.voight["italian",], models, stat.3pops.sim, tol=.05, method="mnlogistic")
modsel.ch<-postpr(stat.voight["chinese",], models, stat.3pops.sim, tol=.05, method="mnlogistic")
summary(modsel.ha)
summary(modsel.it)
summary(modsel.ch)

#Goodness of fit
res.gfit.bott=gfit(target=stat.voight["italian",], sumstat=stat.3pops.sim[models=="bott",],
                   statistic=mean, nb.replicate=100)
plot(res.gfit.bott, main="Histogram under H0")
```

6.2.3 Figure 4

```
#Posterior predictive checks

require(abc.data)
data(ppc)
mylabels <- c("Mean nucleotide diversity","Mean Tajima's D", "Var Tajima's D")
par(mfrow = c(1,3), mar=c(5,2,4,0))
for (i in c(1:3)){
  hist(post.bott[,i], breaks=40, xlab=mylabels[i], main="")
  abline(v = stat.voight["italian", i], col = 2)
}
```

6.2.4 Figure 5

```
#Cross validation

stat.italy.sim <- subset(stat.3pops.sim, subset=models=="bott")
head(stat.italy.sim)
head(par.italy.sim)

cv.res.rej <- cv4abc(data.frame(Na=par.italy.sim[,"Ne"]), stat.italy.sim, nval=10,
                     tols=c(.005,.01, 0.05), method="rejection")
cv.res.reg <- cv4abc(data.frame(Na=par.italy.sim[,"Ne"]), stat.italy.sim, nval=10,
                       tols=c(.005,.01, 0.05), method="loclinear")
summary(cv.res.rej)
summary(cv.res.reg)

par(mfrow=c(1,2), mar=c(5,3,4,.5), cex=.8)
plot(cv.res.rej, caption="Rejection")
plot(cv.res.reg, caption="Local linear regression")
```

6.2.5 Figure 6

```
res <- abc(target=stat.voight["italian",], param=data.frame(Na=par.italy.sim[, "Ne"]),
           sumstat=stat.italy.sim, tol=0.05, transf=c("log"), method="neuralnet")
hist(res)
```

6.2.6 Figure 7

```
par(cex=.8)
plot(res, param=par.italy.sim[, "Ne"])
```

References

[1] K Csilléry, L Lemaire, O François, and MGB Blum. Approximate Bayesian computation (ABC) in R: A vignette. *Journal of the American statistical association*, May 2015.

[2] Katalin Csilléry, Michael GB Blum, Oscar E Gaggiotti, and Olivier François. Approximate Bayesian computation (ABC) in practice. *Trends in ecology & evolution*, 25(7):410–418, 2010.

[3] Katalin Csilléry, Olivier François, and Michael GB Blum. ABC: an R package for approximate Bayesian computation (ABC). *Methods in ecology and evolution*, 3(3):475–479, 2012.

[4] Haipeng Guo and William Hsu. A survey of algorithms for real-time Bayesian network inference. *Joint Workshop on Real-Time Decision Support and Diagnosis Systems*, 2002.

[5] Deukwoo Kwon and Isildinha M Reis. Simulation-based estimation of mean and standard deviation for meta-analysis via approximate Bayesian computation (ABC). *BMC medical research methodology*, 15(1):1, 2015.

[6] Jean-Michel Marin, Pierre Pudlo, Christian P Robert, and Robin J Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 2012.

[7] Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, 2003.

[8] Jonathan K Pritchard, Mark T Seielstad, Anna Perez-Lezaun, and Marcus W Feldman. Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798, 1999.

[9] Mikael Sunnåker, Alberto Giovanni Busetto, Elina Numminen, Jukka Corander, Matthieu Foll, and Christopher Dessimoz. Approximate Bayesian computation. *PLoS Comput Biol*, 2013.

[10] Simon Tavaré, David J Balding, Robert C Griffiths, and Peter Donnelly. Inferring coalescence times from dna sequence data. *Genetics*, 145(2):505–518, 1997.

[11] Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael PH Stumpf. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009.

Semi-supervised machine learning for textual anomaly detection

Carl Steyn 11009757

WST795 Research Report

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

Supervisor: Dr A de Waal, Co-supervisor: J Mazarura

Department of Statistics, University of Pretoria

November 2, 2016

Abstract

Anomaly detection comprises the identification of observations which do not follow the expected patterns of the assumed data set. We attempt to simplify the problem of textual anomaly detection by constructing a Multinomial Naïve Bayes classifier and enhancing it with an augmented Expectation Maximization (EM) algorithm. By doing so, we utilize large amounts of unlabelled data and show how the EM algorithm could increase the accuracy of the Naïve Bayes classifier. The process is applied to a binary classification environment in order to detect anomalies in text.

Declaration

I, *Carl Frederick Steyn,* 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.

_ _
Carl Frederick Steyn

_ _
A de Waal

_ _
J Mazarura

_ _
Date

Contents

1 Introduction **5**

2 Background Theory **6**
2.1 Literature review 6
2.2 Important assumptions 6
2.3 Model parameters 6
2.4 Parameter estimation 7
2.5 Expectation maximization 9
2.6 Augmented EM 10

3 Experimental design **11**
3.1 20 Newsgroups data 11
3.2 The log-sum-exp trick 11

4 Practical application **12**
4.1 Naïve Bayes and EM with 10 classes 12
4.2 Anomaly Detection with Naïve Bayes and EM 12

5 Conclusion **14**

Appendix: Python Code **16**

List of Figures

1 EM convergence on 10-class data set 13
2 ROC curve for augmented binary classifier on 20 Newsgroups data 13

1 Introduction

In many research areas there exists a requirement to find patterns or trends in data which do not conform to a certain behaviour [1]. Examples of such instances include finding abnormal traffic patterns in a computer network system to detect suspicious data transfers, identifying fraudulent activity in a banking environment or analyzing streamed twitter data to identify important events.

In a text-analysis problem, consider for example a company that collects text data in order to analyze the success of a marketing campaign for their client, the financial service group Liberty. The company will require a large corpus containing documents with specific keywords such as their client's name, "Liberty". However, the word "liberty" is used in many contexts and is therefore not only associated with the client. Consequently, the company's large corpus possibly contains irrelevant documents that need to be identified and filtered out.

Anomaly detection relates to finding a solution to the above problems, for the specific domain at hand. The general problem faced in anomaly detection is that the performance of each algorithm or method largely depends on the domain it is applied to. One increasingly popular research area relevant to anomaly detection is text analysis. Web-pages, news articles, social media etc. are examples of instances where text is used to convey useful information.

Text analysis is a challenging field because in many instances it requires both quantitative and qualitative reasoning. To clarify, finding the number of times a certain word is used in a document falls under quantitative reasoning whereas finding the semantic (linguistic) content in the text data relates to qualitative reasoning. However, despite the complexity of properly using text analysis or finding anomalies in text, much benefit is to be gained from it. Anomalies in text are almost always domain-specific phenomena and usually requires sufficient domain knowledge to detect. For example, when reading nursery rhymes for children the phrase "gross domestic product" would appear anomalous to an expert with domain knowledge on the subject. Therefore, the problem of defining a clear distinction between normal and abnormal depends on the domain and usually requires human assistance.

In this paper we start by constructing a Naïve Bayes text classifier. In order to increase the performance of our classifier, we use the augmented Expectation Maximization (EM) algorithm which has been shown in previous work to perform well with Naïve Bayes [2][9]. In order to implement EM later on, we will need access the parameter estimates of the model. Therefore, we need to construct the classifier from first principles in order to preserve important information such as the word-class matrix and the class-prior probabilities. This process shall be covered extensively in the "Background Theory" section. We then attempt to improve the accuracy of the Naïve Bayes classifier by using an augmented form of EM[9].

To demonstrate the performance of the augmented classifier, we apply it to the well-known 20 Newsgroups data set [5]. Since the ultimate objective of this research is to identify and eliminate anomalous/irrelevant text documents from a corpus, we change the classifier into a binary classifier. Lastly, we sample from the 20 Newsgroup data in order to create a dataset emulating anomalous items in text data.

2 Background Theory

2.1 Literature review

Chandola et al. provide an overview of the research done on anomaly detection. In their paper [1] they group existing techniques into different categories (domains) and structurize the key assumptions made in the different domains to define "normal" regions. Mahapatra et al. [7] provide insight into using linguistic (semantic) content of text data for anomaly detection and how it could reduce the number of false positives. The text data they use is of a more literary nature where they focus on syntax and reading difficulty. In response, Kumarashwami et al. [6] demonstrate that domain-specific feature selection is more important than linguistic features for anomaly detection in text data.

In his paper [9], Nigam used Expectation-Maximization (EM) to utilize large and inexpensive unlabeled text data sets in order to improve his supervised learning algorithm. What we aim to achieve in this paper is similar. We will deploy a hybrid procedure where we use unlabeled and labeled data together to increase the accuracy of a text classifier by using Nigam's EM method and doing anomaly detection by binary classification.

2.2 Important assumptions

In order to derive statistical characteristics from text data, we must assume that the data originates from some generative model. This assumption creates a framework under which we make another two important assumptions:

1. The data is produced from a mixture model.
2. There is a one-to-one relationship between a class and a mixture component from said mixture model [9].

Our goal is to classify documents by using a Naïve Bayes Classifier. With this approach we make one more simplifying assumption: all words within a document are independent of each other given a class. The classifier is called Naïve because we do not actually expect this assumption to be true in practice, but it has been proved [3] that the Naïve Bayes Classifier works well despite having violated its assumptions. This is because the reduction in parameters due to the word-independence assumption makes the model more immune to over-fitting [8].

2.3 Model parameters

Let us define $C = \{c_1, c_2, ..., c_{|C|}\}$ as the collection of all mixture components (classes) in our model where $|C|$ is the total amount of mixture components. Then under the above framework we assume that every document in the corpus has been generated with a certain mixture component $c_j \in C$ and a corresponding set of parameters $\boldsymbol{\theta}$. The probability of some document d_i being generated by a specific mixture component can be expressed as follows:

$$P(c_j|\boldsymbol{\theta})P(d_i|c_j;\boldsymbol{\theta}), \tag{1}$$

where $d_i \in D = \{d_1, ..., d_{|D|}\}$, $|D|$ being the total amount of documents in the corpus. Therefore, the probability of a document being generated (by any mixture component) is the sum of total probability over all mixture components:

$$P(d_i|\boldsymbol{\theta}) = \sum_{j=1}^{|C|} P(c_j|\boldsymbol{\theta})P(d_i|c_j;\boldsymbol{\theta}). \tag{2}$$

Let V be the vocabulary such that $V =< w_1, w_2, ..., w_{|V|} >$ where $|V|$ is the total amount of unique words in the vocabulary and each w_i represents one unique word. Each document d_i can be expressed as an ordered

list of words $< w_{d_i,1}, w_{d_i,2}, ..., w_{d_i,|d_i|} >$ where $w_{d_i,j}$ is the jth word in the document and $|d_i|$ is the total amount of words in the document. In this model we assume that the length of a document is independent of the mixture component that generated it [9]. Keeping the word-independence assumption in mind, we can now expand the second factor from (1) by writing the probability of a document given a class in terms of its document length and word probabilities:

$$P(d_i|c_j;\boldsymbol{\theta}) = P(< w_{d_i,1}, w_{d_i,2}, ..., w_{d_i,|d_i|} > |c_j;\boldsymbol{\theta}) = P(|d_i|)\prod_{k=1}^{|d_i|} P(w_{d_i,k}|c_j;\boldsymbol{\theta}). \tag{3}$$

We now introduce the parameters used in our model, $\boldsymbol{\theta}$. Intuitively, from (3) above we see that the parameters of a specific mixture component c_j is a set of word probabilities. We use the notation $\theta_{w_t|c_j} = P(w_t|c_j;\boldsymbol{\theta})$ for a single word probability given a mixture component c_j.
It is important to note that

$$\sum_{t=1}^{|V|} P(w_t|c_j;\boldsymbol{\theta}) = 1.$$

Furthermore, we assume that document length is identically and independently distributed and therefore does not need to be parameterized [9]. The only remaining set of parameters used in the model is called the class prior distribution, $\theta_{c_j} = P(c_j|\boldsymbol{\theta})$. Therefore, the set of parameters in our model is formally defined as follows:

$$\boldsymbol{\theta} = \{\theta_{w_t|c_j} : w_t \in V, c_j \in C; \theta_{c_j} : c_j \in C\}. \tag{4}$$

2.4 Parameter estimation

The first step in building the classifier is finding the estimates of the parameters that maximize the probability $P(\boldsymbol{\theta}; D)$. This method is called Maximum a Posteriori (MAP) estimation. In order to classify a document, we choose the class c_j that maximizes the probability $P(y_i = c_j|d_i;\hat{\boldsymbol{\theta}})$.

The above probability can be extended by using Bayes' rule:

$$P(y_i = c_j|d_i;\hat{\boldsymbol{\theta}}) = \frac{P(c_j|\hat{\boldsymbol{\theta}})P(d_i|c_j;\hat{\boldsymbol{\theta}})}{P(d_i|\hat{\boldsymbol{\theta}})}.$$

By substituting equations (2) and (3) into the above formula, we obtain the following result:

$$= \frac{P(c_j|\hat{\boldsymbol{\theta}})\prod_{k=1}^{|d_i|} P(w_{d_i,k}|c_j;\hat{\boldsymbol{\theta}})}{\sum_{h=1}^{|C|} P(c_h|\hat{\boldsymbol{\theta}})\prod_{k=1}^{|d_i|} P(w_{d_i,k}|c_h;\hat{\boldsymbol{\theta}})}. \tag{5}$$

We calculate the parameter estimates $P(c_j|\hat{\boldsymbol{\theta}})$ and $P(w_{d_i,k}|c_j;\hat{\boldsymbol{\theta}})$ by using observed frequencies in the training data. The frequencies are represented by the following notation:

- The label indicator function $z_{ij} = \begin{cases} 1 & \text{if document i has mixture component j} \\ 0 & \text{otherwise} \end{cases}$.
- The word counting function $N(w_t, d_i) =$ the number of times word w_t appears in document d_i.

It follows that:

$$P(c_j|\hat{\boldsymbol{\theta}}) = \frac{\text{Number of documents with mixture component } c_j}{\text{Number of documents in the training set}}$$

$$= \frac{\sum_{i=1}^{|D|} z_{ij}}{\sum_{j=1}^{|C|}\sum_{i=1}^{|D|} z_{ij}}$$

$$= \frac{\sum_{i=1}^{|D|} z_{ij}}{|D|}. \tag{6}$$

$$P(w_{d_i,k}|c_j;\hat{\boldsymbol{\theta}}) = \frac{\text{Number of times the word } w_{d_i,k} \text{ occurs in mixture component } c_j}{\text{Number of words in mixture component } c_j}$$

$$= \frac{\sum_{i=1}^{|D|} N(w_t, d_i) z_{ij}}{\sum_{k=1}^{|V|}\sum_{i=1}^{|D|} N(w_k, d_i) z_{ij}}. \tag{7}$$

A common problem faced when calculating equations (6) and (7) are zero-probabilities. This could occur if a certain word never appears in a class. Zero probabilities can cause significant damage to the accuracy of this model since any other evidence in the same product as the zero probability is discarded. A simple way to prevent this is to use Laplace-smoothing [9], i.e. we add extra counts in the numerator and denominator as follows:

$$P(c_j|\hat{\boldsymbol{\theta}}) = \frac{1 + \sum_{i=1}^{|D|} z_{ij}}{\sum_{j=1}^{|C|}\left(1 + \sum_{i=1}^{|D|} z_{ij}\right)}$$

$$= \frac{1 + \sum_{i=1}^{|D|} z_{ij}}{|C| + |D|}. \tag{8}$$

$$P(w_{d_i,k}|c_j;\hat{\boldsymbol{\theta}}) = \frac{1 + \sum_{i=1}^{|D|} N(w_t, d_i) z_{ij}}{\sum_{k=1}^{|V|}\left(1 + \sum_{i=1}^{|D|} N(w_k, d_i) z_{ij}\right)}$$

$$= \frac{1 + \sum_{i=1}^{|D|} N(w_t, d_i) z_{ij}}{|V| + \sum_{k=1}^{|V|}\sum_{i=1}^{|D|} N(w_k, d_i) z_{ij}}. \tag{9}$$

Results (8) and (9) can now be substituted into result (5) to calculate the probability of a label given a specific document. Finally, the Naïve Bayes classifier returns the value $argmax_{c_j \in C}\left\{P(y_i = c_j|d_i;\hat{\boldsymbol{\theta}})\right\}$, which is the mixture component/class label associated with the highest probability.

2.5 Expectation maximization

Expectation Maximization (EM) is an iterative algorithm used to estimate parameters when dealing with incomplete data. By iterating over (E) and (M) steps, the algorithm converges by maximizing the complete log-likelihood of the model.

For the rest of the paper we shall use a training set that consists of some labelled documents and large amounts of unlabelled data. Define $D = \left\{D^l, D^u\right\}$ as the collection of all our training data. Then D consists of two partitions, namely D^l which contains all the labelled data and D^u which contains the unlabelled data. Due to the fact that we have missing labels in our training set and therefore have latent parameters in our model, our data is incomplete. Although this might seem problematic, our aim is to draw information from the large set of unlabelled data by using EM which accounts for incomplete data.

The log-likelihood of this model can be expressed as follows:

$$l(\boldsymbol{\theta}|D) = log(P(\boldsymbol{\theta})) + \sum_{d_i \in D^u} log \sum_{j=1}^{|C|} P(c_j|\boldsymbol{\theta}) P(d_i|c_j;\boldsymbol{\theta}) + \sum_{d_i \in D^l} log(P(y_i = c_j|\boldsymbol{\theta}) P(d_i|y_i = c_j;\boldsymbol{\theta})). \tag{10}$$

It would be inefficient to attempt to maximize the above expression since it contains a log of sums. However, EM provides an alternative approach.

Suppose that we knew the class labels for all the documents. We could then construct a matrix of binary variables $Z = \left(z_1, ..., z_{|D|}\right)$ such that $z_i = \left\langle z_{i1}, ..., z_{i|C|} \right\rangle$ and z_{ij} remains defined as in section 2.3. It follows that we can express the complete log-likelihood as follows:

$$l_c(\boldsymbol{\theta}|D; z) = log(P(\boldsymbol{\theta})) + \sum_{d_i \in D} \sum_{j=1}^{|C|} z_{ij} log(P(c_j|\boldsymbol{\theta}) P(d_i|c_j;\boldsymbol{\theta})). \tag{11}$$

The expected value of z_{ij} when $i \in D^u$ is equal to the probabilistic labels given by equation (5). By replacing the $z_{ij}^{'s}$ in equation (11) with their expected values we can find a lower bound for equation (10) with each iteration [9].

To summarize, the EM iteration process is explained in Algorithm 1:

Algorithm 1 Expectation Maximization

1. As an initial (M)-step, using equations (8) and (9) we determine the MAP estimates, $\hat{\theta}$, of the Naïve Bayes model given the labelled training data.
2. (E)-step: Determine the expected value of z given the current parameter estimates, $\hat{\boldsymbol{\theta}}$.
3. (M)-step: Using the updated expected value of z, determine new parameter estimates using equations (8) and (9).

Steps 2 and 3 are repeated until convergence, i.e. until the log-likelihood reaches a local maximum and cannot increase any further.

2.6 Augmented EM

We have mentioned in sections 2.2 and 2.3 that certain assumptions need to be made about the nature of the data in this environment in order to fit a model. As suggested in the name, Naïve Bayes tends to perform well in practice despite some minor violations of the assumptions. By combining labelled and unlabelled data to train our model, we risk violating our assumptions to a greater extent and damaging the performance of our model more than improving it.

Nigam [9] addresses this issue by introducing two separate strategies. The first strategy is to multiply the parameter estimates obtained from the unlabelled data with a scalar $\lambda \in [0, 1]$ in the MAP estimator function. Intuitively, using a λ-value of 1 would be equal to standard EM as in the previous section. EM brings unsupervised clustering to the classification process, which is why our algorithm is semi-supervised. The λ scalar controls the intensity with which EM performs unsupervised clustering in our model [9].

The second strategy is to change our assumption from section 2.2 regarding the one-to-one relationship between a mixture component and a class label. This is a strong assumption which, if violated, could cause severe performance issues in the model. The assumption can be manipulated to be less strict by assuming a many-to-one relationship between mixture components and class labels instead. Therefore, instead of pairing one label y_i with one mixture component c_j, the strategy suggests that we model the possibility of multiple mixture components for any class.

We focus on strategy 1 in this paper, i.e. scaling down the effect of the unlabelled data on our model parameters with a value we define as:

$$\Lambda = \begin{cases} 1 & \text{if} \quad d_i \in D^l \\ \lambda & \text{if} \quad d_i \in D^u \end{cases} .$$

Equation (10) illustrates how the log-likelihood consists of log-probabilities from both labelled and unlabelled data. To scale down the unlabelled data, we rewrite the complete log-likelihood in a similar fashion and multiply the unlabelled data with λ:

$$l_c(\boldsymbol{\theta}|D; z) = log(P(\boldsymbol{\theta})) + \sum_{d_i \in D^l} \sum_{j=1}^{|C|} z_{ij} log(P(c_j|\boldsymbol{\theta}) P(d_i|c_j; \boldsymbol{\theta}))$$
$$+\lambda \Big(\sum_{d_i \in D^u} \sum_{j=1}^{|C|} z_{ij} log(P(c_j|\boldsymbol{\theta}) P(d_i|c_j; \boldsymbol{\theta})) \Big). \tag{12}$$

We also need to scale down the counts from unlabelled data when estimating the parameters in equations (6) and (7). Equations (13) and (14) are therefore our new parameter estimates under the augmented EM algorithm:

$$P(c_j|\hat{\boldsymbol{\theta}}) = \frac{1 + \sum_{i=1}^{|D|} \Lambda_i P(y_i = c_j|d_i; \hat{\boldsymbol{\theta}})}{|C| + \lambda |D^u| + |D^l|}, \tag{13}$$

$$P(w_{d_i,k}|c_j; \hat{\boldsymbol{\theta}}) = \frac{1 + \sum_{i=1}^{|D|} \Lambda_i N(w_t, d_i) I(c_j, d_i)}{|V| + \sum_{k=1}^{|V|} \sum_{i=1}^{|D|} \Lambda_i N(w_k, d_i) I(c_j, d_i)}. \tag{14}$$

Algorithm 1 in section 2.5 remains unchanged except for the fact that we now use equation (12) for the log-likelihood and equations (13) and (14) to find the MAP estimates for the model parameters.

3 Experimental design

3.1 20 Newsgroups data

We use the 20 Newsgroups data to test our algorithm. This data set is a collection of 20000 news-related articles and emails spread across 20 different topics. In order to use text documents for classification, some preprocessing is required. This includes removing unimportant words, also known as stop words. Once we have cleaned the data, we tokenize the text data by separating each word in a document and changing them to lower case. This enables the next step, which is to represent the text data with numbers. This is achieved by constructing a vector called a vocabulary, which consists of an ordered list containing each unique word in the corpus. Every document can then be vectorized by expressing the document as a collection of words from the vocabulary and their corresponding amounts of occurrences, or counts. We used Scikit-Learn's CountVectorizer [10] package to vectorize the data.

3.2 The log-sum-exp trick

Both the numerator and denominator in equation (5) contain the product over many small word-probabilities. For a large enough vocabulary, the resulting product over all the word-probabilities becomes smaller than the computer can store in memory. This occurrence is known as arithmetic underflow. A solution to this problem is to take the log of the formula, which makes the calculation considerably easier to work with on a computer. We proceed as follows:

$$log\left(\frac{P(c_j|\hat{\boldsymbol{\theta}})\prod_{k=1}^{|d_i|}P(w_{d_i,k}|c_j;\hat{\boldsymbol{\theta}})}{\sum_{h=1}^{|C|}P(c_h|\hat{\theta})\prod_{k=1}^{|d_i|}P(w_{d_i,k}|c_h;\hat{\theta})}\right)$$

$$= logP(c_j|\hat{\boldsymbol{\theta}}) + \sum_{k=1}^{|d_i|} logP(w_{d_i,k}|c_j;\hat{\boldsymbol{\theta}}) - log\left[\sum_{h=1}^{|C|}\left(P(c_h|\hat{\boldsymbol{\theta}})\prod_{k=1}^{|d_i|}P(w_{d_i,k}|c_h;\hat{\boldsymbol{\theta}})\right)\right]. \tag{15}$$

Here a new problem presents itself. The log over a product is easy calculable, however the log over a sum is not. In order to calculate the log over a sum, we use the log-sum-exp trick.
Consider the last term from equation (15):

$$log\left[\sum_{h=1}^{|C|}\left(P(c_h|\hat{\boldsymbol{\theta}})\prod_{k=1}^{|d_i|}P(w_{d_i,k}|c_h;\hat{\boldsymbol{\theta}})\right)\right].$$

We start off by taking the logs and then the exponential of the values inside the summation to get:

$$= log\left[\sum_{h=1}^{|C|} exp\left(logP(c_h|\hat{\boldsymbol{\theta}}) + \sum_{k=1}^{|d_i|} logP(w_{d_i,k}|c_h;\hat{\boldsymbol{\theta}})\right)\right].$$

Now, let b be the largest log-probability in the above expression, i.e.

$$b = max\left\{logP(c_h|\hat{\boldsymbol{\theta}}), logP(w_{d_i,1}|c_h;\hat{\boldsymbol{\theta}}), ..., logP(w_{d_i,|d_i|}|c_h;\hat{\boldsymbol{\theta}})\right\}.$$

By factoring out the largest log-probability b, we then get:

$$= log\left[e^b \sum_{h=1}^{|C|} exp\left(logP(c_h|\hat{\boldsymbol{\theta}}) + \sum_{k=1}^{|d_i|} logP(w_{d_i,k}|c_h;\hat{\boldsymbol{\theta}}) - b\right)\right]$$

$$= b + log\left[\sum_{h=1}^{|C|} exp\left(logP(c_h|\hat{\boldsymbol{\theta}}) + \sum_{k=1}^{|d_i|} logP(w_{d_i,k}|c_h;\hat{\boldsymbol{\theta}}) - b\right)\right].$$

By doing this, we have scaled down the magnitude of the above values which, being the logs of many small probabilities, had large absolute values. This expression is a more accurate approximation to the log of a sum since only the maximum value in the summation may cause underflow instead of several such occurrences. The log of equation (5) can now be approximated with higher accuracy. Finally, equation (15) can be expressed as:

$$logP(c_j|\hat{\boldsymbol{\theta}}) + \sum_{k=1}^{|d_i|} logP(w_{d_i,k}|c_j;\hat{\boldsymbol{\theta}}) - b - log\left[\sum exp\left(logP(c_h|\hat{\boldsymbol{\theta}}) + \sum_{k=1}^{|d_i|} logP(w_{d_i,k}|c_h;\hat{\boldsymbol{\theta}}) - b\right)\right]. \quad (16)$$

4 Practical application

4.1 Naïve Bayes and EM with 10 classes

The training data set consists of 300 labelled documents spread across 10 classes. The data has been structured into 30 documents per class to maintain class balance. We first estimate the model parameters using equations (8) and (9) and the set of training data. We then calculate probabilistic labels for 1000 unlabelled documents by using our parameter estimates.

The EM iteration is then initialized by recalculating the parameter estimates by using the labelled data together with the unlabelled data (equations (13) and (14)), and using a λ-value of 0.6 to scale down the unsupervised effect of the unlabelled data on the model. Figure 1 shows the convergence of the log-likelihood during the EM iteration process. The ordinary Naïve Bayes scored an accuracy of 69%, whereas the EM algorithm increased the accuracy to 75%. On the same test set we applied another EM model by using a λ-value of 1. Here we only scored an accuracy of 72%, showing that the the augmented EM actually improves prediction accuracy when the correct λ-value is chosen.

4.2 Anomaly Detection with Naïve Bayes and EM

The prospect of finding an anomalous document inside a large corpus differs significantly from the problem in the previous section. For this paper, we define an anomaly as a document in a corpus that bears no relation to the other documents, i.e. the document is irrelevant to the rest of the documents in the corpus. That said, the relevant documents need to be related to each other by having the same class or similar classes. We therefore need a binary classifier that distinguishes documents between "normal" and "abnormal".

We continue to use the 20 Newsgroups as the basic data set. We simulate a data set with anomalous data by combining six overlapping classes that consist of topics such as electronics and science. We define this as our relevant (negative) data. We create our irrelevant (positive) data by combining the remaining

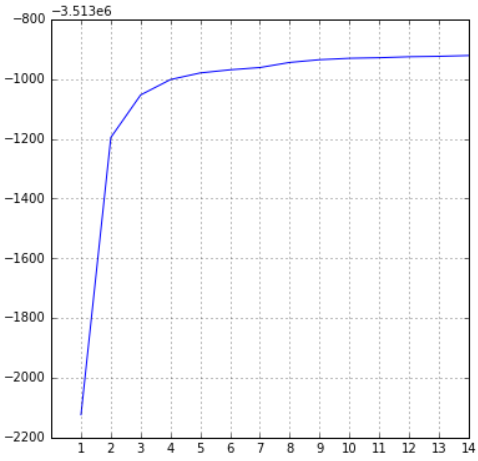


Figure 1: EM convergence on 10-class data set

fourteen classes that do not relate to electronics or science. Note that the irrelevant documents do not necessarily need to be relevant to each other.

In the training set, our negative class consists of 300 documents and the positive class contains 100 documents. An anomaly is, by definition, not expected to occur in a corpus as frequently as a "relevant" document is. This results in a large class imbalance in any real-world supervised anomaly detection environment. We attempt to replicate this class imbalance to some extent by creating an unlabelled data set with 800 relevant documents and 100 anomalous documents.

The ordinary Naïve Bayes classifier shows an unusually high accuracy of 89%. It is known that accuracy cannot be used as the only performance measure in the presence of class imbalance [8]. We therefore require a different method to measure the performance of our model. The Receiver Operating Characteristic (ROC) curve reflects the model's ability to distinguish between two variables with overlapping distributions in a binary classification environment, making this evaluation technique more appropriate in this instance than a standard hit-ratio.

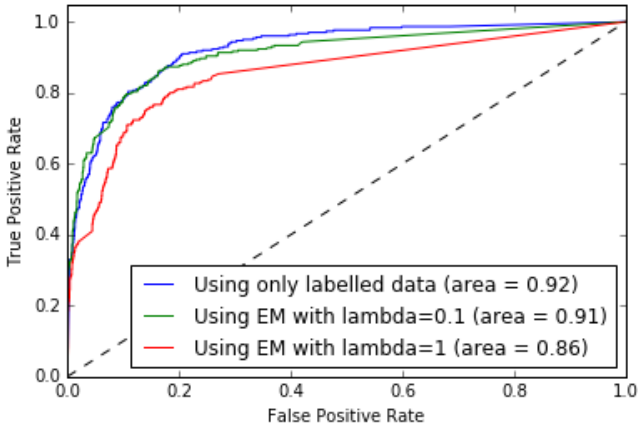


Figure 2: ROC curve for augmented binary classifier on 20 Newsgroups data

Figure 2 illustrates how class imbalance affects the model's performance when using different values of λ.

Here we see a graphical illustration of the damage to the model when adding unlabelled data that violate assumptions. An optimal value of λ therefore depends on the domain in which the model is applied, and some trial-and-error approximation. In this case, a small value of λ is preferable since the set of unlabelled data increases the magnitude of the present class imbalance through EM. With a λ-value of 0.1 we increased the accuracy to 93%, although the area under the ROC curve has dropped with 0.01. It is clear that the unlabelled data has worsened the class-inbalance and further skewed the metrics we used to evaluate our model.

5 Conclusion

Our goal was to find a simple solution to the generally complex problem that is textual anomaly detection. We attempted to use large amounts of unlabelled data to increase the performance of our Naïve Bayes classifier. As was illustrated in this paper, unlabelled text data can significantly improve a classifier given the correct conditions. The classifier we constructed for the 10-class data set showed a 5% increase in prediction accuracy once we implemented augmented EM. As shown extensively in [9] and in Figure 2, different values of the λ-scalar will produce different results. One constraint that we found in this paper is that large amounts of unlabelled data are required for a sustainable increase in prediction accuracy.

Under the binary classifier we constructed for anomaly detection, we encountered the problem of class imbalance. This is problematic for EM since any large amount of unlabelled text data in an anomaly detection environment could worsen the class imbalance and might even decrease accuracy after EM convergence. We aim to apply this algorithm to a more realistic data set, similar to the Liberty example mentioned in the introduction. This would allow us to observe how the semi-supervised model performs under larger violations of assumptions and class imbalance. A suggested improvement on our algorithm would be to address the class imbalance by introducing boosting which has been shown to work well with Naïve Bayes and unbalanced classes in past work [4].

Further research endeavors will involve the use of topic modelling to address anomaly detection in an unsupervised manner. Furthermore, we want to extend our research to combine a topic model such as LDA with a classifier. This is not a trivial exercise, since LDA does not have a 'complete data counterpart' in the same way that the EM algorithm has with Naïve Bayes.

References

[1] V Chandola, A Banerjee, and V Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2007.

[2] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (Methodological)*, pages 1–38, 1977.

[3] P Domingos and M Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

[4] C Elkan. Boosting and naive Bayesian learning. Technical report, University of California, San Diego, 1997.

[5] T Joachims. A probabilistic analysis of the Rocchio algorithm with tfidf for text categorization. Technical report, DTIC Document, 1996.

[6] R Kumaraswamy, A Wazalwar, T Khot, J Shavlik, and S Natarajan. Anomaly detection in text: The value of domain knowledge. In *The Twenty-Eighth International Flairs Conference*, 2015.

[7] A Mahapatra, N Srivastava, and J Srivastava. Contextual anomaly detection in text data. *Algorithms*, 5(4):469–489, 2012.

[8] KP Murphy. *Machine Learning: a Probabilistic Perspective*. MIT press, 2012.

[9] K Nigam. *Using unlabeled data to improve text classification*. PhD thesis, Massachusetts Institute of Technology, 2001.

[10] F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, pages 2825–2830, 2011.

Appendix: Python Code

Parameter estimation and class prediction module

```
# -*- coding: utf-8 -*-
"""
Created on Wed May 25 17:44:20 2016

@author: Carl Steyn
"""

import numpy as np
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import CountVectorizer
from gensim.utils import simple_preprocess
from math import exp, log
from scipy.misc import logsumexp
from sklearn import metrics
import matplotlib.pyplot as plt
from scipy.sparse import find

class MNB:

    def tokenize(self,text):
        tok = []
        for i in range(len(text)):
            tok.append(' '.join(simple_preprocess(text[i])))
        return tok

    def ROC(self,test_labels,probs,pos=1):
        fpr, tpr, thresholds = metrics.roc_curve(test_labels, probs[:,1],
                                                 pos_label=pos)
        roc_auc = metrics.auc(fpr,tpr)
        plt.figure()
        plt.plot(fpr, tpr, label='ROC curve (area = %0.2f)' % roc_auc)
        plt.plot([0, 1], [0, 1], 'k--')
        plt.xlim([0.0, 1.0])
        plt.ylim([0.0, 1.05])
        plt.xlabel('False Positive Rate')
        plt.ylabel('True Positive Rate')
        plt.title('Receiver operating Characteristic')
        plt.legend(loc="lower right")
        plt.show()

    def import_data(self,subset,categories,size):
        dataset = fetch_20newsgroups(subset = subset,categories=categories)
        labels = np.array(dataset.target)[0:size]
        dataset = self.tokenize(dataset.data[0:size])
        unique, counts = np.unique(labels, return_counts=True)
        labelcounts = np.asarray((unique,counts)).T
        return dataset,labels,labelcounts
```

```
def vectorize(self,data):
    countvect = CountVectorizer(stop_words = 'english', lowercase = False)
    counts = countvect.fit_transform(data)
    V = len(countvect.get_feature_names())
    D = len(data)
    return counts,D,V

    # E-step: Calculate probabilistic labels for unlabelled data
def posterior(self,counts,wordprior,classprior,D_U,C):
    array = counts.toarray()
    predict = np.zeros(shape=(D_U,C),dtype='float')
    for index,k in np.ndenumerate(predict):
        words = [i for i,x in enumerate(array[index[0]]) if x > 0]
        docprobs = []
        for i in words:
            docprobs.append(log(wordprior[i,index[1]]))
        docprobs.append(log(classprior[index[1]]))
        numerator_product = sum(docprobs)
        S = []
        for j in range(C):
            L = []
            for i in words:
                L.append(log(wordprior[i,j]))
            L.append(log(classprior[j]))
            S.append(sum(L))
        denom_product = logsumexp(S)
        predict[index] = exp(numerator_product-denom_product)
    return predict

    # M-step: Estimate model parameters using MAP estimation
def parameters_EM(self,training_counts,test_counts,z_L,z_U,V,C,scalar):

   training_D = (training_counts[:,0].shape)[0]
   test_D = (test_counts[:,0].shape)[0]

   word_indices_U = np.asarray([find(test_counts[i,:])[1] for i in
   range(test_D)])
   word_indices_L = np.asarray([find(training_counts[i,:])[1] for i in
   range(training_D)])
   N_wt_di_L = [[(x,training_counts[i,x]) for x in word_indices_L[i]]
   for i in range(len(word_indices_L))]

   N_wt_di_U = [[(x,test_counts[i,x]) for x in word_indices_U[i]] for i in
   range(len(word_indices_U))]
   N_vector_L = [self.replaceNull([[x[1] for x in N_wt_di_L[i] if
   x[0]==word] for i in range(training_D)]) for word in range(V)]
   N_vector_U = [self.replaceNull([[x[1] for x in N_wt_di_U[i] if
   x[0]==word] for i in range(test_D)]) for word in range(V)]
   doccount_L = [sum([item[1] for item in N_wt_di_L[i]]) for i in
   range(len(N_wt_di_L))]
   doccount_U = [sum([item[1] for item in N_wt_di_U[i]]) for i in
   range(len(N_wt_di_U))]
   # M step
```

```
    denominator_L = np.zeros(C,dtype = float)
    denominator_U = np.zeros(C,dtype = float)
    for j in range(C):
        denominator_L[j] = np.dot(doccount_L,z_L[:,j])
        denominator_U[j] = np.dot(doccount_U,z_U[:,j])
    denominator = V + denominator_L + scalar*denominator_U
    numerator_L = np.zeros((V,C), dtype = float)
    numerator_U = np.zeros((V,C), dtype = float)
    for i in range(V):
        for j in range(C):
            numerator_L[i,j] = np.dot(N_vector_L[i],z_L[:,j])
            numerator_U[i,j] = np.dot(N_vector_U[i],z_U[:,j])
    numerator = 1 + numerator_L + scalar*numerator_U

    wordprob_EM = np.divide(numerator,denominator)

    classprior_EM = np.zeros(C,dtype='float')
    for j in range(C):
        classprior_EM[j] = (1 + scalar*z_U[:,j].sum() +
        z_L[:,j].sum())/(C + training_D + scalar*test_D)

    dirichlet = np.log(wordprob_EM).sum() + np.log(classprior_EM).sum()
    sum_U = 0
    sum_L = 0
    for i in range(test_D):
        for j in range(C):
            sum_U += scalar*z_U[i,j]*(np.log(classprior_EM[j]) - np.log(V)
            + np.log(wordprob_EM[find(test_counts[i,:])[1],j]).sum())

    for i in range(training_D):
        for j in range(C):
            sum_L += z_L[i,j]*(np.log(classprior_EM[j]) - np.log(V) +
            np.log(wordprob_EM[find(training_counts[i,:])[1],j]).sum())

    log_likelihood = dirichlet + sum_U + sum_L

    return wordprob_EM , classprior_EM , log_likelihood

def replaceNull(self,x): # Replace empty values in a vector with 0's
    for i,item in enumerate(x):
        if not item:
            x[i] = 0
        else:
            x[i] = item[0]

    return x
```

Application to binary classification

```
# -*- coding: utf-8 -*-
"""
Created on Fri Jul 15 11:56:03 2016

@author: Carl Steyn
"""
import os
os.chdir('C:\\Carl\\Work\\2016\\WST_795\\Binary_classificaton_on_liberty_data')
import numpy as np
from sklearn.datasets import fetch_20newsgroups
from class_MNB import MNB
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score
from scipy.sparse import find
import matplotlib.pyplot as plt


# Import and vectorize training data
mnb = MNB()
categories = ['alt.atheism','comp.graphics','comp.os.ms-windows.misc',
'comp.sys.ibm.pc.hardware','comp.sys.mac.hardware','comp.windows.x',
'misc.forsale','rec.autos','rec.motorcycles','rec.sport.baseball',
'rec.sport.hockey','sci.crypt','sci.electronics','sci.med','sci.space',
'soc.religion.christian','talk.politics.guns','talk.politics.mideast',
'talk.politics.misc','talk.religion.misc']

neg_category = ['comp.graphics','comp.os.ms-windows.misc',
'comp.sys.ibm.pc.hardware','comp.sys.mac.hardware','comp.windows.x',
'sci.electronics']
pos_category = [x for x in categories if x not in neg_category]

positive = fetch_20newsgroups(subset = 'train',remove=('headers', 'footers',
'quotes'),categories=pos_category,shuffle=True, random_state=42).data[0:100]
negative = fetch_20newsgroups(subset = 'train',remove=('headers', 'footers',
'quotes'),categories=neg_category,shuffle=True, random_state=42).data[0:300]

training_data = positive + negative
training_data = mnb.tokenize(training_data)
countvect = CountVectorizer(stop_words = 'english', lowercase = False)
training_counts = countvect.fit_transform(training_data)
training_D = len(training_data)
V = len(countvect.get_feature_names())

# Create labels for positive and negative classes
pos_label = np.full((len(positive),1),1,dtype=int)
neg_label = np.full((len(negative),1),0,dtype=int)
training_labels = np.append(pos_label,neg_label)

# Z variable for labelled data
z_L = np.array(np.c_[(training_labels == 0),(training_labels == 1)].astype(int))
```

```
# Initial M-step (Naive Bayes on labelled data only)
word_indices_L = np.asarray([find(training_counts[i,:])[1]
for i in range(training_D)]) :

N_wt_di_L = [[(x,training_counts[i,x]) for x in word_indices_L[i]]
for i in range(len(word_indices_L))] #word id's and counts in each doc

N_vector_L = [mnb.replaceNull([[x[1] for x in N_wt_di_L[i] if x[0]==word]
for i in range(training_D)]) for word in range(V)] #vector of word-doc counts

doccount = [sum([item[1] for item in N_wt_di_L[i]])
for i in range(len(N_wt_di_L))] #total number of words in each doc

wordprob = np.zeros(shape=(V,2),dtype='float')
for i in range(V):
    numerator = (1 + np.dot(N_vector_L[i],z_L[:,0])),
(1 + np.dot(N_vector_L[i],z_L[:,1]))
    denominator = (V + np.dot(doccount,z_L[:,0]),V + np.dot(doccount,z_L[:,1]))
    wordprob[i] = (np.divide(numerator,denominator))

classprior = np.array((1 + z_L[:,0].sum(),1 + z_L[:,1].sum()))/(2 + training_D)
np.save('wordprob1.npy',wordprob)
np.save('classprior1.npy',classprior)
# log_likelihood
sum_L = 0
dirichlet = (np.log(wordprob[:,0]).sum() + np.log(classprior[0])) +
(np.log(wordprob[:,1]).sum() + np.log(classprior[1]))
for i in range(training_D):
    for j in range(2):
        sum_L += z_L[i,j]*(np.log(classprior[j]) - np.log(V) +
        np.log(wordprob[find(training_counts[i,:])[1],j]).sum())
log_likelihood = dirichlet + sum_L

#Import unlabelled data
pos_test = fetch_20newsgroups(subset = 'test',remove=('headers', 'footers',
'quotes'),categories=pos_category,shuffle=True, random_state=42).data[0:100]
neg_test = fetch_20newsgroups(subset = 'test',remove=('headers', 'footers',
'quotes'),categories=neg_category,shuffle=True, random_state=42).data[0:800]
pos_label = np.full((len(pos_test),1),1,dtype=int)
neg_label = np.full((len(neg_test),1),0,dtype=int)
test_labels = np.append(pos_label,neg_label)
test_data = pos_test + neg_test
test_data = mnb.tokenize(test_data)
test_counts = countvect.transform(test_data)
test_D = len(test_data)
unique, counts = np.unique(test_labels, return_counts=True)
test_labelcounts = np.asarray((unique,counts)).T
word_indices_U = np.asarray([find(test_counts[i,:])[1] for i in range(test_D)])

# First E-step
z_U = mnb.posterior(test_counts,wordprob,classprior,test_D,2)
mnb.ROC(test_labels,z_U) # ROC curve for Naive Bayes w/o EM
```

```
# Reconstruct data for EM
scalar = 1
new_data = training_data + test_data
new_counts = countvect.fit_transform(new_data)
training_counts = countvect.transform(training_data)
test_counts = countvect.transform(test_data)
new_D = len(new_data)
test_D = len(test_data)
V = len(countvect.get_feature_names())

log_likelihood = []
accuracy = []
inc = 0
# EM iteration starts here
# M-step
while True:
    wordprob_EM , classprior_EM , ll = mnb.parameters_EM(training_counts,
    test_counts,z_L,z_U,V,2,scalar)
    inc += 1
    print('Iteration %d:' %inc)
    log_likelihood.append(ll)
# E-step
    z_U = mnb.posterior(test_counts,wordprob_EM,classprior_EM,test_D,2)
    MAP = np.zeros(shape=test_D,dtype='int64')
    for i in enumerate(z_U):
        MAP[i[0]] = np.argmax(z_U[i[0]],axis=0)
    acc = accuracy_score(test_labels,MAP,normalize=True)
    accuracy.append(acc)
# output
    print('log-likelihood = %d\n' %ll)
    if inc>2:
        if ll == log_likelihood[-2]:
            break
#
#plt.subplot(222)
plt.figure(figsize=(6,6))
x = np.arange(1,len(log_likelihood)+1)
plt.plot(x,log_likelihood)
plt.yscale('linear')
plt.title('log-likelihood')
plt.xticks(np.arange(min(x), max(x)+1, 1))
plt.grid(True)

plt.figure(figsize=(6,6))
x = np.arange(1,len(accuracy)+1)
plt.plot(x,accuracy)
plt.yscale('linear')
plt.title('accuracy')
plt.xticks(np.arange(min(x), max(x)+1, 1))
plt.grid(True)
```

Application to multi-class classification

```
# -*- coding: utf-8 -*-
```

```
"""
Created on Sun Sep  4 14:21:04 2016

@author: Carl Steyn
"""

import os
os.chdir('C:\\Carl\\Work\\2016\\WST_795\\NB_augmentedEM_20newsgroups')
import numpy as np
from sklearn.datasets import fetch_20newsgroups
from class_MNB import MNB
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score , f1_score
from scipy.sparse import find
import matplotlib.pyplot as plt

# NAIVE BAYES
# Import and vectorize training data
mnb = MNB()
countvect = CountVectorizer(stop_words = 'english', lowercase = False)
categories = ['alt.atheism','comp.graphics','comp.os.ms-windows.misc',
'comp.sys.ibm.pc.hardware','comp.sys.mac.hardware','comp.windows.x',
'misc.forsale','rec.autos','rec.motorcycles','rec.sport.baseball',
'rec.sport.hockey','sci.crypt','sci.electronics','sci.med','sci.space',
'soc.religion.christian','talk.politics.guns','talk.politics.mideast',
'talk.politics.misc','talk.religion.misc']

training = fetch_20newsgroups(subset = 'train',categories=categories,shuffle=True)
class1 = [training.data[i] for i,x in enumerate(training.target) if
training.target_names[x] ==  'comp.graphics'][0:10]
class2 = [training.data[i] for i,x in enumerate(training.target) if
training.target_names[x] ==  'talk.politics.mideast'][0:10]
class3 = [training.data[i] for i,x in enumerate(training.target) if
 training.target_names[x] ==  'sci.electronics'][0:10]
class4 = [training.data[i] for i,x in enumerate(training.target) if
 training.target_names[x] ==  'rec.sport.baseball'][0:10]
class5 = [training.data[i] for i,x in enumerate(training.target) if
training.target_names[x] ==  'rec.motorcycles'][0:10]
class6 = [training.data[i] for i,x in enumerate(training.target) if
training.target_names[x] ==  'soc.religion.christian'][0:10]
class7 = [training.data[i] for i,x in enumerate(training.target) if
training.target_names[x] ==  'comp.sys.mac.hardware'][0:10]
class8 = [training.data[i] for i,x in enumerate(training.target) if
training.target_names[x] ==  'rec.autos'][0:10]
class9 = [training.data[i] for i,x in enumerate(training.target) if
training.target_names[x] ==  'alt.atheism'][0:10]
class10 = [training.data[i] for i,x in enumerate(training.target) if
training.target_names[x] ==  'misc.forsale'][0:10]
training_D = len(class1) + len(class2) + len(class3) + len(class4) +
len(class5) + len(class6) + len(class7) + len(class8) + len(class9)
 + len(class10)
training_data = mnb.tokenize(class1 + class2 + class3 + class4 + class5 +
 class6 + class7 + class8 + class9 + class10)
```

```
training_labels = []
C = 10
for i in range(C):
    training_labels.extend([i]*len(class1))
training_labels = np.array(training_labels)

training_counts = countvect.fit_transform(training_data)
V = len(countvect.get_feature_names())
unique, counts = np.unique(training_labels, return_counts=True)
training_labelcounts = np.asarray((unique,counts)).T
z_L = np.zeros((training_D,C), dtype=float)
for i in range(C):
    z_L[:,i] = (training_labels == i)
word_indices_L = np.asarray([find(training_counts[i,:])[1]
for i in range(training_D)])

N_wt_di_L = [[(x,training_counts[i,x]) for x in word_indices_L[i]]
for i in range(len(word_indices_L))]
N_vector_L = [mnb.replaceNull([[x[1] for x in N_wt_di_L[i] if x[0]==word]
for i in range(training_D)]) for word in range(V)]
doccount = [sum([item[1] for item in N_wt_di_L[i]])
for i in range(len(N_wt_di_L))]

# Find MAP estimates of parameters
denominator = np.zeros(C, dtype=float)
for j in range(C):
    denominator[j] = V + np.dot(doccount,z_L[:,j])
numerator = np.zeros((V,C), dtype=float)
for i in range(V):
    for j in range(C):
        numerator[i,j] = 1 + np.dot(N_vector_L[i],z_L[:,j])
wordprob = np.divide(numerator,denominator)
classprior = np.zeros(C,dtype='float')
for j in range(C):
    classprior[j] = (1 + z_L[:,j].sum())/(C + training_D)

# Import unlabelled data
scalar = 0.3
test = fetch_20newsgroups(subset = 'test',categories=categories,shuffle=True)
#

class1 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] == 'comp.graphics'][0:20]
class2 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] == 'talk.politics.mideast'][0:20]
class3 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] == 'sci.electronics'][0:20]
class4 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] == 'rec.sport.baseball'][0:20]
class5 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] == 'rec.motorcycles'][0:20]
class6 = [test.data[i] for i,x in enumerate(test.target)
```

```
if test.target_names[x] ==  'soc.religion.christian'][0:20]
class7 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'comp.sys.mac.hardware'][0:20]
class8 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'rec.autos'][0:20]
class9 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'alt.atheism'][0:20]
class10 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'misc.forsale'][0:20]
test_D = len(class1) + len(class2) + len(class3) + len(class4) + len(class5) +
len(class6) + len(class7) + len(class8) + len(class9) + len(class10)
test_data = mnb.tokenize(class1 + class2 + class3 + class4 + class5 + class6 +
class7 + class8 + class9 + class10)

test_labels = []
for i in range(C):
   test_labels.extend([i]*len(class1))
test_labels = np.array(test_labels)
test_D = len(test_data)

test_counts = countvect.transform(test_data)
word_indices_U = np.asarray([find(test_counts[i,:])[1] for i in range(test_D)])
# Estimate probabilitstic labels for unlabelled data
z_U = mnb.posterior(test_counts,wordprob,classprior,test_D,C)
MAP = np.zeros(shape=test_D,dtype='int64')
for i in enumerate(z_U):
    MAP[i[0]] = np.argmax(z_U[i[0]],axis=0)
print(accuracy_score(test_labels,MAP,normalize=True))
f1_score(test_labels,MAP,average='weighted')
# Restructure data for EM
EM_data = training_data + test_data;
EM_counts = countvect.fit_transform(EM_data)
EM_D = len(EM_data)
training_counts = countvect.transform(training_data)
test_counts = countvect.transform(test_data)
V = len(countvect.get_feature_names())
inc = 0
log_likelihood = []
accuracy = []

class1 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'comp.graphics'][100:200]
class2 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'talk.politics.mideast'][100:200]
class3 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'sci.electronics'][100:200]
class4 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'rec.sport.baseball'][100:200]
class5 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'rec.motorcycles'][100:200]
class6 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'soc.religion.christian'][100:200]
class7 = [test.data[i] for i,x in enumerate(test.target)
```

```
if test.target_names[x] ==  'comp.sys.mac.hardware'][100:200]
class8 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'rec.autos'][100:200]
class9 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'alt.atheism'][100:200]
class10 = [test.data[i] for i,x in enumerate(test.target)
if test.target_names[x] ==  'misc.forsale'][100:200]
EM_test_D = len(class1) + len(class2) + len(class3) + len(class4) + len(class5)
 + len(class6) + len(class7) + len(class8) + len(class9) + len(class10)
EM_test_data = mnb.tokenize(class1 + class2 + class3 + class4 + class5 +
class6 + class7 + class8 + class9 + class10)
C = 10
EM_test_labels = []
for i in range(C):
    EM_test_labels.extend([i]*len(class1))
EM_test_labels = np.array(EM_test_labels)
EM_test_counts = countvect.transform(EM_test_data)


# EM iteration starts here
while True:
# M-step
    inc += 1
    print('Iteration %d:' %inc)
    wordprob_EM , classprior_EM , ll = mnb.parameters_EM(training_counts,
    test_counts,z_L,z_U,V,C,scalar)
    log_likelihood.append(ll)
# E-step
    z_U = mnb.posterior(test_counts,wordprob_EM,classprior_EM,test_D,C)
    z_EM_test = mnb.posterior(EM_test_counts,wordprob_EM,classprior_EM,
                              EM_test_D,C) #test on external test data

    EM_MAP = np.zeros(shape=EM_test_D,dtype='int64')
    for i in enumerate(z_EM_test):
        EM_MAP[i[0]] = np.argmax(z_EM_test[i[0]],axis=0)
# Model evaluation and output

    acc = accuracy_score(EM_test_labels,EM_MAP,normalize=True)
    f1 = f1_score(EM_test_labels,EM_MAP,average='weighted')
    accuracy.append(acc)
    print('Accuracy: %f' %acc)
    print('log-likelihood: %f' %ll)
    print('f1_score L %f' %f1)
# Convergence checker
    if inc>2:
        if ll == log_likelihood[-2]:

            break
#np.save('log_likelihood_300train_1000test',log_likelihood)
# Plot log-likelihood after convergence
plt.figure(figsize=(6,6))
x = np.arange(1,len(log_likelihood)+1)
plt.plot(x,log_likelihood)
```

```
plt.yscale('linear')
plt.title('log-likelihood')
plt.xticks(np.arange(min(x), max(x)+1, 1))
plt.grid(True)

#Plot change in accuracy
plt.figure(figsize=(6,6))
x = np.arange(1,len(accuracy)+1)
plt.plot(x,accuracy)
plt.yscale('linear')
plt.title('accuracy')
plt.xticks(np.arange(min(x), max(x)+1, 1))
plt.grid(True)
#%%
```

Robot localisation with the EZRobot

Carel van Niekerk 13013492

WST795 Research Report

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

Supervisor: Dr. I. Fabris-Rotelli

Department of Statistics, University of Pretoria

2 November 2016

Abstract

In this report we look at the Kalman filter algorithm, the Markov localisation algorithm and solving the robot localisation problem in general. We then apply the Markov localisation algorithm, as a solution to the robot localisation problem, to the EZRobot to find its location in a corridor as a simple illustration of how these algorithms can be used.

Declaration

I, *Carel van Niekerk,* 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.

_____
Carel van Niekerk

_____
Dr. I. Fabris-Rotelli

_____
Date

Acknowledgements

This work is based on the research supported in part by the National Research Foundation of South Africa for the grant number 90315. This work was also supported by STATOMET at the Department of Statistics, University of Pretoria. The views expressed in this report are those of the author and do not necessarily reflect those of STATOMET or the NRF. Neither STATOMET nor the NRF are responsible for the information provided in this document.

Contents

1 Introduction **6**

2 Literature Review **6**
2.1 Robot localisation 6
2.1.1 State of the art localisation methods 7
2.2 The Kalman filter 7
2.3 Markov localisation 7

3 Robot localisation **7**
3.1 Information Sources 8
3.2 Localisation techniques 8

4 The Kalman filter **9**
4.1 Background information 9
4.2 Kalman filter algorithm 10
4.3 The extended Kalman filter algorithm 10

5 Markov localisation **14**

6 Application **19**

Appendix **22**

List of Figures

1 Illustration of robot movement 6
2 Wall follower illustration 8
3 Kalman filter process diagram 9
4 Comparison of Kalman filter estimates and measurement positions 11
5 Map of the environment in the extended Kalman filter example 12
6 Likelihood contour plot for extended Kalman filter estimates. 15
7 Markov localisation algorithm diagram 16
8 Markov localisation example illustration 17
9 Markov localisation example belief function graphs 18
10 Map of corridor for application 19
11 Belief function graphs for the application. 20

1 Introduction

Visualize trying to find an object in the dark. This task could be very challenging, not because you do not know where the object is but rather because it is very difficult to know where you are relative to the object. When designing and building an autonomous robot, one faces the same challenge of enabling the robot to position itself. You may say that the robot has sensors and can measure how far it moves, but with all these inputs it is still a challenge for the robot to determine its exact location in an environment using only the sensor input. This problem is known as the robot localisation problem. One might think that positioning an autonomous robot is straightforward and that if you program a robot to move 5 cm forward, turn 19° to the left and then move another 3 cm forward it would end up in the desired position illustrated by the green dot in Figure 1. However, in reality the outcome may be different with the robot ending up at the red dot due to bad surface traction.

Figure 1: Illustration of robot movement

The input that a robot receives from its sensors contains noise and cannot be interpreted without applying some sort of a belief function. By solving the robot localisation problem using a localisation algorithm we can combine the information from the sensors, the knowledge the robot has about its movement and uncertainty we have, due to noise, to determine an estimated pose which we are less uncertain about. According to Thrun in his book Probabilistic Robotics [17] localization is to find a connection between the coordinates on a map and the robots local coordinates. In this report we cover the Kalman filter algorithm, the Markov localisation algorithm and solving the robot localisation problem. We also cover exactly what robot localization entails. We then apply the Markov localisation algorithm, as a solution to the robot localisation problem, to the EZRobot to find its location in a corridor as a simple illustration of how these algorithms can be used.

According to [17] localisation is a problem which can be split into many different subgroups such as local and global localisation, where in local localisation the initial pose of a robot is known and in global localisation it is unknown, or single- and multi- robot localisation. In this report we will focus on global single-robot localisation and also using the Markov assumption, meaning we will assume that our belief of the pose of the robot is only based on its previous position and no position before that.

In further research attention is also given to state of the art robot localisation techniques such as SLAM (Simultaneous Localisation and Mapping) and the use of RGB-D cameras (cameras which provides an RGB image as well as per-pixel depth data) to create a 3D mapped image of the robots environment. These are the same cameras used in games such as the Microsoft Xbox Kinect to detect where the players are and what they are doing.

2 Literature Review

2.1 Robot localisation

Localisation is described as one of the most important topics in robotics and artificial intelligence by many resources such as [17, 12, 16]. The localisation problem is the problem of linking a robots pose to a coordinate on a map. Thrun [16] illustrates how a small error in measurements can have a large effect on the pose of a robot after a few times and the importance of using a localisation algorithm. In this source

different localisation algorithms such as Multi-Planar Maps, expectation maximisation, Bayesian filters, etc are also compared and explained. Localisation can be broken up into many different subgroups such as local and global localisation as well as single and multi-robot localisation. These different types of localisation are described in Chapter 7 of the book Probabilistic Robotics [17]. In this book the concept of localising using the Markov assumption is also explained, where the current pose of the robot only depends on current measurement inputs and its previous position and no positions before that [17]. In [17, 12] explanations are also given on how the Kalman filter can be applied to solve the problem of localisation and noisy systems using the Kalman filter. In [10, 3] mapping, localisation and path finding strategies , that can be used during the application of the Kalman filter to solve the localisation problem, are discussed. In [4] the Markov localisation approach is discussed in detail.

2.1.1 State of the art localisation methods

A key area of robotics that corresponds with the localisation problem is environment mapping. According to [14] these two problems can be solved simultaneously using a procedure called Simultaneous Localisation and Mapping (SLAM) which as its name suggests localises the robots and maps the surroundings of the robot simultaneously. The idea of SLAM was first discussed at the 1986 IEEE Robotics and Automation Conference by researchers such as Cheeseman, Crowley and Durrant-Whyte [1]. In this source they also discuss the latest developments in multi-robot SLAM (where multiple robots communicate to localise more efficiently and to create better maps of their surroundings). In [6, 7] the use of RGB-D cameras, such as those used in the X-box Kinect, to create 3D maps of the robots environment is discussed. These sources also discuss the use of RGB-D cameras for SLAM with the iterative closest point algorithm (ICP). The disadvantages such as a lack of robustness of RGB-D 3D maps are also discussed.

2.2 The Kalman filter

The Kalman filter algorithm designed by Swerling and Kalman [5] is a method of filtering and estimating in linear Gaussian systems. The algorithm is explained in depth in the book Probabilistic Robotics [17]. An illustration of how the Kalman filter can be applied to solve the problem of robot localisation is given in [8]. In [18] Tusell discusses and compares different packages in R [13] that applies the Kalman filter in state space estimation. In [17] the extended Kalman filter, a extension of the Kalman filter which allows for non-linear systems, is also explained.

2.3 Markov localisation

The Markov localisation algorithm is a localisation algorithm derived from Bayes filter and it can be used to solve the global localisation problem given that we have a map input. The Markov localisation algorithm is discussed in detail in [4, 17].

3 Robot localisation

According to Thrun [17] the robot localisation problem can be defined as predicting a robots pose relative to some external reference system. Localisation can hence be described as the answer to the question by the robot "Where am I?". This is seen to be one of the most important problems in the field of autonomous robotics by researchers such as Thrun [17] and Negenborn [12], for example. According to Leonard and Durrant-Whyte [9] an important aspect in autonomous robotics is the robot's ability to navigate, and the first step in navigation is finding out where you are (localising) after which the robot can decide where it wants to go and how to get there. There are many different algorithms which can be used for localisation, which all have their respective advantages and disadvantages which needs to be considered during application. As discussed by Filliate and Meyer [3] two of the more difficult aspects of navigation is learning and mapping the surroundings of the robot and finding the position of the robot in that environment (localisation). So in the state of the art localisation algorithms focus is placed on not only finding out where the robot is but also learning its environment and mapping it at the same time, this technique is reffered to as Simultaneous

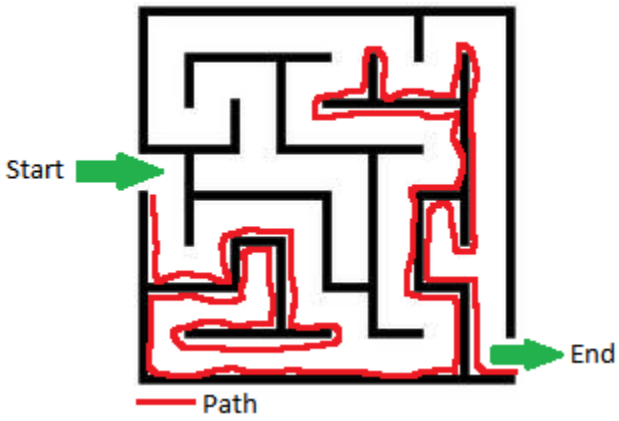


Figure 2: Wall follower illustration

Localisation and Mapping (SLAM). This idea was first discussed at the 1986 IEEE Robotics and Automation Conference by researchers Cheeseman, Crowley and Durrant-Whyte [1].

3.1 Information Sources

In the processes of localisation or learning and mapping the surroundings of a robot, information about the environment the robot is in is required. Generally this information is extracted from sensors and the movements of the robot. This information is categorized into two categories, idiothetic and allothetic sources [10]. Idiothetic information is information about the motion of the robot such as distance moved or angle turned, etc. Allothetic information is information about the surroundings of the robot such as sensor information about where obstacles and targets are, etc. The quality of this information is mainly dependent on the hardware used to build the robot. With technological advances such as RGB-D cameras three dimentional maps of an environment can be plotted and used in the localisation process, or with fast wireless communications between multiple robots informations from all robots can used as an extra information source to make localisation more efficient [6, 7, 14].

3.2 Localisation techniques

The problem with most of the information the robot has to use for localisation is that it is not reliable enough because of natural error in the information. Two of the many algorithms which help eliminate this error and more accurately estimate the position of the robot is the Kalman filter and the Markov localisation algorithm [16, 5, 4]. Later in this report we will discuss the intricate details of the Kalman filter and the Markov localisation algorithms and how to apply them, but for now we will discuss how they can be used for localisation and mapping. It is clear that if we had a predefined detailed map of the environment it would be easy to just use the estimated position from the Kalman filter as the position of the robot. But in reality there would possibly be more than one information source in which case we would apply the Kalman filter multiple times at a single timestep to implement all of the available information. In the case of an unknown environment the Kalman filter can also be used to estimate the position of objects such as walls or a ball or any other obstacles or targets [5]. In the case of a known environment but an unknown starting position we can apply the Markov localisation to effectively find the current position of the robot.

A simple example of where localisation can be used is traversing through a maze. A simple algorithm which can be used to achieve this is the wall follower algorithm described by Mishra and Bande [11]. To solve a maze the robot simply keeps driving next to the left/right wall which it started next to and it will reach the end of the maze as illustrated in Figure 2 .

As it can be seen in this illustration the robot would simply follow the right hand wall until it exits the maze. This makes the decision making on where to go very simple if the robot can turn right then turn right,

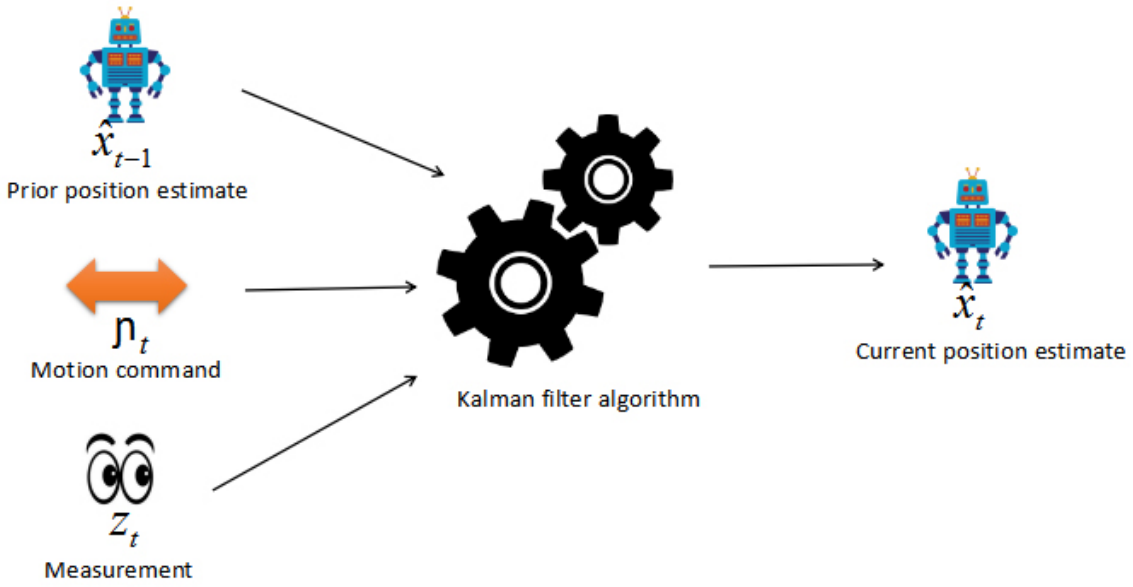


Figure 3: Kalman filter process diagram

if the robot can't turn right but can go straight then go straight, if the robot can't turn right or continue going straight then turn left and if the robot reaches a dead end turn around. So we now have a simple system the robot can use to answer "Where do I want to go next?". It can be seen that this is not always the most efficient choice though. If the left hand wall was chosen the path would be very long.

4 The Kalman filter

4.1 Background information

The Kalman filter algorithm designed by Swerling and Kalman [5] is a method of filtering and estimating in linear Gaussian systems. This algorithm recognises that measurement data in a state space model has an error factor, which would result in direct interpretation of measurement data being unreliable. Thus this algorithm allows for this error factor by adjusting the effect a measurement would have, based on its error factor, hence it provides a smooth estimate (estimates which like the real world states progresses realistically and does not jump around) for the state space model. This is important in robotics because it will be hard to make decisions about which actions to take if the estimated pose jumps around randomly [17, 5].
Assumptions made by the Kalman filter:

1. Noise terms in model are Gaussian noise with zero mean.
2. Belief function of a state is Gaussian (the current state has a multivariate normal distribution).
3. Markov assumption (the current state depends only on the current measurement information and the previous state and no states before that).
4. State space model is linear (the relationship between the current state and the previous state and the current state and the measurement information is linear).

The diagram in Figure 3 illustrates how the Kalman filter can be used to estimate the current pose of a robot.

4.2 Kalman filter algorithm

We call $\mathbf{x_t}$ the state vector of dimension $n \times 1$ at time t and it can be expressed as:

$$\mathbf{x_t} = \mathbf{A_t x_{t-1}} + \mathbf{B_t}\eta_\mathbf{t} + \epsilon_\mathbf{t}$$

with prior knowledge $\mathbf{x_0} \sim N(\mu_\mathbf{0}, \sum_\mathbf{0})$, where $\eta_\mathbf{t}$ is the control vector (the motion command given to the robot, for example move a units forward and b units left) at time t, $\mathbf{A_t}$ is a square matrix of size $n \times n$ (n is the dimension of the state vector), $\mathbf{B_t}$ is a matrix of size $n \times m$ (m is the dimension of the control vector) and $\epsilon_\mathbf{t}$ is a $n \times 1$ Gaussian random vector with zero mean and covariance matrix $\mathbf{R_t}$.
We call $\mathbf{y_t}$ the $k \times 1$ measurement vector (input from sensors, etc.) at time t and it can be expressed as:

$$\mathbf{y_t} = \mathbf{G_t x_t} + \delta_\mathbf{t}$$

where $\mathbf{G_t}$ is a matrix of size $k \times n$ (k is the dimension of the measurement vector) and $\delta_\mathbf{t}$ is a $k \times 1$ Gaussian random vector with zero mean and covariance matrix $\mathbf{Q_t}$.

The Kalman filter algorithm. [Probabilistic Robotics, Thrun [17]]
Input $\left(\mu_\mathbf{t-1}, \sum_\mathbf{t-1}, \eta_\mathbf{t}, \mathbf{y_t}\right)$

1. $\bar{\mu}_\mathbf{t} = \mathbf{A_t}\mu_\mathbf{t-1} + \mathbf{B_t}\eta_\mathbf{t}$ (Mean of belief $\bar{bel}(\mathbf{x_t})$ (Belief after motion command $\eta_\mathbf{t}$ is incorporated))
2. $\bar{\sum}_\mathbf{t} = \mathbf{A_t}\sum_\mathbf{t-1}\mathbf{A_t^T} + \mathbf{R_t}$ (Covariance of belief $\bar{bel}(\mathbf{x_t})$)
3. $\mathbf{K_t} = \bar{\sum}_\mathbf{t}\mathbf{G_t^T}\left(\mathbf{G_t}\bar{\sum}_\mathbf{t}\mathbf{G_t^T} + \mathbf{Q_t}\right)^{-1}$(Kalman gain)
4. $\mu_\mathbf{t} = \bar{\mu}_\mathbf{t} + \mathbf{K_t}\left(\mathbf{y_t} - \mathbf{G_t}\bar{\mu}_\mathbf{t}\right)$(Note $\hat{\mathbf{x}}_\mathbf{t} = \mu_\mathbf{t}$)
5. $\sum_\mathbf{t} = \left(\mathbf{I} - \mathbf{K_t G_t}\right)\bar{\sum}_\mathbf{t}$

Returns $(\mu_\mathbf{t}, \sum_\mathbf{t})$.

Example

To illustrate how the Kalman filter can be used a simple simulation is done where random measurement vectors are generated from a multivariate normal distribution with the pose of the robot as the mean of this distribution. It is also assumed that the measurement is a reading from a sensor which provides the x and y coordinates and the heading of the robot. The code for this simulation can be found in the appendix. The results of this simulation are illustrated in Figure 4.

In Figure 4 the Kalman filter estimates and simlated measurement positions are compared to illustrate how the Kalman filter can be applied in autonomous robotics. It can be seen from this illustration that the Kalman filter smoothly estimates the current pose. It can also be seen that the Kalman filter estimate is more accurate than the measurement position (The position of the robot based on servo input).

4.3 The extended Kalman filter algorithm

The system that we are using the Kalman filter in is the positioning of an autonomous robot on a map. One limitation of the Kalman filter in this situation is the fact that the relationship between states and motion commands and between states and meaurements must be linear. This is generally not the case in robotics

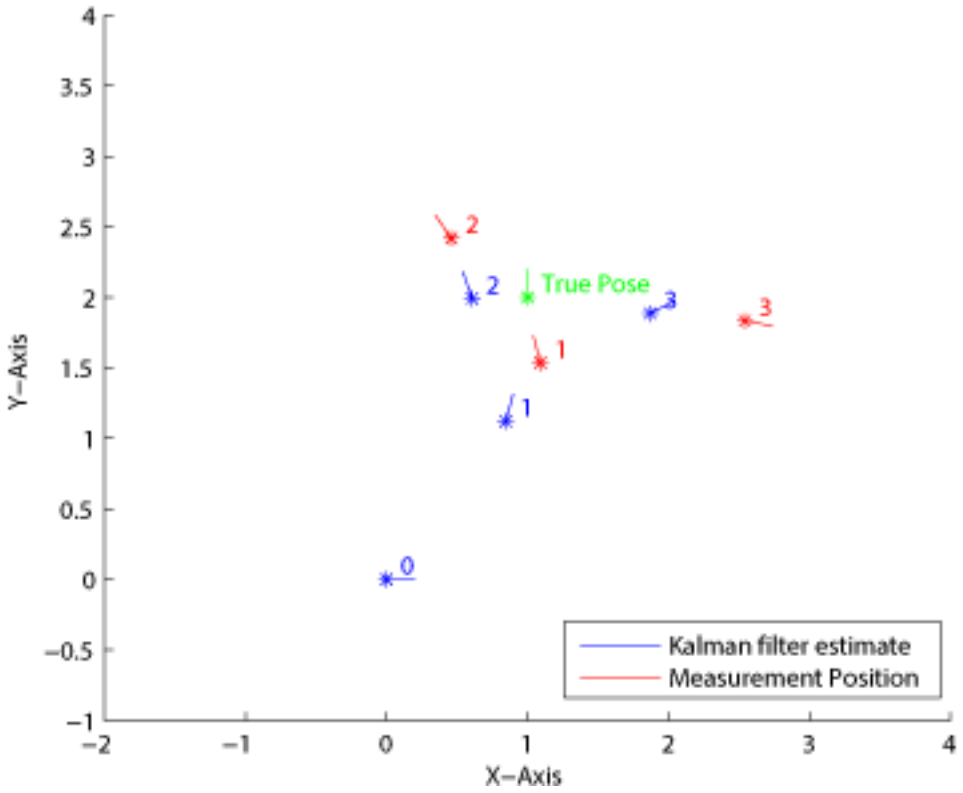


Figure 4: Comparison of Kalman filter estimates and measurement positions

because the measurements provided by the sensors on a robot do not often have a linear relationship with the position of the robot [5]. This problem is solved by extending the Kalman filter to allow these relationships to be non-linear.
We call $\mathbf{x_t}$ the state vector of dimension $n \times 1$ at time t and it can be expressed as:

$$\mathbf{x_t} = g(\mathbf{x_{t-1}}, \eta_t) + \epsilon_\mathbf{t}$$

with prior knowledge $\mathbf{x_0} \sim N(\mu_\mathbf{0}, \sum_\mathbf{0})$, where $g(\mathbf{x}, \eta)$ is a function of $\mathbf{x}$ and η (which does not have to be linear) and is a $n \times 1$ vector, $\eta_\mathbf{t}$ is the control vector (motion command) at time t and $\epsilon_\mathbf{t}$ is a $n \times 1$ Gaussian random vector with zero mean and covariance matrix $\mathbf{R_t}$.
We call $\mathbf{y_t}$ the $k \times 1$ measurement vector (input from sensors, etc.) at time t and it can be expressed as:

$$\mathbf{y_t} = h(\mathbf{x_t}) + \delta_\mathbf{t}$$

where $h(\mathbf{x})$ is a function of $\mathbf{x}$ (which does not have to be linear) and is a $k \times 1$ vector and $\delta_\mathbf{t}$ is a $k \times 1$ Gaussian random vector with zero mean and covariance matrix $\mathbf{Q_t}$.

The Extended Kalman filter algorithm. [Probabilistic Robotics, Thrun [17]]
Input $(\mu_{\mathbf{t-1}}, \sum_{\mathbf{t-1}}, \eta_\mathbf{t}, \mathbf{y_t})$

1. $\bar{\mu}_\mathbf{t} = g(\mu_{\mathbf{t-1}}, \eta_\mathbf{t})$ {Mean of belief $\bar{bel}(\mathbf{x_t})$ (Belief after motion command $\eta_\mathbf{t}$ is incorporated)}

2. $\mathbf{G_t} = \frac{\partial g}{\partial \mathbf{x}}|_{\mu_{t-1}, \eta_t}$
3. $\bar{\sum}_{\mathbf{t}} = \mathbf{G_t} \sum_{\mathbf{t-1}} \mathbf{G_t^T} + \mathbf{R_t}$ {Covariance of belief $\bar{bel}(\mathbf{x_t})$ }
4. $\mathbf{H_t} = \frac{\partial h}{\partial \mathbf{x}}|_{\bar{\mu}_t}$
5. $\mathbf{K_t} = \bar{\sum}_{\mathbf{t}} \mathbf{H_t^T} \left(\mathbf{H_t} \bar{\sum}_{\mathbf{t}} \mathbf{H_t^T} + \mathbf{Q_t}\right)^{-1}$(Kalman Gain)
6. $\mu_{\mathbf{t}} = \bar{\mu}_{\mathbf{t}} + \mathbf{K_t}\left(\mathbf{y_t} - h\left(\bar{\mu}_{\mathbf{t}}\right)\right)$
7. $\sum_{\mathbf{t}} = \left(\mathbf{I} - \mathbf{K_t H_t}\right) \bar{\sum}_{\mathbf{t}}$

Returns $(\mu_{\mathbf{t}}, \sum_{\mathbf{t}})$.

Example

Suppose a robot is positioned at (3,2) and there is a obstacle at position (8,9) as illustrated by the map in Figure 5.

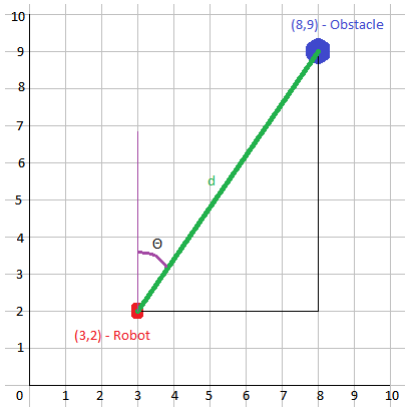


Figure 5: Map of the environment in the extended Kalman filter example

Also suppose this robot has a sensor which can detect the distance d from the object when pointed to the object at angle θ. This information is given to the robot in the form of a measurement vector $\mathbf{y} = \begin{pmatrix} d \\ \theta \end{pmatrix}$.

The theoretical value of this vector is calculated below.

$$\begin{aligned} d &= \sqrt{(8-3)^2 + (9-2)^2} \\ &= \sqrt{74} \\ &= 8.602325 \end{aligned}$$

and

$$\begin{aligned} \sin(\theta) &= \frac{|8-3|}{\sqrt{(8-3)^2 + (9-2)^2}} \\ &= \frac{5}{\sqrt{74}} \\ \therefore \theta &= \sin^{-1}\left(\frac{5}{\sqrt{74}}\right) \\ &= 0.620249 \end{aligned}$$

so the theoretical measurement vector in this situation is $\mathbf{y} = \begin{pmatrix} 8.602325 \\ 0.620249 \end{pmatrix}$.
Now for the extended Kalman filter we have the functions:

$$\begin{aligned} g(\mathbf{x}, \eta) &= \mathbf{x} + \eta = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} \\ &= \begin{pmatrix} x + \eta_1 \\ y + \eta_2 \end{pmatrix} \\ &= \begin{pmatrix} g_1(x, y, \eta_1, \eta_2) \\ g_2(x, y, \eta_1, \eta_2) \end{pmatrix} \end{aligned}$$

and

$$\begin{aligned} \mathbf{G_t} &= \frac{\partial g}{\partial \mathbf{x}}|_{\mu_{\mathbf{t-1}}, \eta_{\mathbf{t}}} \\ &= \begin{pmatrix} \frac{\partial g_1}{\partial x} & \frac{\partial g_1}{\partial y} \\ \frac{\partial g_2}{\partial x} & \frac{\partial g_2}{\partial y} \end{pmatrix} |_{\mu_{\mathbf{t-1}}, \eta_{\mathbf{t}}} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} |_{\mu_{\mathbf{t-1}}, \eta_{\mathbf{t}}} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

and

$$h(\mathbf{x}) = \begin{pmatrix} h_1(x, y) \\ h_2(x, y) \end{pmatrix}$$

where

$$\begin{aligned} h_1(x, y) &= \sqrt{(8 - x)^2 + (9 - y)^2} \\ h_2(x, y) &= sin^{-1}\left(\frac{|8 - x|}{\sqrt{(8 - x)^2 + (9 - y)^2}}\right) \end{aligned}$$

and

$$\begin{aligned} \mathbf{H_t} &= \frac{\partial h}{\partial \mathbf{x}}|_{\bar{\mu}_{\mathbf{t}}} \\ &= \begin{pmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} \\ \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} \end{pmatrix} |_{\bar{\mu}_{\mathbf{t}}} \end{aligned}$$

where

$$
\begin{aligned}
\frac{\partial h_1}{\partial x} &= \frac{x-8}{\sqrt{(8-x)^2+(9-y)^2}} \\
\frac{\partial h_1}{\partial y} &= \frac{y-9}{\sqrt{(8-x)^2+(9-y)^2}} \\
\frac{\partial h_2}{\partial x} &= \frac{\frac{(8-x)|8-x|}{\left((8-x)^2+(9-y)^2\right)^{\frac{3}{2}}} - \frac{8-x}{|8-x|\sqrt{(8-x)^2+(9-y)^2}}}{\sqrt{1-\frac{(8-x)^2}{(8-x)^2+(9-y)^2}}} \\
\frac{\partial h_2}{\partial y} &= \frac{(9-y)|8-x|}{\left((8-x)^2+(9-y)^2\right)^{\frac{3}{2}}\sqrt{1-\frac{(8-x)^2}{(8-x)^2+(9-y)^2}}}
\end{aligned}
$$

The code for the extended Kalman filter algorithm function in MATLAB [15] in this case is provided in the appendix.
A simple illustration of the application of the extended Kalman filter is done by generating a measurement vector from a multivariate normal distribution with the theoretical measurement vector calculated above as the mean and then applying the filter. The code for this application can be found in the appendix. The results of this example can be seen in Figure 6.
In Figure 6 the distributions of the Kalman filter estimates for the position of the robot after each iteration of applying the Kalman filter is compared. It can be seen that after each iteration the estimate becomes closer to the actual position at the point 3 on the x_1 axis and 2 on the x_2 axis. It can also be seen from the contour plot that the estimates become more reliable after each iteration which illustrates the usefullness of the Kalman filter.

5 Markov localisation

Global localisation is the problem of localisation where the robots initial state is not known. The Markov localisation algorithm is an algorithm which was derived from the Bayes filter using a map representing the environment of the robot as an input [4, 17]. The assumption made by the Markov localisation algorithm is the Markov assumption, namely the current state depends only on the current measurement information and the previous state and no states before that. The Markov localisation algorithm uses the known information about the robot's environment and information from its sensors together with a belief function of the robot's position to iteratively improve the belief function until it eventually knows exactly, or with high certainty where the robot is. To do this the algorithm firstly shifts the belief function of the previous time step to allow for the movement of the robot after which the Bayes filter is applied using the map input and sensor observations to refine the belief function. This new belief function can then be used to estimate where the robot could possibly be and then navigate to where it wants to go.
The diagram in Figure 7 illustrates how the Markov localisation algorithm can be used to estimate the current pose of a robot.

The Markov localisation algorithm. [4]
We call x_t the state of the robot, η_t the control command, z_t the measurement, $bel\left(x_{t-1}\right)$ the belief function of the previous state at time t and m the map. The belief function represents the belief that a state takes on a specified value.
Input $\left(bel\left(x_{t-1}\right), \eta_t, z_t, m\right)$
For all possible values x_t can take on, say all l, do:

1. Use η_t to appropriately shift the $bel\left(x_{t-1}\right)$ function to allow for the movement of the robot.

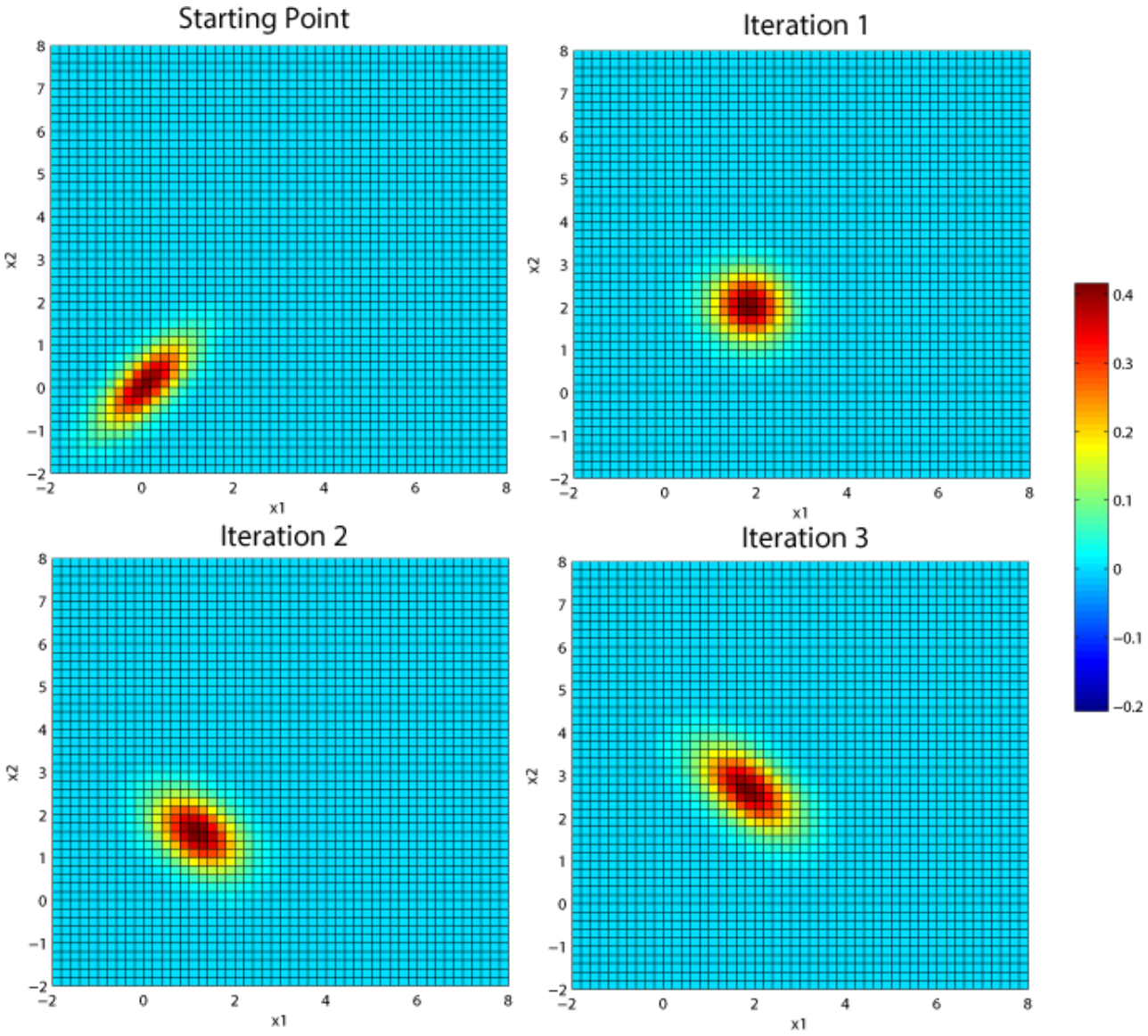


Figure 6: Likelihood contour plot for extended Kalman filter estimates.

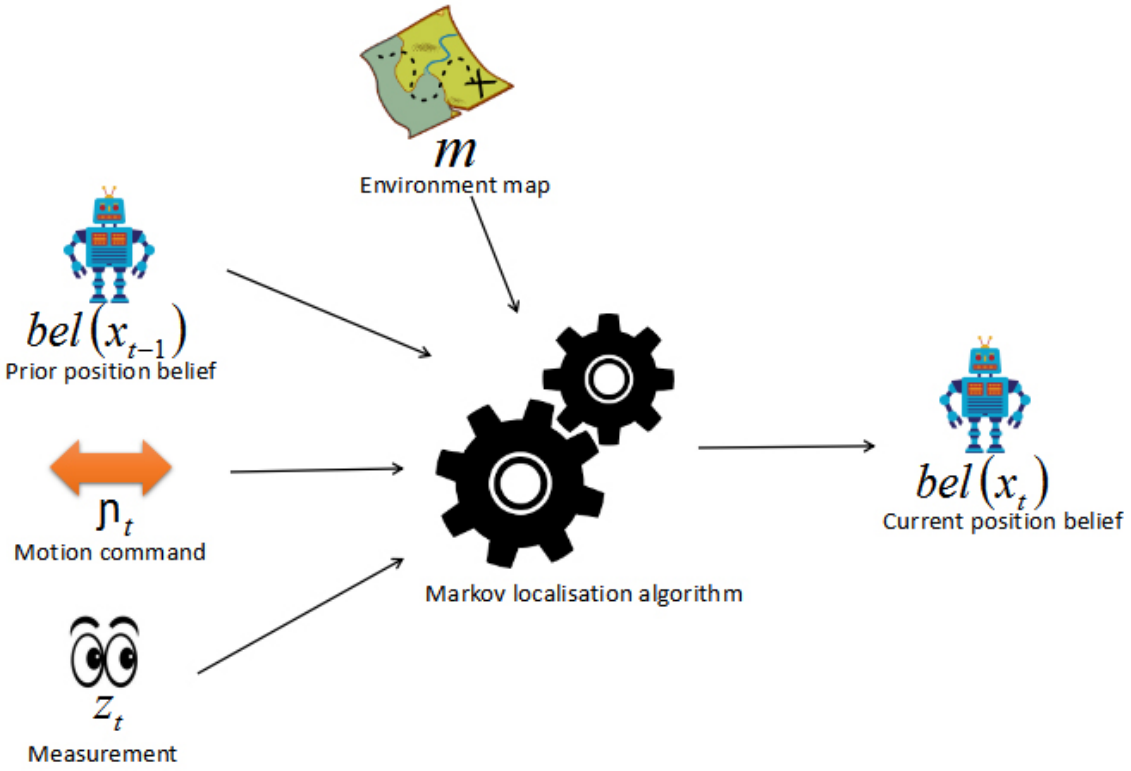


Figure 7: Markov localisation algorithm diagram

2. $\bar{bel}\,(x_t) = \sum_l P\,(x_t = l|x_{t-1}, \eta_t, m)\,bel\,(x_{t-1})$ (In this step the belief function is altered to consider possible position based on the motion command).
3. $bel\,(x_t) = \kappa P\,(observation|x_t = l, m)\,\bar{bel}\,(x_t)$ where κ is a normalization factor. (In this step the belief function is altered to consider possible position based on the measurements, e.g. sensor readings, and then normalised).

Return $bel\,(x_t)$.

Example
Suppose we have a corridor with two landmarks in which the robot can drive up and down as illustrated in Figure 8. For the purpose of this example we will also assume we know the robot has starting position 5 even though we will apply the algorithm as if it is a global localisation problem.
To initialise the algorithm we will assume $bel\,(x_0) = \frac{1}{10}$.
Now at time step 1 the robot moves one spot forward and does not observe a landmark; at time step 2 moves another step forward and observes a landmark; at time step 3 moves one spot backwards and does not observe a landmark; at time step 4 moves another step backwards and does not observe a landmark; at time step 5 moves another step backwards and does not observe a landmark and at time step 6 moves another step backwards and observes a landmark. The code for this application can be found in the appendix. The belief probability graphs are shown in Figure 9.

Figure 9 shows the bar plots of the belief funtion of the state of the robot as time progresses. As can be seen by assigning equal probability to all possible position initially and then applying the Markov localisation algorithm effectively finds the position of the robot.

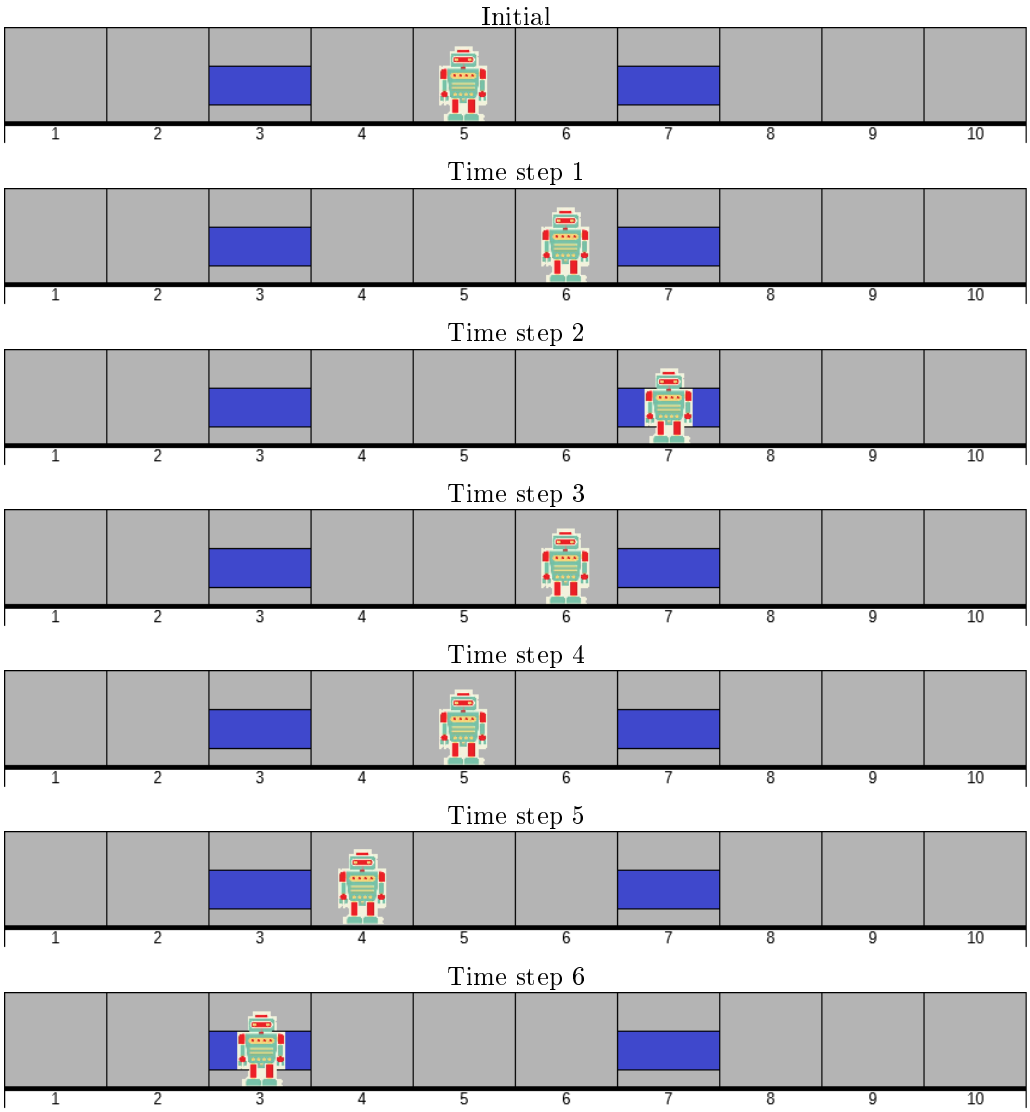


Figure 8: Markov localisation example illustration

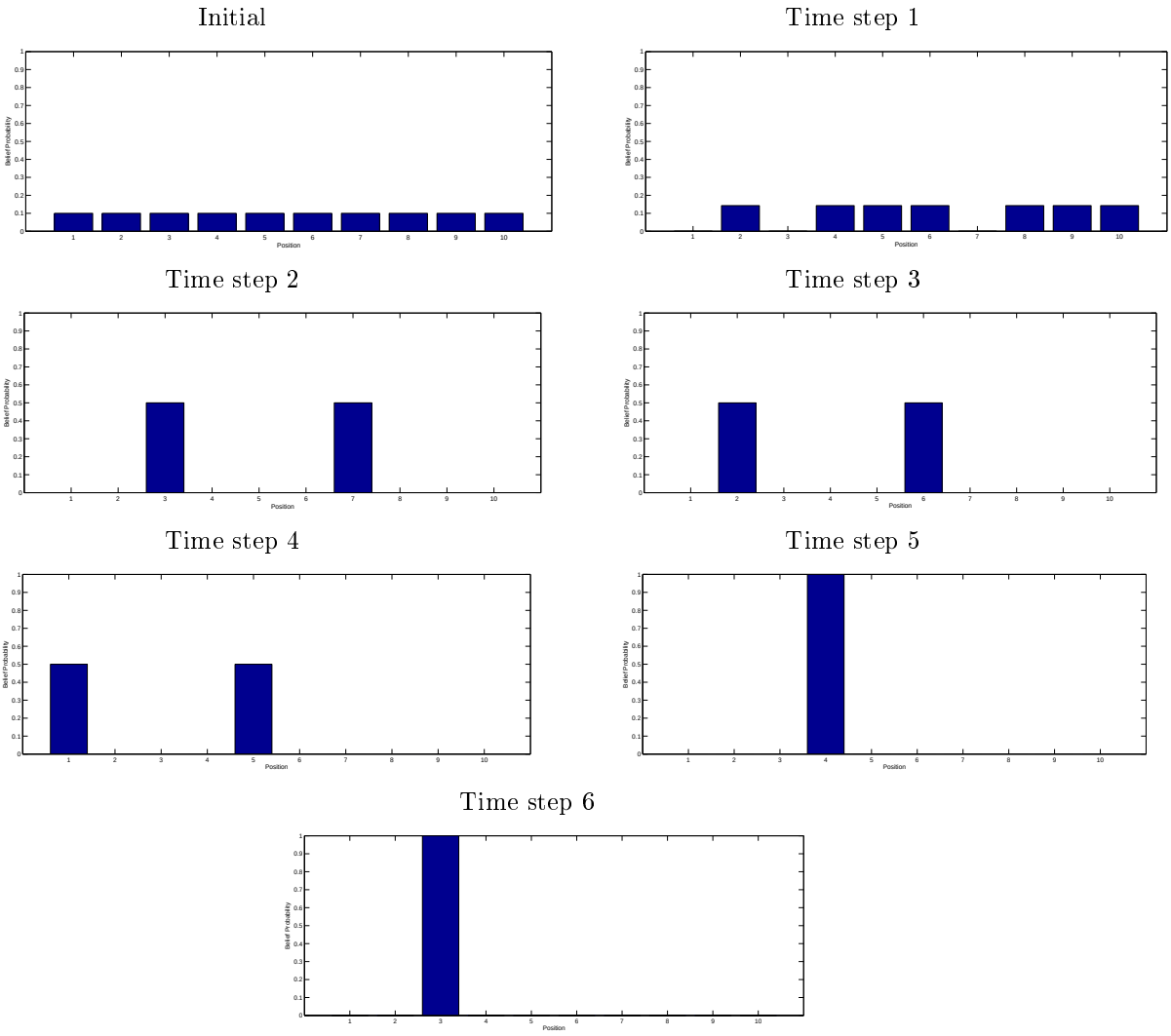


Figure 9: Markov localisation example belief function graphs

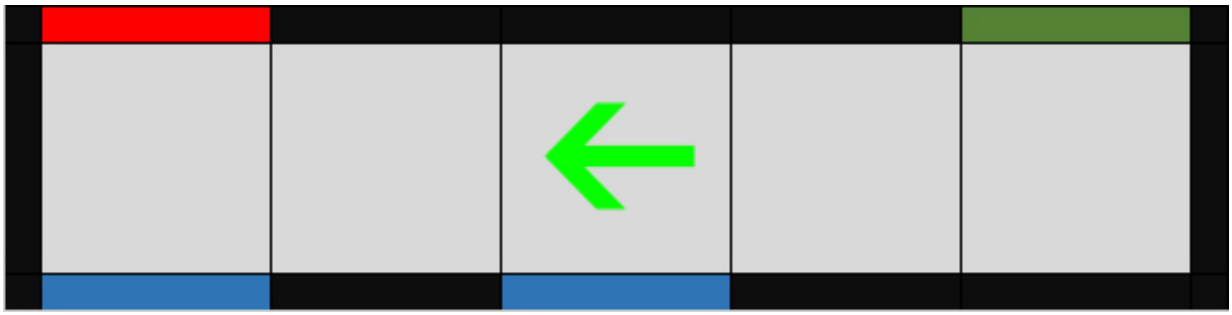


Figure 10: Map of corridor for application

6 Application

For the application section of this report we assemble a simple corridor with two doors similar to the situation in the Markov localisation example. We use the Markov localisation algorithm to find the position given that the EZRobot does not know its initial location. This application is a very simple maze navigation example which can be extended in future research.

In Figure 10 we see the map of the corridor that will be used for the application. For simplicity we will assume that the robot will always be facing towards the red end marker, as indicated by the arrow. The red and green markers are used as end markers for the robot to know when it has reached the end of the corridor. The blue markers are the doors in the corridor and these are used as landmarks in the robot's environment which it uses to localise.

Algorithm

1. Initialise the belief function for the Markov localisation algorithm. This is done by assuming that the probability of being in a block is uniformly distributed across the five blocks.
2. If this is the first iteration skip steps 3 and 4.
3. Look right and record the colour observed (This colour observed is not used for localisation but purely to know when the ends or the corridor is reached).
4. A random number is generated and depending on the random number the robot moves forwards or backwards and records the direction moved.
5. Look left and record whether a blue door is observed or not.
6. Apply the Markov localisation algorithm explained earlier.
7. Repeat steps 2 - 6 until we are $100\,(1-\alpha)\,\%$ certain of the location of the robot.
8. Say "I am in block *" (where block * is the block we are $100\,(1-\alpha)\,\%$ certain of being in).

Results

In Figure 11 we can see how the belief function converges after each iteration until we are $100\,(1-\alpha)\,\%$ certain of the location of the robot. Hence the robot has successfully localised and now knows where it is in the corridor.

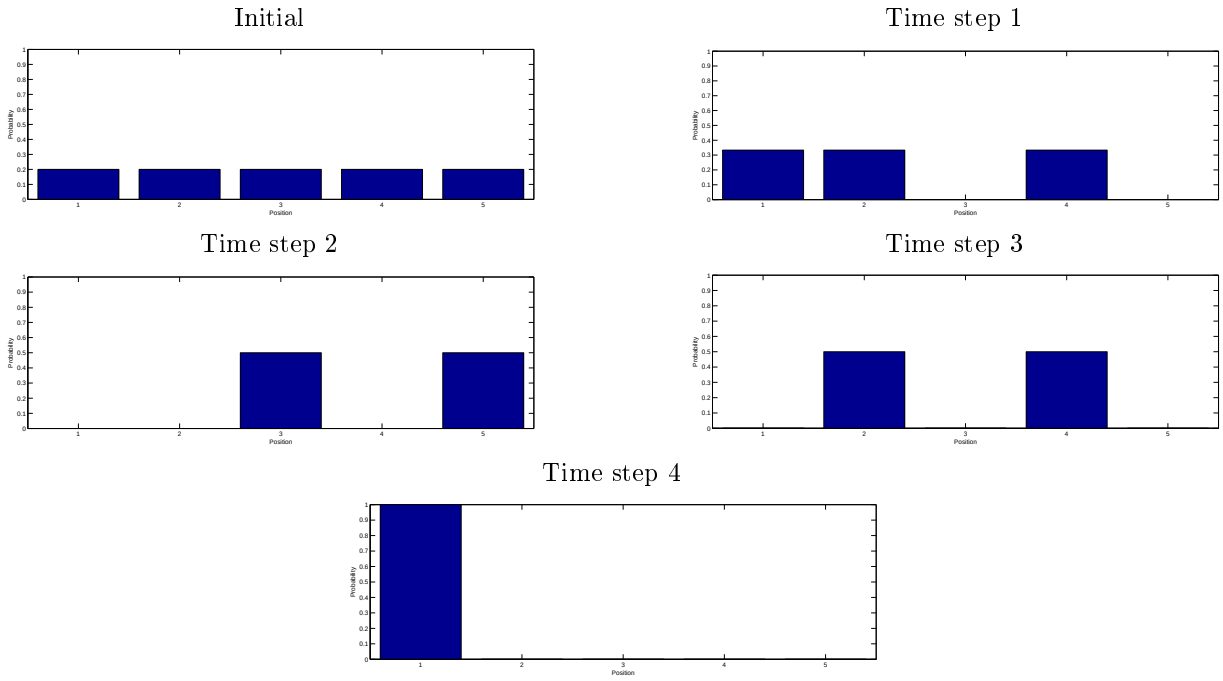


Figure 11: Belief function graphs for the application.

Conclusion

As seen above we successfully apply the Markov localisation algorithm to solve the localisation problem. Although some of the problems that we still experience in the application section are issues such as inconsistency in the distance and direction the robot walks as well as in the colour recognition process. The inconsitency with the colour recognition is resulting in the application with the robot only working occasionally since it does not always recognise colours correctly, most likely due to sensitivity of the robot's camera to lighting conditions. In further research we can look at applying algorithms such as the Kalman filter to remove these inconsistencies, and also look at extending the map to a 2 or even 3-dimensional map and investigate how efficiently the Markov localisation algorithm can be applied in these situations to help an autonomous robot localise and move closer to being artificially intelligent.

References

[1] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006.

[2] EZ-Robot Inc. *EZ-Builder Version 2016.08.11.00.* EZ-Robot Inc., Calgary, Alberta, 2016.

[3] David Filliat and Jean-Arcady Meyer. Map-based navigation in mobile robots:: I. a review of localization strategies. *Cognitive Systems Research*, 4(4):243–282, 2003.

[4] Dieter Fox. *Markov localization - A probabilistic framework for mobile robot localization and navigation.* PhD thesis, Institute of Computer Science III, University of Bonn, Germany, 1998.

[5] Leonie Freeston. Applications of the Kalman filter algorithm to robot localisation and world modelling. *Electrical engineering final year project. University of Newcastle, Australia*, 2002.

[6] Xiang Gao and Tao Zhang. Robust RGB-D simultaneous localization and mapping using planar point features. *Robotics and Autonomous Systems*, 72:1–14, 2015.

[7] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663, 2012.

[8] Robert A.R. King. Statistics for robotics: Kalman filter. In *Workshop at University of Pretoria*, 2015.

[9] John J Leonard and Hugh F Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991.

[10] Jean-Arcady Meyer and David Filliat. Map-based navigation in mobile robots:: Ii. a review of map-learning and path-planning strategies. *Cognitive Systems Research*, 4(4):283–317, 2003.

[11] Swati Mishra and Pankaj Bande. Maze solving algorithms for micro mouse. In *IEEE International Conference on Signal Image Technology and Internet Based Systems, 2008. SITIS'08.*, pages 86–93. IEEE, 2008.

[12] Rudy Negenborn. Robot localization and Kalman filters on finding your position in a noisy world. Master's thesis, Institute of Information and Computing Sciences, Utrecht University, 2003.

[13] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2016.

[14] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3–46, 2016.

[15] The MathWorks Inc. *MATLAB version 9.0 (R2016a).* The MathWorks Inc., Natick, Massachusetts, 2016.

[16] Sebastian Thrun. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, 1:1–35, 2002.

[17] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics.* The MIT Press, 2005.

[18] Fernando Tusell. Kalman filtering in R. *Journal of Statistical Software*, 39(2), March 2011.

Appendix

MATLAB [15] Code for Kalman filter function:

```
%Carel van Niekerk
%2016
%Kalman filter function
function [mu,Sig] = Kalman_filter(mu_p,Sig_p,Control,Measure,A,B,G,R,Q)
%mu_p is the mean of the belief function at previous time step
%Sig_p covariance structure of the belief function at previous time step
%Control is the current motion command
%Measure is the current measurement vector
%A and B form function x=A*x_p+B*control+e
%R is covariance structure of e
%G form function y=G*x+d
%Q is covariance structure of d
mu_b=A*mu_p+B*Control; %Mean of the belief function after motion command
Sig_b=A*Sig_p*(A.')+R; %Covariance structure of the belief function after motion command
K=(Sig_b*(G.'))/(G*Sig_b*(G.')+Q); %Kalman gain calculation
mu=mu_b+K*(Measure-G*mu_b); %Updated mean of the belief function at current time step
[n,m]=size(K);
Sig=(eye(n)-K*G)*Sig_b; %Updated covariance structure of the belief function at current time step
end
```

MATLAB [15] Code for function to plot pose of the robot:

```
%Carel van Niekerk
%2016
%Position plot function
function Plot_Position(vec,t,colour)
%vec is the 3 dimensional state vector
%t is the timestep number
%colour is the matlab colour code for the plot.
x = vec(1);
y = vec(2);
theta = vec(3);
L=0.2; %lenght of the direction line
xEnd = x+L*cos(theta); %calculation of direction line endpoints
yEnd = y+L*sin(theta);
hold on;
axis([x+[-2 2] y+[-2 2]]); %defining axis
axis equal;
plot([x xEnd],[y yEnd],'Color',colour); %plot direction line
plot(x,y,'*','Color',colour); %plot position
c=num2str(t);
dx = 0.1;
dy = 0.1;
text(x+dx, y+dy, c,'Color',colour); %add time step label
end
```

MATLAB [15] Code for Kalman filter example:

```
clear all;
n=3;
```

```
start=[0;0;0];
eta=[0;0;0];
A=eye(3);
B=eye(3);
G=eye(3);
R=eye(3);
Q=[0.81 0 0;
    0 0.73 0;
    0 0 0.68];
mu=[1;2;1.5708];
S=Q;
figure1=figure;
hold on;
c=[0 0 1];
Plot_Position(start,0,c);
for i=1:n
    y=mvnrnd(mu,S,1)';
    if i==1
        x=start;
        Sig=[0.4 0.3 0.34;
              0.3 0.45 0.28;
              0.34 0.28 0.41];
    end
    [x_new,Sig_new]=Kalman_filter(x,Sig,eta,y,A,B,G,R,Q);
    x=x_new;
    Sig=Sig_new;
    c=[0 0 1];
    Plot_Position(x,i,c);
    c=[1 0 0];
    Plot_Position(y,i,c);
end
pos=mu;
c=[0 1 0];
Plot_Position(pos,0,c);
clear all;
```

MATLAB [15] Code for the extended Kalman filter function for the example:

```
function [mu,Sig] = Ext_Kalman_Ex(mu_p,Sig_p,Control,Measure,R,Q)
mu_b=mu_p+Control;
G=eye(2);
Sig_b=G*Sig_p*G'+R;
x=mu_b(1,1);
y=mu_b(2,1);
val1=(x-8)/sqrt((x-8)^2 + (y-9)^2);
val2=(y-9)/sqrt((x-8)^2 + (y-9)^2);
v1=((8-x)*abs(8-x))/(((x-8)^2 + (y-9)^2)^(3/2));
v2=(8-x)/(sqrt((x-8)^2 + (y-9)^2)*abs(8-x));
v3=((8-x)^2)/((8-x)^2 + (9-y)^2);
v4=sqrt(1-v3);
val3=(v1-v2)/v4;
v5=((9-y)*abs(8-x))/(((x-8)^2 + (y-9)^2)^(3/2));
val4=v5/v4;
```

```
H=[val1 val2;
    val3 val4];
K=(Sig_b*H')/(H*Sig_b*H'+Q);
val5=sqrt((8-x)^2 + (9-y)^2);
v6=abs(8-x)/val5;
val6=asin(v6);
h_ut=[val5;val6];
mu=mu_b+K*(Measure-h_ut);
[n,m]=size(K);
Sig=(eye(n)-K*H)*Sig_b;
end
```

MATLAB [15] Code for function to plot contour plot for the pose of the robot:

```
function Contour_plot(x,Sig)
x1 = -2:.2:8; x2 = -2:.2:8;
[X1,X2] = meshgrid(x1,x2);
F = mvnpdf([X1(:) X2(:)],x,Sig);
F = reshape(F,length(x2),length(x1));
figure;
hold on;
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-2 8 -2 8 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Probability Density');
end
```

MATLAB [15] Code for the extended Kalman filter example:

```
clear all;
n=3;
R=0.2*eye(2);
Q=[0.4 0;
    0 0.52];
x0=[0;0];
Sig0=[0.4 0.3;
0.3 0.45];
Contour_plot(x0',Sig0);
eta=[0;0];
mu=[8.602325;0.620249];
S=Q;
for i=1:n
    if i==1
        xp=x0;
        Sigp=Sig0;
    else
        xp=xn;
        Sigp=Sign;
    end
    y=mvnrnd(mu,S,1)';
    [xn,Sign]=Ext_Kalman_Ex(xp,Sigp,eta,y,R,Q);
    Contour_plot(xn',Sign);
end
```

```
clear all;
```

MATLAB [15] Code for Markov localisation function:

```
function Bel = Markov_loc(map,A,Z,bel_pr)
zero=0;
if A==1
    bel_p=[zero,bel_pr(1:9)];
    Prob_matrix=[[zeros(9,1),eye(9)];zeros(1,10)];
else
    bel_p=[bel_pr(2:10),zero];
    Prob_matrix=[zeros(1,10);[eye(9),zeros(9,1)]];
end;
Bel_bar=zeros(1,10);
for i=1:10;
    S=trace(diag(Prob_matrix(:,i)));
    Bel_bar(i)=S*bel_p(i);
end;
objects=trace(diag(map(2,:)));
open=10-objects;
P=zeros(1,10);
if Z==1
    for l=1:10
        prob=1/objects;
        if map(2,l)==1
            P(l)=prob;
        end
    end
else
    for l=1:10
        prob=1/open;
        if map(2,l)==0
            P(l)=prob;
        end
    end
end
BelUN=zeros(1,10);
for j=1:10;
    BelUN(j)=Bel_bar(j)*P(j);
end;
eta=trace(diag(BelUN));
Bel=zeros(1,10);
for k=1:10;
    Bel(k)=BelUN(k)/eta;
end;
end
```

MATLAB [15] Code for Markov localisation example:

```
clear all;
map=[[1 2 3 4 5 6 7 8 9 10];
[0 0 1 0 0 0 1 0 0 0]];
bel_x0=[0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1];
figure;
```

```
bar(bel_x0);
Z=0;
A=1;
bel_x1=Markov_loc(map,A,Z,bel_x0);
figure;
bar(bel_x1);
Z=1;
A=1;
bel_x2=Markov_loc(map,A,Z,bel_x1);
figure;
bar(bel_x2);
Z=0;
A=-1;
bel_x3=Markov_loc(map,A,Z,bel_x2);
figure;
bar(bel_x3);
Z=0;
A=-1;
bel_x4=Markov_loc(map,A,Z,bel_x3);
figure;
bar(bel_x4);
Z=0;
A=-1;
bel_x5=Markov_loc(map,A,Z,bel_x4);
figure;
bar(bel_x5);
Z=1;
A=-1;
bel_x6=Markov_loc(map,A,Z,bel_x5);
figure;
bar(bel_x6);
clear all;
```

EZ-Script [2] Code for the Markov localisation algorithm:

```
$ind0 = $n_blocks - 1

DefineArray($Temp,$n_blocks)

REPEAT ($i,0,$ind0,1)
  $Temp[$i] = $Belief[$i]
  $i++
ENDREPEAT

IF ($Dir = 1)
  $Belief[0] = 0
  REPEAT ($i,1,$ind0,1)
    $ind = $i - 1
    $Belief[$i] = $Temp[$ind]
  ENDREPEAT
ENDIF

$ind1 = $n_blocks - 2
```

```
IF ($Dir = -1)
  $Belief[4] = 0
  REPEAT ($i,0,$ind1,1)
    $ind = $i + 1
    $Belief[$i] = $Temp[$ind]
  ENDREPEAT
ENDIF

DefineArray($Prob,$n_blocks)
IF ($Obs = 0)
  $Prob[0] = 1/3
  $Prob[1] = 1/3
  $Prob[2] = 0
  $Prob[3] = 1/3
  $Prob[4] = 0
ENDIF
IF ($Obs = 1)
  $Prob[0] = 0
  $Prob[1] = 0
  $Prob[2] = 0.5
  $Prob[3] = 0
  $Prob[4] = 0.5
ENDIF

DefineArray($Bel_bar,$n_blocks)
REPEAT ($i,0,$ind0,1)
  $Bel_bar[$i] = $Prob[$i] * $Belief[$i]
ENDREPEAT

$Sum = $Bel_bar[0] + $Bel_bar[1] + $Bel_bar[2] + $Bel_bar[3] + $Bel_bar[4]

DefineArray($Belief_new,$n_blocks)
REPEAT ($i,0,$ind0,1)
  $Belief_new[$i] = $Bel_bar[$i] / $Sum
ENDREPEAT

REPEAT ($i,0,$ind0,1)
  $Belief[$i] = $Belief_new[$i]
ENDREPEAT
```

EZ-Script [2] Code to initialise the belief function:

```
DefineArray($Belief,$n_blocks)
$indDEF = $n_blocks - 1
REPEAT ($i,0,$indDEF,1)
  $Belief[$i] = 1 / $n_blocks
ENDREPEAT
```

EZ-Script [2] Code to find the maximum of the belief function:

```
$max = 0
$mpos = 0
$ind0 = 4
```

```
REPEAT ($i,0,$ind0,1)
  IF ($Belief[$i] > $max)
    $max = $Belief[$i]
    $mpos = $i + 1
  ENDIF
ENDREPEAT
```

EZ-Script [2] Code for random movement:

```
$u=GetRandom(0,10000)/10000
IF ($u >= 0.5)
  IF ($CameraObjectColor="Red")
    ControlCommand("Auto Position", AutoPositionAction, "Walk Back 1")
    $Dir = -1
  ELSE
    ControlCommand("Auto Position", AutoPositionAction, "Walk 1 Block")
    $Dir = 1
  ENDIF
ENDIF
IF ($u < 0.5)
  IF ($CameraObjectColor="Green")
    ControlCommand("Auto Position", AutoPositionAction, "Walk 1 Block")
    $Dir = 1
  ELSE
    ControlCommand("Auto Position", AutoPositionAction, "Walk Back 1")
    $Dir = -1
  ENDIF
ENDIF
```

EZ-Script [2] Code for the application of the Markov localisation algorithm:

```
$err = 0.90

$n_blocks = 5
ControlCommand("Initialise Belief Function", ScriptStart)

$max_prob = 0
$position = 0
$int0 = 5000
$k = 0
$int1 = 20000
repeatuntil($max_prob > $err)
  if ($k > 0)
    Servo(D0,0)
    Sleep($int0)
    $int = 2000
    $CameraObjectColor = ""
    ControlCommand("Camera", CameraMultiColorTrackingEnable)
    Sleep($int)
    ControlCommand("Camera", CameraMultiColorTrackingDisable)
    Servo(D0,90)
    Sleep($int0)
    ControlCommand("Random Movement", ScriptStart)
    sleep($int1)
```

```
  endif

  Servo(D0,180)
  Sleep($int0)
  $int = 2000
  $CameraObjectColor = ""
  ControlCommand("Camera", CameraMultiColorTrackingEnable)
  Sleep($int)
  ControlCommand("Camera", CameraMultiColorTrackingDisable)
  Servo(D0,90)
  Sleep($int0)
  if ($CameraObjectColor = "Blue")
    $Obs = 1
  ELSE
    $Obs = 0
  endif
  if ($k = 0)
    $Dir = 0
  endif

  ControlCommand("Markov Localisation Algorithm", ScriptStart)
  Sleep(2000)
  ControlCommand("Find Maximum", ScriptStart)

  $max_prob = $max
  $position = $mpos

  $k++
endrepeatuntil

SayEZB("I am at block " + $position)
```

Performance evaluation of Suzuki-type channels

Marcus Warriner 12094422

WST795 Research Report

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

Supervisor: Johan Ferreira, co-supervisor: Prof Andriëtte Bekker

Department of Statistics, University of Pretoria

2 November 2016

Abstract

This study focuses on the construction of composite fading/shadowing distributions. A group of distributions all belonging to the regular exponential class are systematically described and its construction in the composite paradigm is motivated. Statistical properties and results are derived, including the probability density function, moment generating function, and moments. Novel contributions to the wireless communications arena are presented with these new distributions. The literature is further enriched with derivation of the corresponding signal-to-noise ratio distributions for each of the newly derived distributions. These distributions are comparatively investigated in terms of their outage probability.

Declaration

I, *Marcus Warriner,* 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.

_____
Marcus Warriner

_____
Johan Ferreira

_____
Prof Andriëtte Bekker

_____
Date

Acknowledgements

I would like to thank the National Research Foundation for the grantholder-linked student assistantship, ref. CPRR13090132066 grant no. 91497

I would also like to thank Johan Ferreira and Professor Andriëtte Bekker for their valuable guidance and continuous support throughout the year.

Contents

1 Introduction **8**

2 Composite fading/shadowing distributions **9**
2.1 Fading distributions . . . 9
2.1.1 Nakagami-q . . . 9
2.1.2 Nakagami-n . . . 9
2.1.3 Nakagami-m . . . 9
2.1.4 Rayleigh . . . 9
2.2 Shadowing distributions . . . 10
2.2.1 Lognormal . . . 10
2.3 Composite fading/shadowing distributions . . . 10
2.3.1 Suzuki distribution . . . 10
2.3.2 Other composite distributions . . . 10

3 Derivation of compound distributions **11**

4 Fading distributions **12**
4.1 Fading distributions from the exponential class . . . 12
4.1.1 Exponential distribution . . . 12
4.1.2 Gamma distribution . . . 13
4.1.3 Rayleigh distribution . . . 14
4.1.4 Weibull distribution . . . 15
4.1.5 Nakagami-m distribution . . . 16
4.2 Compound fading distributions . . . 17
4.2.1 Compound-Weibull fading . . . 18
4.2.2 Compound-Rayleigh fading . . . 20
4.3 Relationship between fading distributions . . . 22
4.3.1 Gamma to exponential transformation . . . 22
4.3.2 Rayleigh to exponential transformation . . . 22
4.3.3 Weibull to exponential transformation . . . 23
4.3.4 Nakagami-m to exponential transformation . . . 23
4.3.5 Compound-Weibull to compound-Rayleigh . . . 23

5 Lognormal shadowing **24**

6 Composite distributions: fading in a lognormally shadowed environment **25**
6.1 Exponential fading with lognormal shadowing . . . 25
6.1.1 PDF of the composite exponential/lognormal distribution . . . 25
6.1.2 MGF of the composite exponential/lognormal distribution . . . 26
6.1.3 Moments of the composite exponential/lognormal distribution . . . 26
6.2 Gamma fading with lognormal shadowing . . . 27
6.2.1 PDF of the composite gamma/lognormal distribution . . . 27
6.2.2 MGF of the composite gamma/lognormal distribution . . . 29
6.2.3 Moments of the composite gamma/lognormal distribution . . . 29
6.3 Rayleigh fading with lognormal shadowing . . . 30
6.3.1 PDF of the composite Rayleigh/lognormal distribution . . . 30
6.3.2 MGF of the composite Rayleigh/lognormal distribution . . . 31
6.3.3 Moments of the composite Rayleigh/lognormal distribution . . . 31
6.4 Weibull fading with lognormal shadowing . . . 32
6.4.1 PDF of the composite Weibull/lognormal distribution . . . 32
6.4.2 MGF of the composite Weibull/lognormal distribution . . . 34
6.4.3 Moments of the composite Weibull/lognormal distribution . . . 34

6.5 Nakagami-m fading with lognormal shadowing . . . 35
6.5.1 PDF of the composite Nakagami-m/lognormal distribution . . . 35
6.5.2 MGF of the composite Nakagami-m/lognormal distribution . . . 36
6.5.3 Moments of the composite Nakagami-m/lognormal distribution . . . 37
6.6 Compound-Weibull fading with lognormal shadowing . . . 37
6.6.1 PDF of the composite compound-Weibull/lognormal distribution . . . 37
6.6.2 MGF of the composite compound-Weibull/lognormal distribution . . . 39
6.6.3 Moments of the composite compound-Weibull/lognormal distribution . . . 40
6.7 Compound-Rayleigh fading with lognormal shadowing . . . 40
6.7.1 PDF of the composite compound-Rayleigh/lognormal distribution . . . 40
6.7.2 MGF of the composite compound-Rayleigh/lognormal distribution . . . 42
6.7.3 Moments of the composite compound-Rayleigh/lognormal distribution . . . 42

7 Application to wireless communication systems 43
7.1 Amplitude of of composite fading/shadowing channels . . . 43
7.2 Signal-to-noise ratio of composite fading/shadowing channels . . . 43
7.2.1 Appropriateness of SNR distribution . . . 43
7.2.2 Mean SNR ($\bar{\omega}$) distribution . . . 43
7.2.3 SNR distribution of the composite Rayleigh/lognormal channel . . . 45
7.2.4 SNR distribution of the composite Nakagami-m/lognormal channel . . . 46
7.2.5 SNR distribution of the composite compound-Rayleigh/lognormal channel . . . 48
7.3 Outage probability (P_{out}) . . . 50
7.3.1 P_{out} of the composite Rayleigh/lognormal channel (7.8) . . . 50
7.3.2 P_{out} of the composite Nakagami-m/lognormal channel (7.10) . . . 51
7.3.3 P_{out} of the composite compound-Rayleigh/lognormal channel (7.12) . . . 53
7.3.4 Appropriateness of the compound-Rayleigh distribution to assess outage probability . 54

8 Conclusion 55

Appendix 57

Abbreviations

PDF: probability density function
CDF: cumulative distribution function
MGF: moment generating function
SNR: signal-to-noise ratio
BEP: bit error probability

Notation

X: random variable
$X|\lambda$: conditional random variable X given λ
$f(x)$: probability density function of X
$f(x|\lambda)$: conditional probability density function of $X|\lambda$
$M_X(t)$: moment generating function of X
$M_{X|\lambda}(t)$: moment generating function of $X|\lambda$
m_r: r^{th} moment of X
$m_r|\lambda$: r^{th} moment of $X|\lambda$
$\mathbb{E}[X]$: expected value of X
$var(X)$: variance of X
$\mathbb{Z}^+$: set of positive integers

Concepts

Fading

Multipath fading is an effect caused by the combination of signal paths which have been randomly delayed, reflected, scattered and/or diffracted. This type of fading results in short-term signal variations, which cause fluctuations in wave size and position over time. This can have a negative impact on performance of the wireless communication system, unless the signal receiver has taking these measures into account [13].

Shadowing

The quality of the link between mobile systems is affected by the mean signal level. Shadowing is a result of obstacles in the signal path such as terrain, buildings and trees, which cause a slow variation in the mean signal level. Performance of communication systems will only depend on shadowing only if the radio receiver is able to average out the fast multipath fading or if the effects of multipath fading have been removed by an efficient transmitter/receiver wireless network system [13].

Signal-to-noise ratio

Random variable that represents the ratio of signal strength to amount of noise at receiver output. The term noise relates to the electrical fluctuation present from input to receiver. SNR is the best measure in the context of performance evaluation of a wireless network, indicating overall accuracy of system. In the fading context, average SNR is a more appropriate measure [13].

Outage probability

Defined as the probability of an error in transmission of information, which occurs when the SNR at the receiver output falls below a protection threshold. This failure to meet the target SNR is due to signal interference and results in insufficient performance over the wireless network system [11, 13, 18, 19].

1 Introduction

In wireless communications theory, statistical distributions are used to model channels between senders and receivers. A transmitted signal is exposed to two types of interference, namely multipath fading and shadowing, both of which are random in nature. This results in a stochastic approach to modelling wireless systems, where fading follows some distribution, e.g. Rayleigh fading, and in a shadowed environment, the lognormal distribution is used. Combining the effects of fading and shadowing on a signal results in a composite fading/shadowing distribution [13]. In section 2, fading distributions and lognormal shadowing proposed in literature are explained. An example of a composite distribution proposed in literature, and the main theoretical focus of this project, is the Suzuki distribution introduced by Hirofumi Suzuki in 1977 as a statistical model for urban radio propagation [15]. The Suzuki channel consists of Rayleigh fading in a lognormally shadowed wireless signal environment. Simply, the Suzuki distribution is derived by compounding the Rayleigh distribution with the lognormal distribution (illustrated in figure 1.1), i.e. if $X|\lambda \sim Rayleigh\,(\lambda)$ and $\lambda \sim lognormal\left(\mu, \sigma^2\right)$, then $X \sim Suzuki\,(\mu, \sigma)$. The theory of compound distributions is explained in section 3.

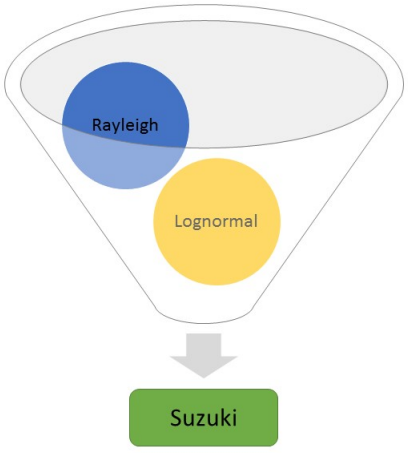


Figure 1.1: Components of the Suzuki distribution

The Rayleigh distribution, like all proposed fading distributions, is a member of the exponential family of distributions. In section 4, the Rayleigh and other distributions related to the exponential class, already proposed in literature, as well as two new distributions - compound-Weibull and compound-Rayleigh - are investigated as fading distributions. In section 5, the lognormal distribution as a shadowing distribution and its properties are briefly discussed. Investigating further into composite distributions, section 6 derives the PDF, MGF and moments of several distributions within the exponential class. Finally, section 7 applies this theory to performance evaluation of wireless communication systems using outage probability, which is obtained by deriving the CDF of the SNR distribution for the composite channel [13].

2 Composite fading/shadowing distributions

In this section, fading, shadowing and composite distributions relevant to this study are briefly described.

2.1 Fading distributions

2.1.1 Nakagami-q

The Nakagami-q distribution is based on the fading parameter q, with range $0 \leq q \leq 1$. A special case of this distribution, when $q = 0$, is known as the one-sided Gaussian fading distribution, which has the largest amount of fading out of all the multipath fading distributions. Another is when $q = 1$, this is the same as the Rayleigh fading distribution. Nakagami-q fading is normally observed on satellite links, contingent on signal fluctuation as it passes through the ionosphere (portion of the Earth's upper atmosphere) [13].

2.1.2 Nakagami-n

The Nakagami-n distribution is based on the fading parameter n, with range $0 \leq n < \infty$, related to the Rician K factor $k = n^2$. Special cases of this distribution: the Rayleigh distribution ($n = 0$) and there is no fading when $n = \infty$. Nakagami-n fading is frequently observed in land-mobile systems, small to large indoor environments, satellite links and radio communication between ships. It is used to model propagation paths which consist of the following components: strong direct line-of-sight and many random weaker ones [13].

2.1.3 Nakagami-m

The Nakagami-m distribution is based on the fading parameter m, with range $m \geq \frac{1}{2}$, and has the widest amount of fading over all multipath fading distributions. The special cases of this type of fading are: the one-sided Gaussian distribution $\left(m = \frac{1}{2}\right)$ and the Rayleigh distribution ($m = 1$). Nakagami-m fading is used for land-mobile communications, multipath propagation in indoor-mobile environments and is observed in ionospheric radio wave fluctuation [13].

Remark: There is a one-to-one relationship between q, n, and m, given by [13]:

$$m = \frac{\left(1+q^2\right)^2}{2\left(1+2q^4\right)}, \ m \leq 1,$$

$$m = \frac{\left(1+n^2\right)^2}{1+2n^2}, \ n \geq 0$$

and

$$n = \sqrt{\frac{\sqrt{m^2-m}}{m-\sqrt{m^2-m}}}, \ m \geq 1.$$

.

2.1.4 Rayleigh

The Rayleigh distribution is used in wireless communication to model multipath fading where there is no direct line-of-sight path [13], i.e. there are obstacles in the transmission path between transmitter and receiver. The Rayleigh distribution measures the path strength at any delay in signal transmission [15]. Empirically, this distribution fits experimental data for mobile systems. It can also be applied to the propagation of reflected and refracted paths through the troposphere (lowest layer of Earth's atmosphere) and ionosphere and radio links between ships [13, 15]. In Suzuki's original work, the fading component proposed is the Rayleigh distribution [15].

2.2 Shadowing distributions

2.2.1 Lognormal

Empirically, it is assumed that the lognormal distribution can be used to model shadowing for various outdoor and indoor environments; the instantaneous SNR is lognormally distributed [13].

2.3 Composite fading/shadowing distributions

Composite multipath/shadowed fading comprises of multipath fading layed over lognormal shadowing. In this channel, multipath fading is not averaged out by the signal receiver, instead it reacts to the instantaneous composite multipath/shadowed signal. Examples of this type of composite fading are; congested urban areas with slow moving pedestrians and vehicles, or mobile systems in urban areas where shadowing is a result of trees and buildings [13].

There are several applications of composite distributions, which fall under wireless communication system performance measures [13]:

- Outage probability;
- Average bit error probability; and
- Average combined output SNR.

2.3.1 Suzuki distribution

The Suzuki distribution is a composite multipath/shadowed fading distribution which consists of Rayleigh fading and lognormal shadowing [20]. This distribution was developed to model radio propagation in urban areas. The distribution is based on experimental data which measured the path strength and path arrival time. Path strength mean and variance were analysed using the path strength data in small geographical areas and distributions were fitted: Rayleigh, Nakagami-m, Nakagami-n and lognormal [15, 16].

The Rayleigh distribution did not fit the data as well as expected, this is because this distribution only has one parameter. The Nakagami-m and Nakagami-n distributions include the Rayleigh distribution in special cases and therefore fit the data better. The next best was the Nakagami-n distribution which fit most of the data and failed to fit the rest of the data. The Nakagami-m is the preferred distribution as it approximates the Nakagami-n distribution and fits the data well. The lognormal distribution fit the data the best in all cases. In conclusion it was established that path strength distribution is that of a Nakagami-m at initial paths and lognormal at increasing excess delay. Mean path strength decreases as excess delay increases and path strength variance is almost constant [15].

2.3.2 Other composite distributions

Fading	**Shadowing**	**Reference**
Nakagami-m	lognormal	[6, 13]
gamma	lognormal	[12]
Weibull	lognormal	[8]
Rayleigh/Nakagami-m	gamma	[2]
gamma	gamma	[1]
$\alpha-\mu$, $\kappa-\mu$, $\eta-\mu$, $\lambda-\mu$	gamma	[14]

Table 2.1: A few examples of composite channels in literature

3 Derivation of compound distributions

This section describes the methodology used to obtain composite distributions.

In general, a compound distribution is one where X has some distribution with parameter λ and this parameter has its own distribution. $F(x|\lambda)$ is the conditional distribution function of $X|\lambda$ and $G(\lambda)$ is the distribution function of λ, with the following PDFs, respectively:

$$\begin{aligned} f(x|\lambda) &: \text{PDF of the } \textit{conditional} \text{ distribution of } X|\lambda; \text{ and} \\ g(\lambda) &: \text{PDF of the distribution of } \lambda. \end{aligned}$$

The formula for the PDF of the compound distribution of X, $h(x)$, is the compound of $f(x|\lambda)$ and $g(\lambda)$ over all values of λ and is given by [7]:

$$h(x) = \int_{\lambda} f(x|\lambda)\, g(\lambda)\, d\lambda \tag{3.1}$$

with properties

$$\mathbb{E}[X] = \mathbb{E}_{\lambda}[\mathbb{E}[X|\lambda]] \tag{3.2}$$

and

$$var(X) = \mathbb{E}_{\lambda}[var(X|\lambda)] + var_{\lambda}(\mathbb{E}[X|\lambda]). \tag{3.3}$$

Composite fading/shadowing distributions are derived through the compound distribution process, where:

$$\begin{aligned} f(x|\lambda) &: \text{PDF of the } \textit{fading} \text{ distribution of } X|\lambda; \text{ and} \\ g(\lambda) &: \text{PDF of the } \textit{shadowing} \text{ distribution of } \lambda. \end{aligned}$$

This means that multipath fading in a shadowed environment has compound distribution with PDF $h(x)$. Equation (3.1) is used to find the expected value of any function of random variable X in the following theorem. The importance of this theorem is that other functions related to the compound distribution can be derived, namely the MGF and the moments. The following theorem provides a formula with which some important theoretical results will be obtained, namely the MGF and moments.

Theorem 1

Let X be a random variable with PDF $h(x)$ and let $X|\lambda$ be the conditional random variable with PDF $f(X|\lambda)$ where λ has PDF $g(\lambda)$. Let $z(x)$ be any Borel measurable function of X. Then

$$\mathbb{E}[z(X)] = \int_{\lambda} \mathbb{E}[z(X|\lambda)]\, g(\lambda)\, d\lambda \tag{3.4}$$

Proof. The expected value of $z(x)$ is

$$\begin{aligned} \mathbb{E}[z(X)] &= \int_{x} z(x)\, h(x)\, dx \\ &= \int_{x} z(x) \int_{\lambda} f(x|\lambda)\, g(\lambda)\, d\lambda dx \text{ by equation 3.1} \\ &= \int_{\lambda} \left[\int_{x} z(x)\, f(x|\lambda)\, dx \right] g(\lambda)\, d\lambda \\ &= \int_{\lambda} \mathbb{E}[z(X|\lambda)]\, g(\lambda)\, d\lambda \end{aligned}$$

which proves the result. ■

By Theorem 1, the MGF and r^{th} moment of the compound distribution of X are derived.

If $z(x) = \exp(tx)$, then

$$M_X(t) = \int_\lambda M_{X|\lambda}(t)\, g(\lambda)\, d\lambda \tag{3.5}$$

If $z(x) = x^r$, then

$$m_r = \int_\lambda m_{r|\lambda} g(\lambda)\, d\lambda \tag{3.6}$$

respectively.

4 Fading distributions

In this section, the PDF, MGF and moments of each fading distribution that will be considered in this study, as well as a graphical illustration of the PDF and CDF.

4.1 Fading distributions from the exponential class

The distributions considered in this section are all contained within the exponential class of distributions.

4.1.1 Exponential distribution

Let $X \sim exponential(\lambda)$ with *scale* parameter $\lambda > 0$, the PDF is given by [7]:

$$f(x) = \frac{1}{\lambda} \exp\left(-\frac{x}{\lambda}\right), \quad x \geq 0. \tag{4.1}$$

Figure 4.1 shows the PDF and CDF with different parameter values.[1]

The MGF is defined as [7]:

$$M_X(t) = \frac{\lambda}{\lambda - t}, \quad t < \lambda. \tag{4.2}$$

The r^{th} moment is defined as [7]:

$$m_r = r!\lambda^r, \quad r \in \mathbb{Z}^+. \tag{4.3}$$

[1] The analysis for this essay was performed using SAS software, Version 9.4 of the SAS System for Windows. Copyright © 2016 SAS Institute Inc., Cary, NC, USA.

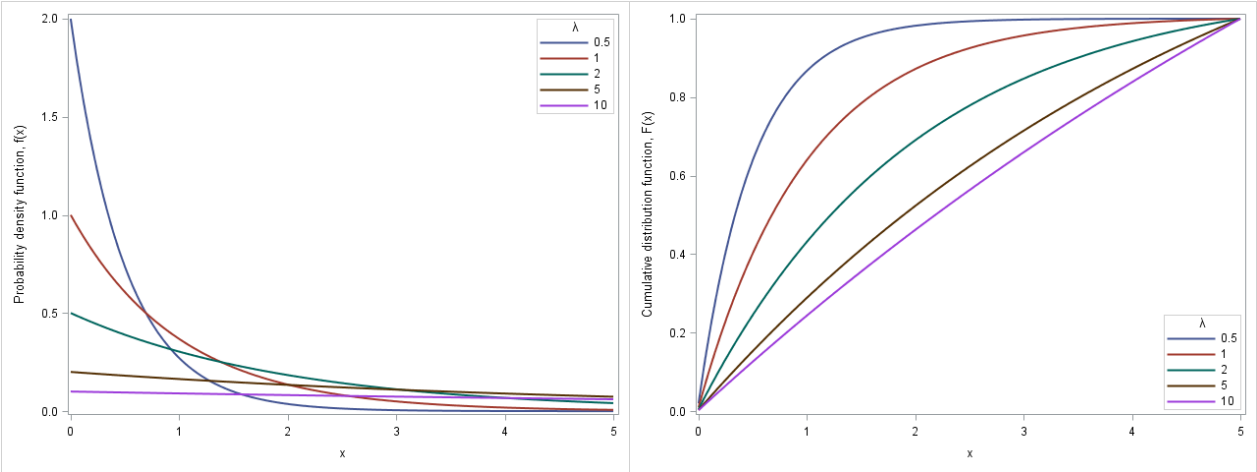


Figure 4.1: PDF (4.1) and CDF, respectively, with varying λ

4.1.2 Gamma distribution

Let $X \sim gamma\,(k, \lambda)$ with *shape* and *scale* parameters $k > 0$ and $\lambda > 0$, respectively, the PDF is given by [7]:

$$f\,(x) \quad = \quad \frac{1}{\Gamma\,(k)\,\lambda^k} x^{k-1} \exp\left(-\frac{x}{\lambda}\right), \quad x > 0. \tag{4.4}$$

Figures 4.2 and 4.3 show the PDF and CDF with different parameter values.

The MGF is defined as [7]:

$$M_X\,(t) \quad = \quad (1 - \lambda t)^{-k}, \quad t < \frac{1}{\lambda}. \tag{4.5}$$

The r^{th} moment is defined as [7]:

$$m_r \quad = \quad \frac{\lambda^r \Gamma\,(k + r)}{\Gamma\,(k)}, \quad r \in \mathbb{Z}^+. \tag{4.6}$$

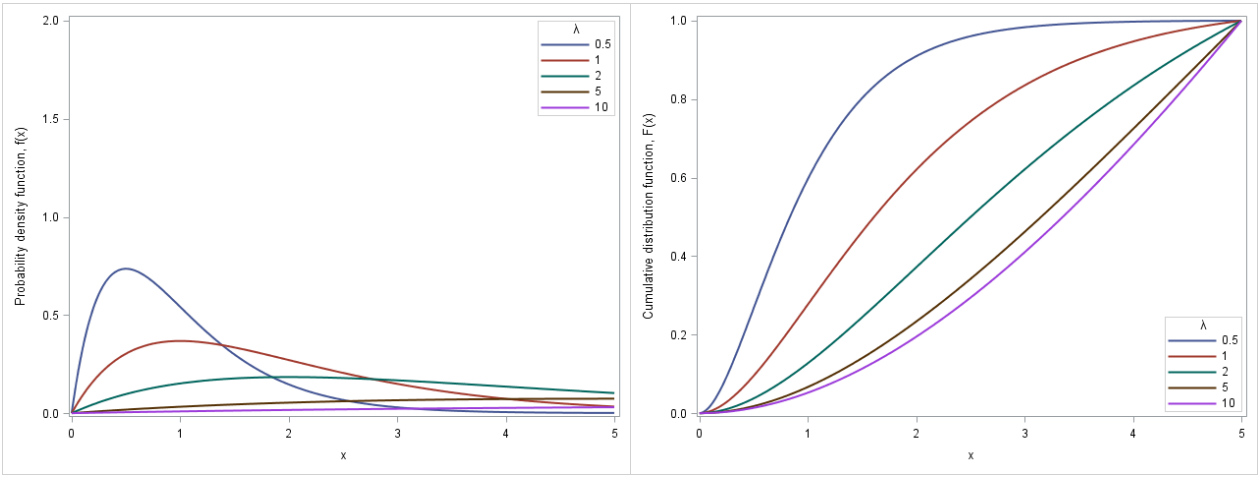


Figure 4.2: PDF (4.4) and CDF, respectively, with $k = 2$ and varying λ

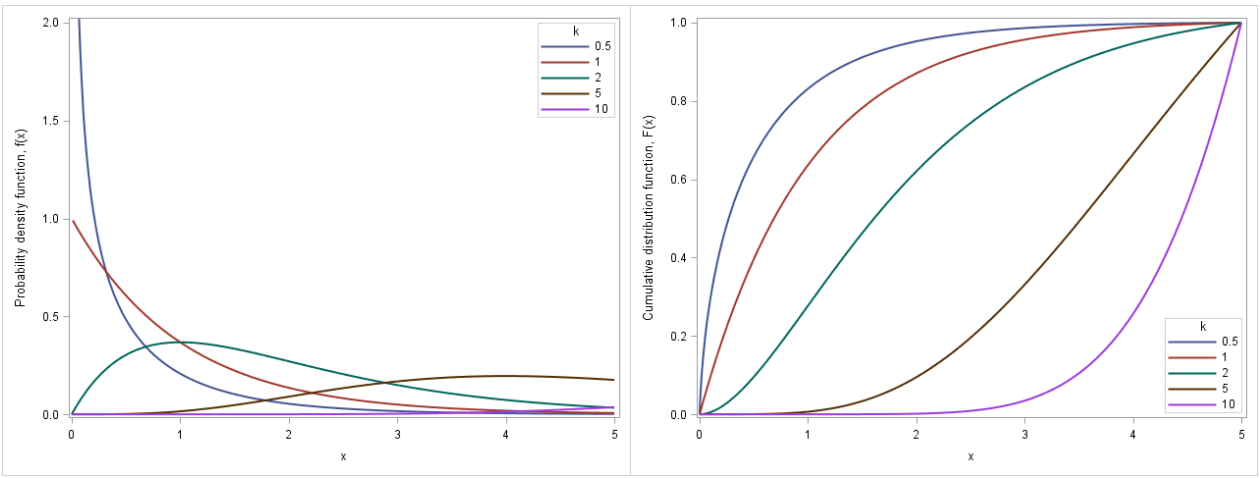


Figure 4.3: PDF (4.4) and CDF, respectively, with $\lambda = 1$ and varying k

4.1.3 Rayleigh distribution

Let $X \sim Rayleigh\,(\lambda)$ with *scale* parameter $\lambda > 0$, the PDF is given by [13]:

$$f\,(x) \quad = \quad \frac{2x}{\lambda} \exp\left(-\frac{x^2}{\lambda}\right), \;\; x \geq 0. \tag{4.7}$$

Figure 4.4 shows the PDF and CDF with different parameter values.

The r^{th} moment is defined as [7]:

$$m_r \quad = \quad \lambda^{\frac{r}{2}} \Gamma\left(1 + \frac{r}{2}\right), \;\; r \in \mathbb{Z}^+. \tag{4.8}$$

The MGF is defined as [7]:

$$M_X\,(t) \quad = \quad 1 + \frac{\sqrt{\pi\lambda}t}{2} \exp\left(\frac{\lambda t^2}{4}\right)\left[1 + \operatorname{erf}\left(\frac{\sqrt{\lambda}t}{2}\right)\right], \;\; t \geq 0,$$

which contains the error function (see Appendix) which may lead to calculation challenges. Alternatively, the MGF is derived by substituting the r^{th} moment (4.8) into the definition of a MGF in [3] and by the exponential Taylor series (see Appendix):

$$\begin{aligned} M_X(t) &= \mathbb{E}\left[\exp(tX)\right] \\ &= \mathbb{E}\left[\sum_{i=0}^{\infty} \frac{(tx)^i}{i!}\right] \\ &= \sum_{i=0}^{\infty} \frac{t^i}{i!}\mathbb{E}\left[X^i\right] \\ &= \sum_{i=0}^{\infty} \frac{t^i \lambda^{\frac{i}{2}} \Gamma\left(1+\frac{i}{2}\right)}{i!}, \quad t \geq 0. \end{aligned} \tag{4.9}$$

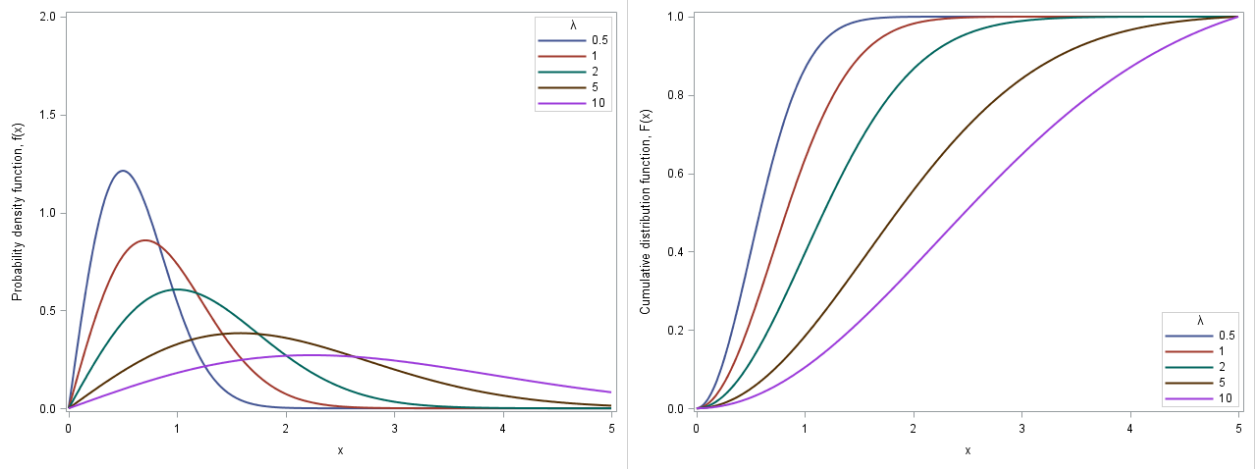


Figure 4.4: PDF (4.7) and CDF, respectively, with varying λ

4.1.4 Weibull distribution

Let $X \sim Weibull(\alpha, \lambda)$ with *shape* and *scale* parameters $\alpha > 0$ and $\lambda > 0$, respectively, the PDF is given by [7]:

$$f(x) = \frac{\alpha}{\lambda}\left(\frac{x}{\lambda}\right)^{\alpha-1} \exp\left(-\left(\frac{x}{\lambda}\right)^{\alpha}\right), \quad x > 0. \tag{4.10}$$

Figures 4.5 and 4.6 show the PDF and CDF with different parameter values.

The r^{th} moment is defined as [7]:

$$m_r = \lambda^r \Gamma\left(1+\frac{r}{\alpha}\right), \quad r \in \mathbb{Z}^+. \tag{4.11}$$

The MGF is derived in a similar way as (4.9) with the r^{th} moment (4.11):

$$M_X(t) = \sum_{i=0}^{\infty} \frac{t^i \lambda^i \Gamma\left(1+\frac{i}{\alpha}\right)}{i!}, \quad t \geq 0. \tag{4.12}$$

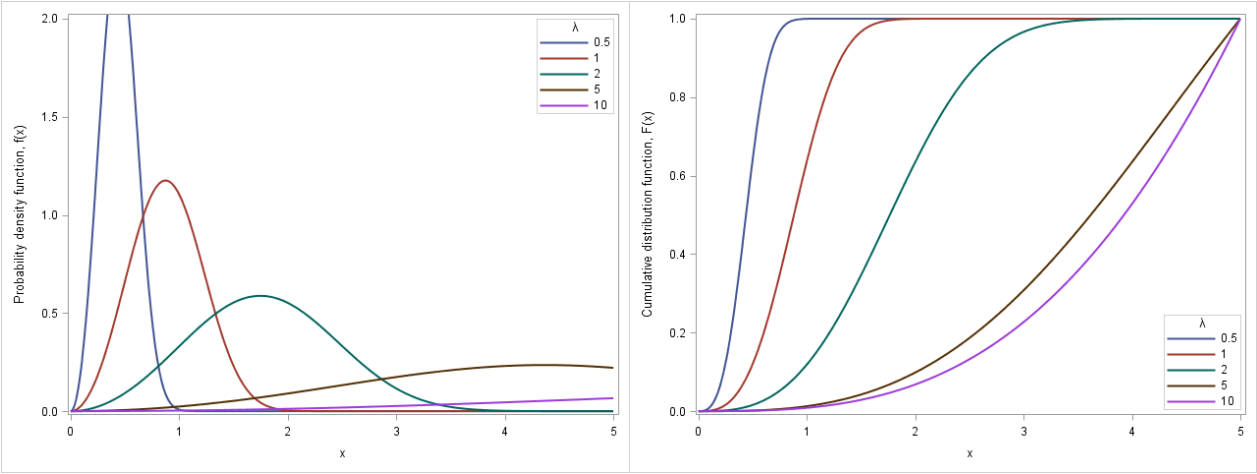


Figure 4.5: PDF (4.10) and CDF, respectively, with $\alpha = 3$ and varying λ

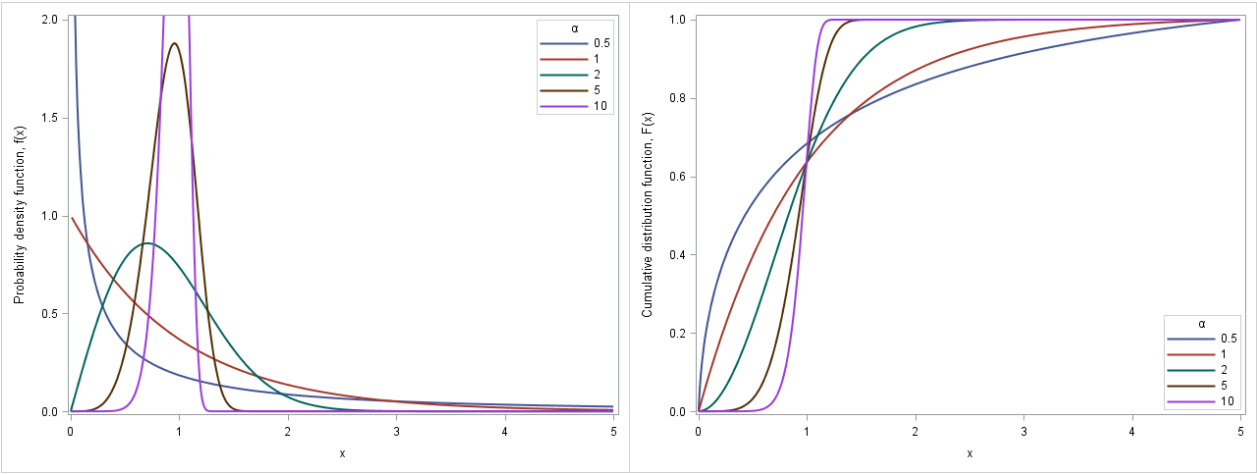


Figure 4.6: PDF (4.10) and CDF, respectively, with $\lambda = 1$ and varying α

4.1.5 Nakagami-m distribution

Let $X \sim Nakagami\,(m, \lambda)$ with *shape* and *scale* parameters $m \geq \frac{1}{2}$ and $\lambda > 0$, respectively, the PDF is given by [13]:

$$f\,(x) \quad = \quad \frac{2m^m x^{2m-1}}{\Gamma\,(m)\,\lambda^m} \exp\left(-\frac{mx^2}{\lambda}\right), \quad x > 0. \tag{4.13}$$

Figures 4.7 and 4.8 show the PDF and CDF with different parameter values.

The r^{th} moment is defined as [17]:

$$m_r \quad = \quad \frac{\lambda^{\frac{r}{2}}\Gamma\left(m + \frac{r}{2}\right)}{m^{\frac{r}{2}}\Gamma\,(m)}, \quad r \in \mathbb{Z}^+. \tag{4.14}$$

The MGF is derived in a similar way as (4.9) with the r^{th} moment (4.14):

$$M_X(t) = \sum_{i=0}^{\infty} \frac{t^i \lambda^{\frac{i}{2}} \Gamma\left(m + \frac{i}{2}\right)}{i! m^{\frac{i}{2}} \Gamma(m)}, \quad t \geq 0. \tag{4.15}$$

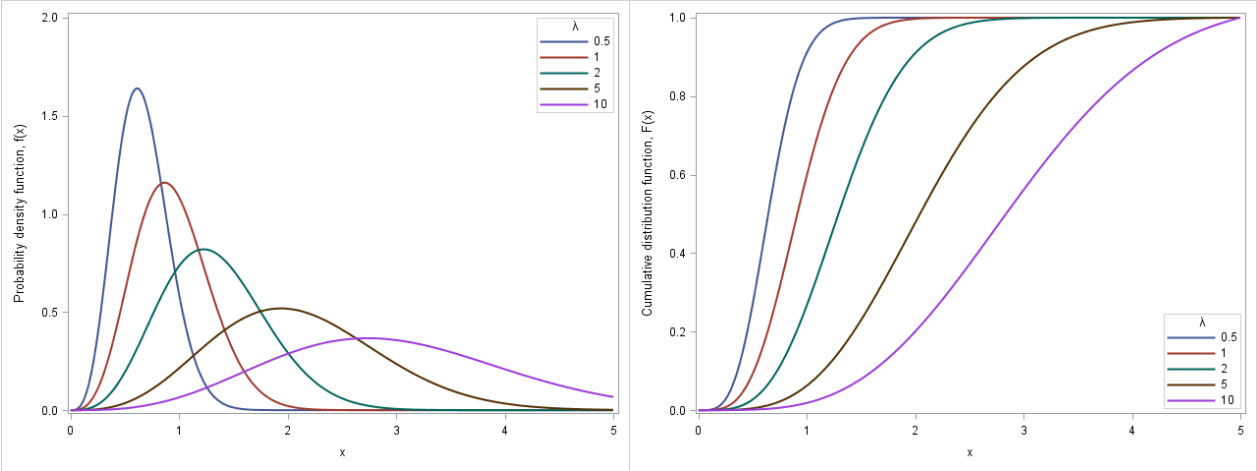


Figure 4.7: PDF (4.13) and CDF, respectively, with $m = 2$ and varying λ

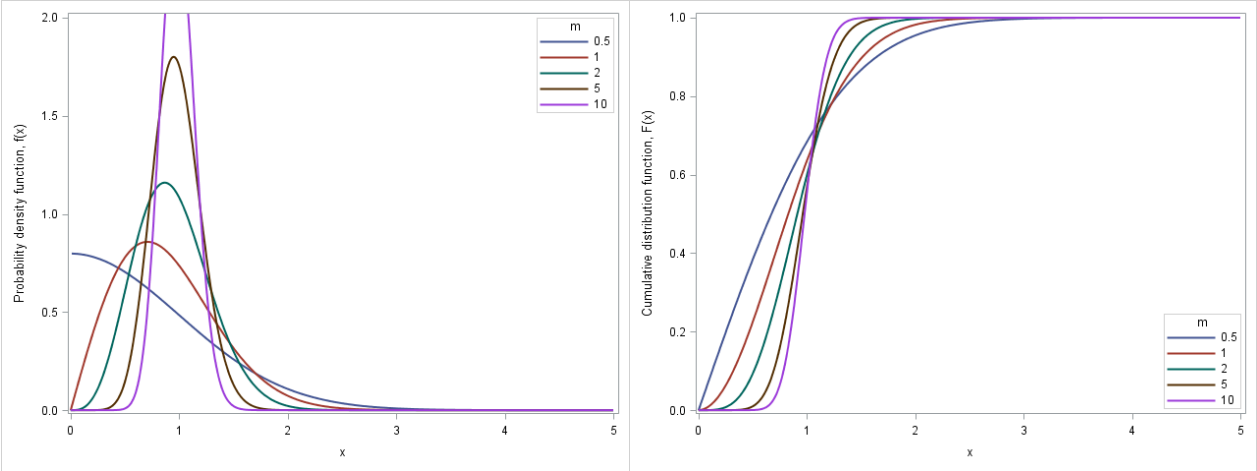


Figure 4.8: PDF (4.13) and CDF, respectively, with $\lambda = 1$ and varying m

4.2 Compound fading distributions

The compound-Weibull and compound-Rayleigh distributions were first introduced as survival models [9]. These distributions were derived by compounding the Weibull and Rayleigh distributions with the gamma distribution, using equation (3.1), where the scale parameter of the Weibull and Rayleigh distributions is gamma distributed. In order to investigate whether these would be suitable fading distributions, the shape of the distribution must be compared with that of the corresponding fading distributions that are already accepted in practice, namely the Weibull and Rayleigh distributions.

4.2.1 Compound-Weibull fading

Let $X|\theta \sim Weibull(\alpha, \theta)$ and $\theta \sim gamma(k, \lambda)$, then $X \sim compWeibull(\alpha, k, \lambda)$ with *shape* and *scale* parameters $\alpha, k > 0$ and $\lambda > 0$, respectively. The PDF is given by substituting (4.10) and (4.4) into (3.1) [9]:

$$\begin{aligned} f(x) &= \int_{\theta} f(x|\theta)\, g(\theta)\, d\theta \\ &= \int_0^{\infty} \frac{\alpha}{\theta}\left(\frac{x}{\theta}\right)^{\alpha-1} \exp\left(-\left(\frac{x}{\theta}\right)^{\alpha}\right) \frac{1}{\Gamma(k)\,\lambda^k}\theta^{k-1} \exp\left(-\frac{\theta}{\lambda}\right) d\theta \\ &= \alpha k \lambda^k x^{\alpha-1}\left(\lambda + x^{\alpha}\right)^{-(k+1)}, \quad x > 0. \end{aligned} \tag{4.16}$$

Figures 4.9, 4.10 and 4.11 show the PDF and CDF with different parameter values.

The MGF is derived using theorem 1, by substituting the MGF of the Weibull distribution (4.12) and the PDF of the gamma distribution (4.4) into (3.5), and by (4.6):

$$\begin{aligned} M_X(t) &= \int_{\theta} M_{X|\theta}(t)\, g(\theta)\, d\theta \\ &= \int_0^{\infty} \left\{ \sum_{i=0}^{\infty} \frac{t^i \theta^i \Gamma\left(1+\frac{i}{\alpha}\right)}{i!} \right\} \frac{1}{\Gamma(k)\,\lambda^k}\theta^{k-1} \exp\left(-\frac{\theta}{\lambda}\right) d\theta \\ &= \sum_{i=0}^{\infty} \frac{t^i \Gamma\left(1+\frac{i}{\alpha}\right)}{i!} \int_0^{\infty} \theta^i \frac{1}{\Gamma(k)\,\lambda^k}\theta^{k-1} \exp\left(-\frac{\theta}{\lambda}\right) d\theta \\ &= \sum_{i=0}^{\infty} \frac{t^i \Gamma\left(1+\frac{i}{\alpha}\right)}{i!} \mathbb{E}\left[\theta^i\right] \\ &= \sum_{i=0}^{\infty} \frac{t^i \lambda^i \Gamma\left(1+\frac{i}{\alpha}\right)\Gamma(k+i)}{i!\Gamma(k)}, \quad t \geq 0. \end{aligned} \tag{4.17}$$

The r^{th} moment is derived using theorem 1, by substituting the r^{th} moment of the Weibull distribution (4.11) and the PDF of the gamma distribution (4.4) into (3.6), and by (4.6):

$$\begin{aligned} m_r &= \int_{\theta} m_{r|\theta} g(\theta)\, d\theta \\ &= \int_0^{\infty} \theta^r \Gamma\left(1+\frac{r}{\alpha}\right) \frac{1}{\Gamma(k)\,\lambda^k}\theta^{k-1} \exp\left(-\frac{\theta}{\lambda}\right) d\theta \\ &= \Gamma\left(1+\frac{r}{\alpha}\right) \int_0^{\infty} \theta^r \frac{1}{\Gamma(k)\,\lambda^k}\theta^{k-1} \exp\left(-\frac{\theta}{\lambda}\right) d\theta \\ &= \Gamma\left(1+\frac{r}{\alpha}\right) \mathbb{E}\left[\theta^r\right] \\ &= \Gamma\left(1+\frac{r}{\alpha}\right) \frac{\lambda^r \Gamma(k+r)}{\Gamma(k)}, \quad r \in \mathbb{Z}^+. \end{aligned} \tag{4.18}$$

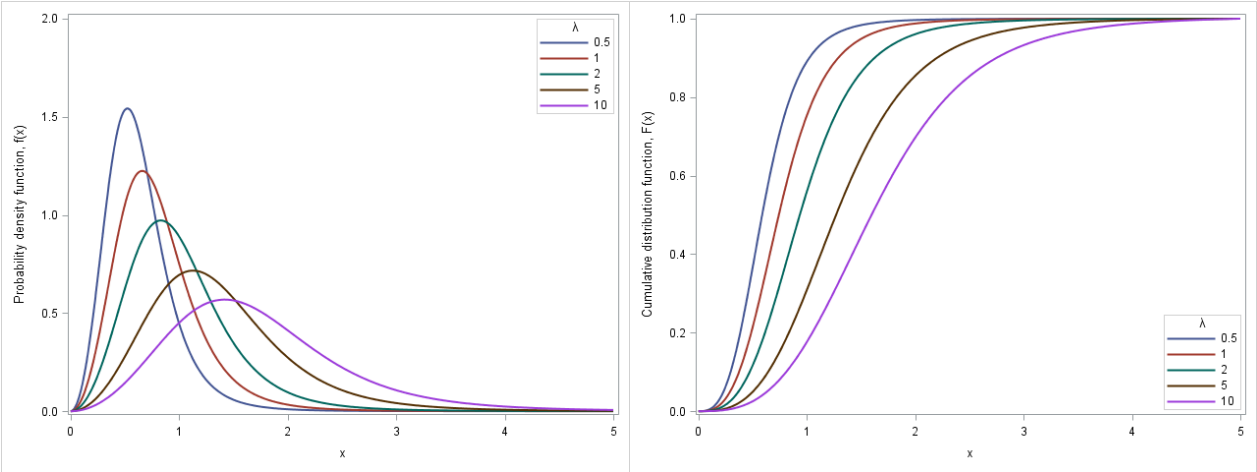


Figure 4.9: PDF (4.16) and CDF, respectively, with $\alpha = 3$, $k = 2$ and varying λ

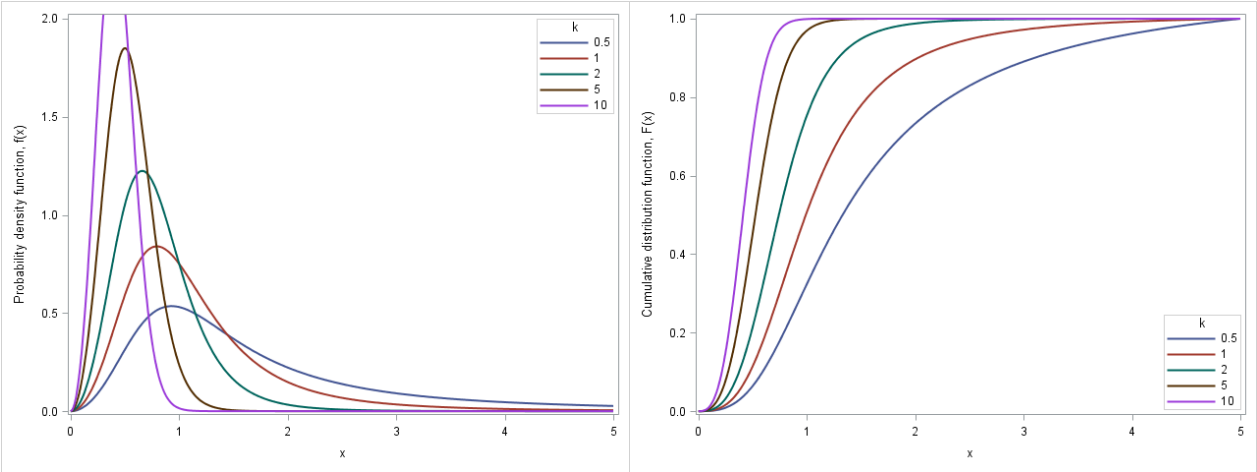


Figure 4.10: PDF (4.16) and CDF, respectively, with $\alpha = 3$, $\lambda = 1$ and varying k

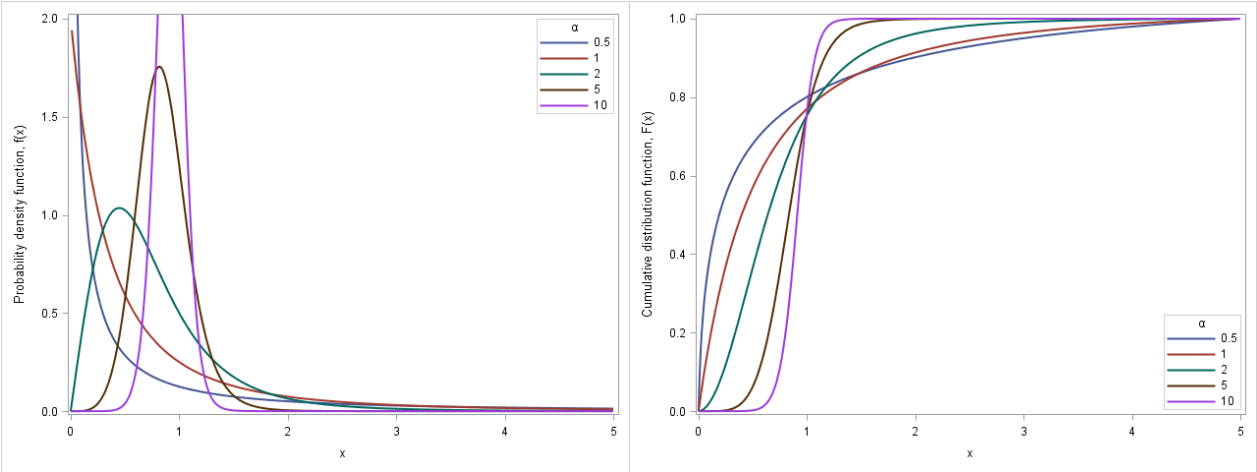


Figure 4.11: PDF (4.16) and CDF, respectively, with $k = 2$, $\lambda = 1$ and varying α

Remark.

Compare figures	k	α	λ	Skewness	Tails
4.10 and 4.5	Small	3	1	Left	Longer
4.10 and 4.5	Large	3	1	Right	Shorter
4.11 and 4.6	2	Small	1	Right	Shorter
4.11 and 4.6	2	Large	1	Left	Longer
4.9 and 4.5	2	3	Small	Left	Longer
4.9 and 4.5	2	3	Large	Right	Shorter

Table 4.1: The effect of k on compound-Weibull distribution when compared to the Weibull distribution

4.2.2 Compound-Rayleigh fading

Let $X|\theta \sim Rayleigh\,(\theta)$ and $\theta \sim gamma\,(k, \lambda)$, then $X \sim compRayleigh\,(k, \lambda)$ with *shape* and *scale* parameters $k > 0$ and $\lambda > 0$, respectively. The PDF is given by substituting (4.7) and (4.4) into (3.1) [9]:

$$\begin{aligned} f(x) &= \int_\theta f(x|\theta)\, g(\theta)\, d\theta \\ &= \int_0^\infty \frac{2x}{\theta} \exp\left(-\frac{x^2}{\theta}\right) \frac{1}{\Gamma(k)\,\lambda^k} \theta^{k-1} \exp\left(-\frac{\theta}{\lambda}\right) d\theta \\ &= 2k\lambda^k x \left(\lambda + x^2\right)^{-(k+1)}, \quad x > 0, \end{aligned} \tag{4.19}$$

which is the PDF of the compound-Weibull distribution with $\alpha = 2$. Figures 4.12 and 4.13 show the PDF and CDF with different parameter values.

The MGF is given by letting $\alpha = 2$ in equation (4.17):

$$M_X(t) = \sum_{i=0}^{\infty} \frac{t^i \lambda^i \Gamma\left(1 + \frac{i}{2}\right) \Gamma(k+i)}{i!\Gamma(k)}, \quad t \geq 0. \tag{4.20}$$

Similarly, the r^{th} moment is given by:

$$m_r = \Gamma\left(1+\frac{r}{2}\right)\frac{\lambda^r\Gamma(k+r)}{\Gamma(k)}, \quad r \in \mathbb{Z}^+. \tag{4.21}$$

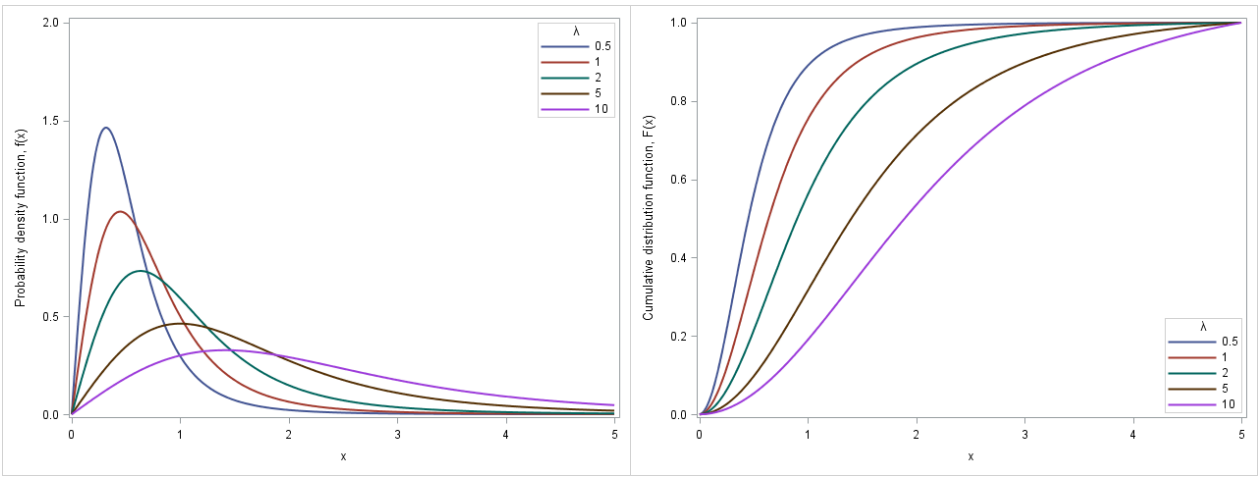


Figure 4.12: PDF (4.19) and CDF, respectively, with $k = 2$ and varying λ

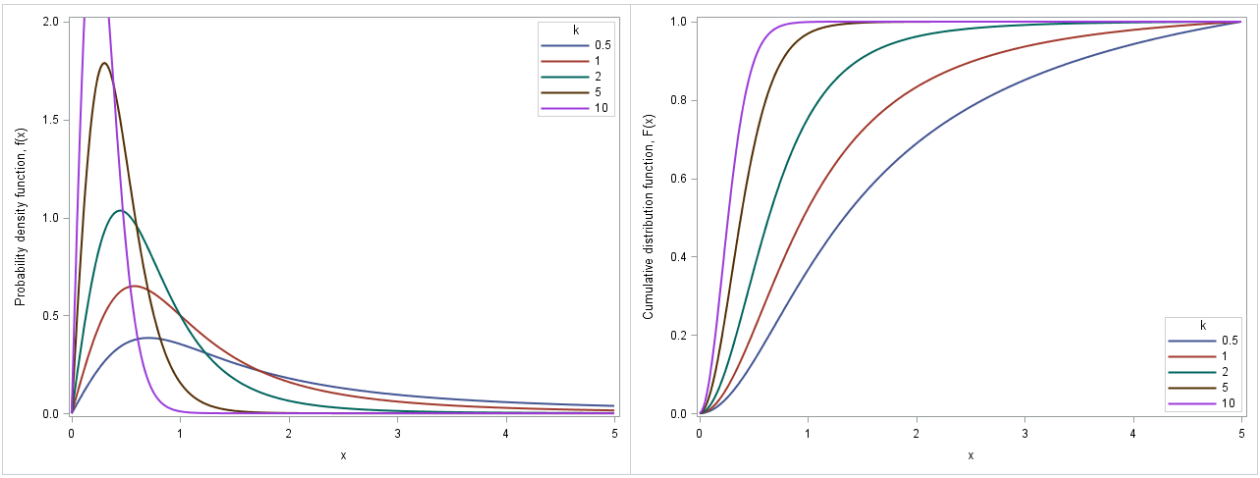


Figure 4.13: PDF (4.19) and CDF, respectively, with $\lambda = 1$ and varying k

Remark.

Compare figures	k	λ	Skewness	Tails
4.13 and 4.4	Small	1	Left	Longer
4.13 and 4.4	Large	1	Right	Shorter
4.12 and 4.4	2	Small	Right	Longer
4.12 and 4.4	2	Large	Right	Shorter

Table 4.2: The effect of k on compound-Rayleigh distribution when compared to the Rayleigh distribution

The extra parameter in the compound-Rayleigh may add an extra dimension to modelling wireless channel fading. This is discussed further in section 7.

4.3 Relationship between fading distributions

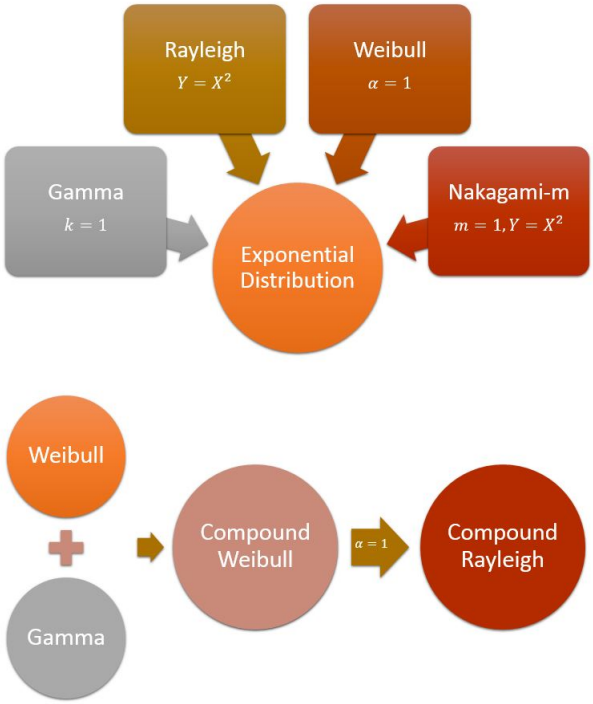


Figure 4.14: Relationship between fading distributions

The exponential distribution is a special case of the gamma, Rayleigh, Weibull and Nakagami-m distributions, as shown in figure 4.14. For the compound distributions, the compound-Rayleigh distribution is a special case of the compound-Weibull distribution.

4.3.1 Gamma to exponential transformation

Theorem 2 Let $X \sim gamma\,(k, \lambda)$ and let $k = 1$, then $X \sim exp\,(\lambda)$.

Proof. Let $f\,(x)$ be the PDF in (4.4), then if $k = 1$:

$$\begin{aligned} f\,(x) &= \frac{1}{\Gamma\,(1)\,\lambda^1} x^{1-1} \exp\left(-\frac{x}{\lambda}\right) \\ &= \frac{1}{\lambda} \exp\left(-\frac{x}{\lambda}\right), \quad x \geq 0, \end{aligned}$$

which is the PDF in (4.1). ■

4.3.2 Rayleigh to exponential transformation

Theorem 3 Let $X \sim Rayleigh\,(\lambda)$ and let $Y = X^2$ be the transformation from X to Y, then $Y \sim exp(\lambda)$.

Proof. Let $f(x)$ be the PDF in (4.7), then the PDF of Y is given by:

$$\begin{aligned} f(y) &= \frac{2\sqrt{y}}{\lambda}\exp\left[-\frac{\left(\sqrt{y}\right)^2}{\lambda}\right]|J(x\to y)| \\ &= \frac{2\sqrt{y}}{\lambda}\exp\left(-\frac{y}{\lambda}\right)\left|\frac{d}{dy}\sqrt{y}\right| \\ &= \frac{2\sqrt{y}}{\lambda}\exp\left(-\frac{y}{\lambda}\right)\frac{1}{2\sqrt{y}} \\ &= \frac{1}{\lambda}\exp\left(-\frac{y}{\lambda}\right),\ \ y\geq 0, \end{aligned}$$

which is the PDF in (4.1). ■

4.3.3 Weibull to exponential transformation

Theorem 4 Let $X\sim Weibull(\alpha,\lambda)$ and let $\alpha=1$, then $X\sim exp(\lambda)$.

Proof. Let $f(x)$ be the PDF in (4.10), then if $\alpha=1$:

$$\begin{aligned} f(x) &= \frac{1}{\lambda}\left(\frac{x}{\lambda}\right)^{1-1}\exp\left(-\left(\frac{x}{\lambda}\right)^1\right) \\ &= \frac{1}{\lambda}\exp\left(-\frac{x}{\lambda}\right),\ \ x\geq 0 \end{aligned}$$

which is the PDF in (4.1). ■

4.3.4 Nakagami-m to exponential transformation

Theorem 5 Let $X\sim Nakagami(m,\lambda)$. Let $m=1$ and let $Y=X^2$ be the transformation from X to Y, then $Y\sim exp(\lambda)$.

Proof. Let $f(x)$ be the PDF in (4.13), then if $m=1$:

$$\begin{aligned} f(x) &= \frac{2\times 1^1 x^{2\times 1-1}}{\Gamma(\times)\lambda^1}\exp\left(-\frac{1\times x^2}{\lambda}\right) \\ &= \frac{2x}{\lambda}\exp\left(-\frac{x^2}{\lambda}\right),\ \ x\geq 0, \end{aligned}$$

which is the PDF of the Rayleigh distribution in (4.7). By Theorem 2, $Y\sim exp(\lambda)$. ■

4.3.5 Compound-Weibull to compound-Rayleigh

Theorem 6 Let $X\sim compWeibull(\alpha,k,\lambda)$and let $\alpha=2$, then $X\sim compRayleigh(k,\lambda)$.

Proof. Let $f(x)$ be the PDF in (4.16), then if $\alpha=2$:

$$\begin{aligned} f(x) &= 2\times k\lambda^k x^{2-1}\left(\lambda+x^2\right)^{-(k+1)} \\ &= 2k\lambda^k x\left(\lambda+x^2\right)^{-(k+1)},\ \ x\geq 0, \end{aligned}$$

which is the PDF in (4.19). ■

5 Lognormal shadowing

The PDF of the lognormal distribution with *location* and *scale* parameters $\mu \in \mathbb{R}$ and $\sigma^2 > 0$, respectively, is given by [7]:

$$g(\lambda) = \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2}\right), \quad \lambda > 0. \tag{5.1}$$

Figure 5.1 and 5.2 shows the PDF and CDF with different parameter values.

The MGF is undefined for $x > 0$. The r^{th} moment is defined as [7]:

$$\mathbb{E}[\lambda^r] = \exp\left(r\mu + \frac{r^2\sigma^2}{2}\right), \quad r \in \mathbb{Z}^+, \tag{5.2}$$

with mean and variance [7]:

$$\mathbb{E}[\lambda] = \exp\left(\mu + \frac{\sigma^2}{2}\right), \tag{5.3}$$

and

$$var(\lambda) = \exp\left(2\mu + \sigma^2\right)\left[\exp\left(\sigma^2\right) - 1\right]. \tag{5.4}$$

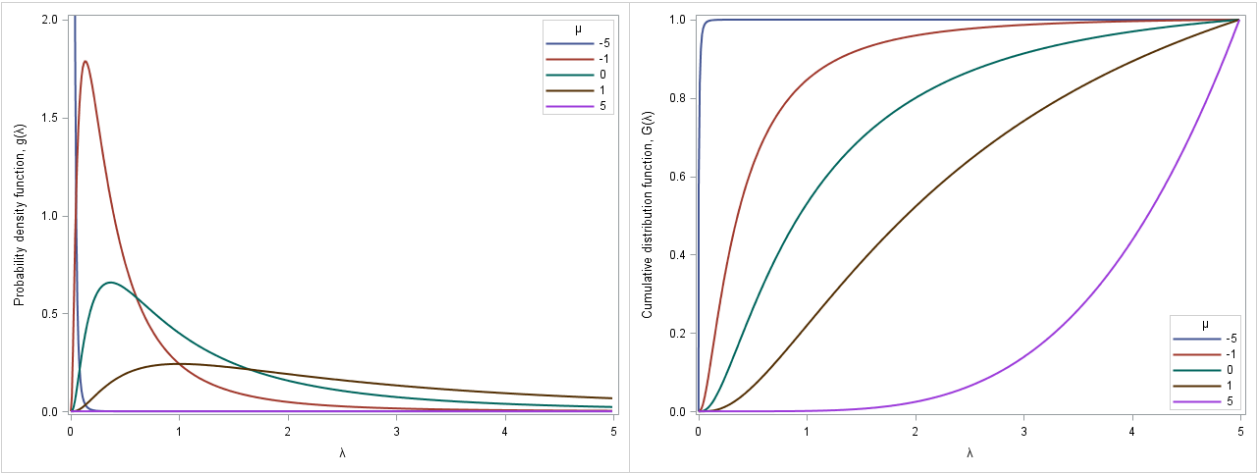


Figure 5.1: PDF (5.1) and CDF, respectively, with $\sigma = 1$ with varying μ

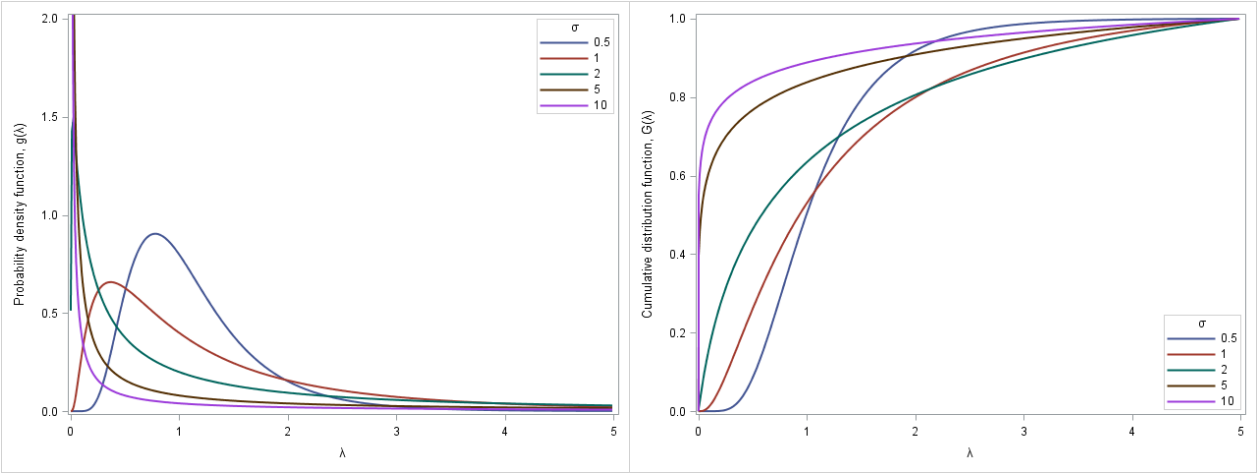


Figure 5.2: PDF (5.1) and CDF, respectively, with $\mu = 0$ and varying σ

6 Composite distributions: fading in a lognormally shadowed environment

This section explores the theory of composite fading/shadowing distributions, used to model multipath fading in a shadowed signal environment. As discussed in section 3, this is done by compounding a fading distribution with lognormal shadowing. Let $X|\lambda$ have a fading distribution conditioned on its *scale* parameter λ which has the shadowing distribution, lognormal with *location* and *scale* parameters μ and σ^2, respectively.

6.1 Exponential fading with lognormal shadowing

6.1.1 PDF of the composite exponential/lognormal distribution

The PDF of the composite exponential/lognormal distribution with *shape* parameters $\mu \in \mathbb{R}$ and $\sigma > 0$ is given by substituting (4.1) and (5.1) into (3.1):

$$\begin{aligned} h(x) &= \int_\lambda f(x|\lambda)\, g(\lambda)\, d\lambda \\ &= \int_0^\infty \frac{1}{\lambda} \exp\left(-\frac{x}{\lambda}\right) \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2}\right) d\lambda \\ &= \int_0^\infty \frac{1}{\sqrt{2\pi}\lambda^2\sigma} \exp\left(-\frac{x}{\lambda} - \frac{(\log\lambda - \mu)^2}{2\sigma^2}\right) d\lambda, \quad x \geq 0. \end{aligned} \tag{6.1}$$

Figures 6.1 and 6.2 show the PDF and CDF with different parameter values.

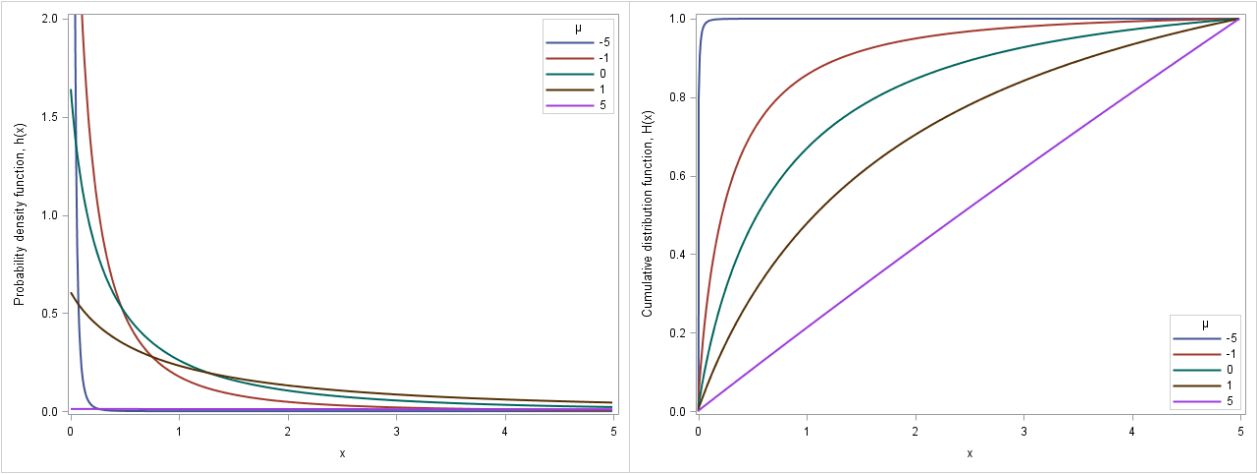


Figure 6.1: PDF (6.1) and CDF, respectively, with $\sigma = 1$ and varying μ

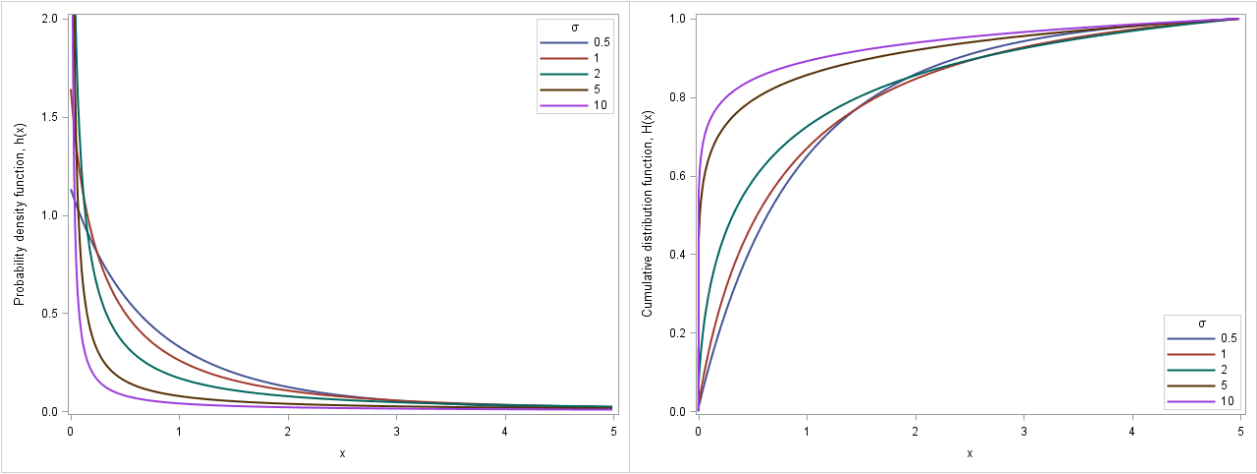


Figure 6.2: PDF (6.1) and CDF, respectively, with $\mu = 0$ and varying σ

6.1.2 MGF of the composite exponential/lognormal distribution

By substituting (4.2) and (5.1) into (3.5) and by (5.2), the MGF is given by:

$$\begin{aligned} M_X(t) &= \int_\lambda M_{X|\lambda}(t)\, g(\lambda)\, d\lambda \\ &= \int_0^\infty \frac{\lambda}{\lambda - t} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2}\right) d\lambda, \quad t < \lambda. \end{aligned} \tag{6.2}$$

6.1.3 Moments of the composite exponential/lognormal distribution

By substituting (4.3) and (5.1) into (3.5) and by (5.2), the r^{th} moment is given by:

$$
\begin{aligned}
m_r &= \int_\lambda m_{r|\lambda} g(\lambda)\, d\lambda \\
&= \int_0^\infty r!\lambda^r \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right) d\lambda \\
&= r! \int_0^\infty \lambda^r \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right) d\lambda \\
&= r!\mathbb{E}\left[\lambda^r\right] \\
&= r! \exp\left(r\mu + \frac{r^2\sigma^2}{2}\right), \quad r \in \mathbb{Z}^+. \qquad (6.3)
\end{aligned}
$$

By (6.3), the mean and variance can be derived as follows:

$$
\begin{aligned}
\mathbb{E}[X] &= m_1 \\
&= 1! \times \exp\left(1 \times \mu + \frac{1^2 \times \sigma^2}{2}\right) \\
&= \exp\left(\mu + \frac{\sigma^2}{2}\right), \qquad (6.4)
\end{aligned}
$$

and

$$
\begin{aligned}
var(X) &= m_2 - m_1^2 \\
&= 2! \times \exp\left(2 \times \mu + \frac{2^2 \times \sigma^2}{2}\right) - \left[\exp\left(\mu + \frac{\sigma^2}{2}\right)\right]^2 \\
&= 2\exp\left(2\mu + 2\sigma^2\right) - \exp\left(2\mu + \sigma^2\right) \\
&= \exp\left(2\mu + \sigma^2\right)\left[2\exp\left(\sigma^2\right) - 1\right]. \qquad (6.5)
\end{aligned}
$$

6.2 Gamma fading with lognormal shadowing

6.2.1 PDF of the composite gamma/lognormal distribution

The PDF of the composite gamma/lognormal distribution with *shape* parameters $k > 0$, $\mu \in \mathbb{R}$ and $\sigma > 0$ is given by substituting (4.1) and (5.1) into (3.1):

$$
\begin{aligned}
h(x) &= \int_\lambda f(x|\lambda)\, g(\lambda)\, d\lambda \\
&= \int_0^\infty \frac{1}{\Gamma(k)\lambda^k} x^{k-1} \exp\left(-\frac{x}{\lambda}\right) \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right) d\lambda \\
&= \int_0^\infty \frac{x^{k-1}}{\Gamma(k)\sqrt{2\pi}\lambda^{k+1}\sigma} \exp\left(-\frac{x}{\lambda} - \frac{(\log\lambda-\mu)^2}{2\sigma^2}\right) d\lambda, \quad x > 0, \qquad (6.6)
\end{aligned}
$$

which is the simple case of the generalised gamma fading/shadowing distribution [12]. Figures 6.3, 6.4 and 6.5 show the PDF and CDF with different parameter values.

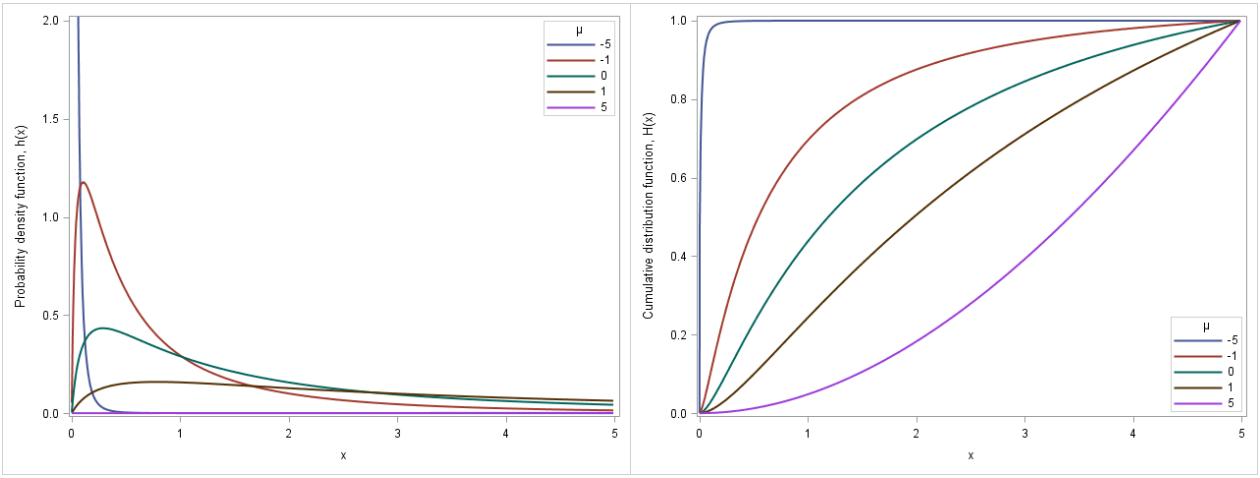


Figure 6.3: PDF (6.6) and CDF, respectively, with $k = 2$, $\sigma = 1$ and varying μ

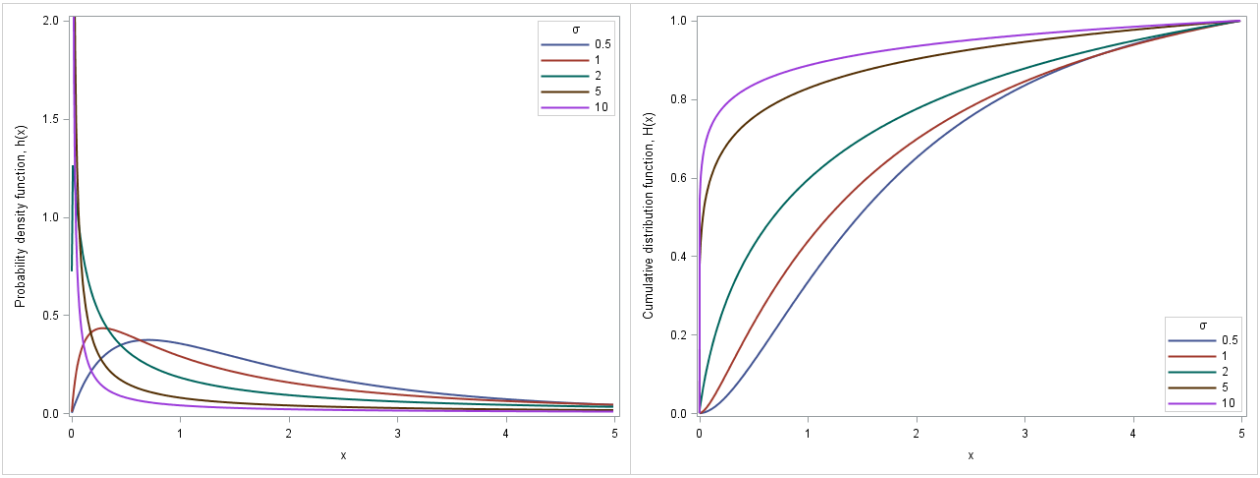


Figure 6.4: PDF (6.6) and CDF, respectively, with $k = 2$, $\mu = 0$ and varying σ

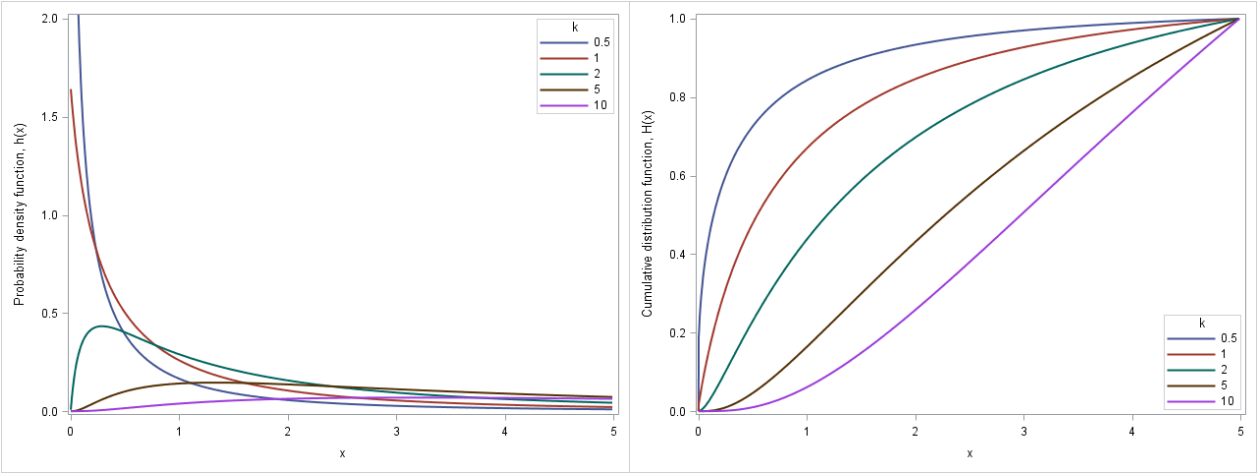


Figure 6.5: PDF (6.6) and CDF, respectively, with $\mu = 0$, $\sigma = 1$ and varying k

6.2.2 MGF of the composite gamma/lognormal distribution

By substituting (4.5) and (5.1) into (3.5) and by (5.2), the MGF is given by:

$$\begin{aligned} M_X(t) &= \int_\lambda M_{X|\lambda}(t)\, g(\lambda)\, d\lambda \\ &= \int_0^\infty (1-\lambda t)^{-k} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right) d\lambda, \quad t < \frac{1}{\lambda}. \end{aligned} \tag{6.7}$$

6.2.3 Moments of the composite gamma/lognormal distribution

By substituting (4.6) and (5.1) into (3.5) and by (5.2), the r^{th} moment is given by:

$$\begin{aligned} m_r &= \int_\lambda m_{r|\lambda} g(\lambda)\, d\lambda \\ &= \int_0^\infty \frac{\lambda^r \Gamma(k+r)}{\Gamma(k)} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right) d\lambda \\ &= \frac{\Gamma(k+r)}{\Gamma(k)} \int_0^\infty \lambda^r \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right) d\lambda \\ &= \frac{\Gamma(k+r)}{\Gamma(k)} \mathbb{E}[\lambda^r] \\ &= \frac{\Gamma(k+r)}{\Gamma(k)} \exp\left(r\mu + \frac{r^2\sigma^2}{2}\right), \quad r \in \mathbb{Z}^+. \end{aligned} \tag{6.8}$$

By (6.8), the mean and variance can be derived as follows:

$$\begin{aligned} \mathbb{E}[X] &= m_1 \\ &= \frac{\Gamma(k+1)}{\Gamma(k)} \exp\left(1 \times \mu + \frac{1^2 \times \sigma^2}{2}\right) \\ &= k \exp\left(\mu + \frac{\sigma^2}{2}\right), \end{aligned} \tag{6.9}$$

and

$$\begin{aligned} var\left(X\right) &= m_2 - m_1^2 \\ &= \frac{\Gamma\left(k+2\right)}{\Gamma\left(k\right)} \exp\left(2 \times \mu + \frac{2^2 \times \sigma^2}{2}\right) - \left[k \exp\left(\mu + \frac{\sigma^2}{2}\right)\right]^2 \\ &= k\left(k+1\right) \exp\left(2\mu + 2\sigma^2\right) - k^2 \exp\left(2\mu + \sigma^2\right) \\ &= \exp\left(2\mu + \sigma^2\right)\left[\left(k^2 + k\right) \exp\left(\sigma^2\right) - k^2\right]. \end{aligned} \tag{6.10}$$

6.3 Rayleigh fading with lognormal shadowing

6.3.1 PDF of the composite Rayleigh/lognormal distribution

The PDF of the composite Rayleigh/lognormal distribution with *shape* parameters $\mu \in \mathbb{R}$ and $\sigma > 0$ is given by substituting (4.7) and (5.1) into (3.1):

$$\begin{aligned} h\left(x\right) &= \int_{\lambda} f\left(x|\lambda\right) g\left(\lambda\right) d\lambda \\ &= \int_0^{\infty} \frac{2x}{\lambda} \exp\left(-\frac{x^2}{\lambda}\right) \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{\left(\log\lambda - \mu\right)^2}{2\sigma^2}\right) d\lambda \\ &= \int_0^{\infty} \sqrt{\frac{2}{\pi}} \frac{x}{\lambda^2\sigma} \exp\left(-\frac{x^2}{\lambda} - \frac{\left(\log\lambda - \mu\right)^2}{2\sigma^2}\right) d\lambda, \quad x \geq 0, \end{aligned} \tag{6.11}$$

which is a reparameterised PDF of the Suzuki distribution given in [15]. Figures 6.6 and 6.7 shows the PDF and CDF with different parameter values.

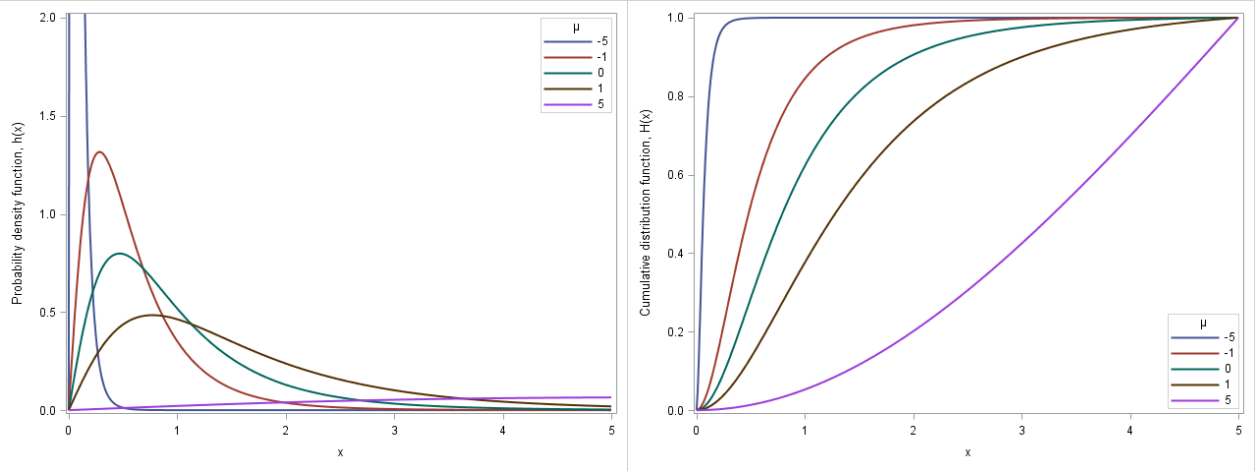


Figure 6.6: PDF (6.11) and CDF, respectively, with $\sigma = 1$ and varying μ

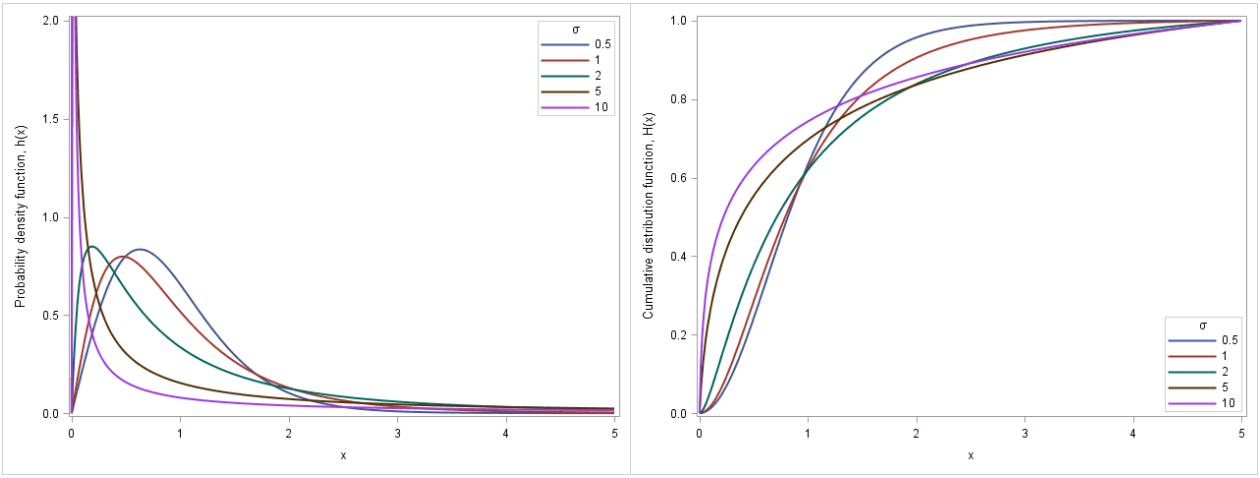


Figure 6.7: PDF (6.11) and CDF, respectively, with $\mu = 0$ and varying σ

6.3.2 MGF of the composite Rayleigh/lognormal distribution

By substituting (4.9) and (5.1) into (3.5) and by (5.2), the MGF is given by:

$$\begin{aligned} M_X(t) &= \int_\lambda M_{X|\lambda}(t)\, g(\lambda)\, d\lambda \\ &= \int_0^\infty \left\{ \sum_{i=0}^\infty \frac{t^i \lambda^{\frac{i}{2}} \Gamma\left(1+\frac{i}{2}\right)}{i!} \right\} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2} \right) d\lambda \\ &= \sum_{i=0}^\infty \frac{t^i \Gamma\left(1+\frac{i}{2}\right)}{i!} \int_0^\infty \lambda^{\frac{i}{2}} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2} \right) d\lambda \\ &= \sum_{i=0}^\infty \frac{t^i \Gamma\left(1+\frac{i}{2}\right)}{i!} \mathbb{E}\left[\lambda^{\frac{i}{2}}\right] \\ &= \sum_{i=0}^\infty \frac{t^i \Gamma\left(1+\frac{i}{2}\right)}{i!} \exp\left(\frac{i\mu}{2} + \frac{i^2\sigma^2}{8} \right), \quad t \geq 0. \end{aligned} \tag{6.12}$$

6.3.3 Moments of the composite Rayleigh/lognormal distribution

By substituting (4.8) and (5.1) into (3.5) and by (5.2), the r^{th} moment is given by:

$$\begin{aligned} m_r &= \int_\lambda m_{r|\lambda} g(\lambda)\, d\lambda \\ &= \int_0^\infty \lambda^{\frac{r}{2}} \Gamma\left(1+\frac{r}{2}\right) \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2} \right) d\lambda \\ &= \Gamma\left(1+\frac{r}{2}\right) \int_0^\infty \lambda^{\frac{r}{2}} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2} \right) d\lambda \\ &= \Gamma\left(1+\frac{r}{2}\right) \mathbb{E}\left[\lambda^{\frac{r}{2}}\right] \\ &= \Gamma\left(1+\frac{r}{2}\right) \exp\left(\frac{r\mu}{2} + \frac{r^2\sigma^2}{8} \right), \quad r \in \mathbb{Z}^+. \end{aligned} \tag{6.13}$$

By (6.13), the mean and variance can be derived as follows:

$$\begin{aligned}\mathbb{E}[X] &= m_1 \\ &= \Gamma\left(1+\frac{1}{2}\right)\exp\left(\frac{1\times\mu}{2}+\frac{1^2\times\sigma^2}{8}\right) \\ &= \sqrt{\frac{\pi}{4}}\exp\left(\frac{\mu}{2}+\frac{\sigma^2}{8}\right), \end{aligned} \tag{6.14}$$

and

$$\begin{aligned} var(X) &= m_2 - m_1^2 \\ &= \Gamma\left(1+\frac{2}{2}\right)\exp\left(\frac{2\times\mu}{2}+\frac{2^2\times\sigma^2}{8}\right) - \left[\sqrt{\frac{\pi}{4}}\exp\left(\frac{\mu}{2}+\frac{\sigma^2}{8}\right)\right]^2 \\ &= \exp\left(\mu+\frac{\sigma^2}{2}\right) - \frac{\pi}{4}\exp\left(\mu+\frac{\sigma^2}{4}\right) \\ &= \exp\left(\mu+\frac{\sigma^2}{4}\right)\left[\exp\left(\frac{\sigma^2}{4}\right)-\frac{\pi}{4}\right]. \end{aligned} \tag{6.15}$$

6.4 Weibull fading with lognormal shadowing

6.4.1 PDF of the composite Weibull/lognormal distribution

The PDF of the composite Weibull/lognormal distribution with *shape* parameters $\alpha > 0$, $\mu \in \mathbb{R}$ and $\sigma > 0$ is given by substituting (4.10) and (5.1) into (3.1):

$$\begin{aligned} h(x) &= \int_\lambda f(x|\lambda)\, g(\lambda)\, d\lambda \\ &= \int_0^\infty \frac{\alpha}{\lambda}\left(\frac{x}{\lambda}\right)^{\alpha-1}\exp\left(-\left(\frac{x}{\lambda}\right)^\alpha\right)\frac{1}{\sqrt{2\pi}\lambda\sigma}\exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right)d\lambda \\ &= \int_0^\infty \frac{\alpha x^{\alpha-1}}{\sqrt{2\pi}\lambda^{\alpha+1}\sigma}\exp\left(-\left(\frac{x}{\lambda}\right)^\alpha-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right)d\lambda,\ \ x>0, \end{aligned} \tag{6.16}$$

which is similar to the distribution developed in [8]. Figures 6.8, 6.9 and 6.10 show the PDF and CDF with different parameter values.

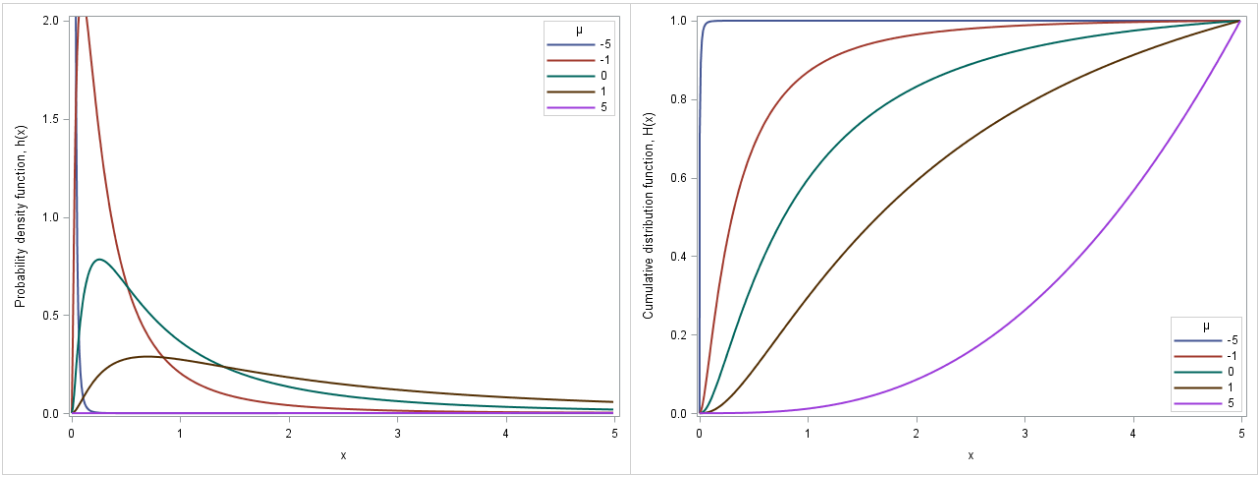


Figure 6.8: PDF (6.16) and CDF, respectively, with $\alpha = 3$, $\sigma = 1$ and varying μ

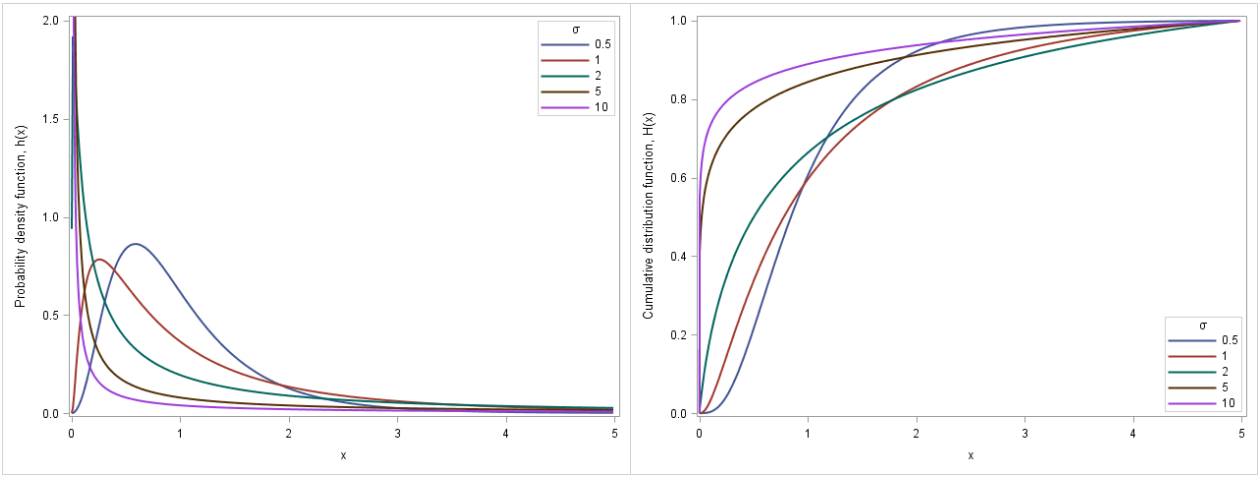


Figure 6.9: PDF (6.16) and CDF, respectively, with $\alpha = 3$, $\mu = 0$ and varying σ

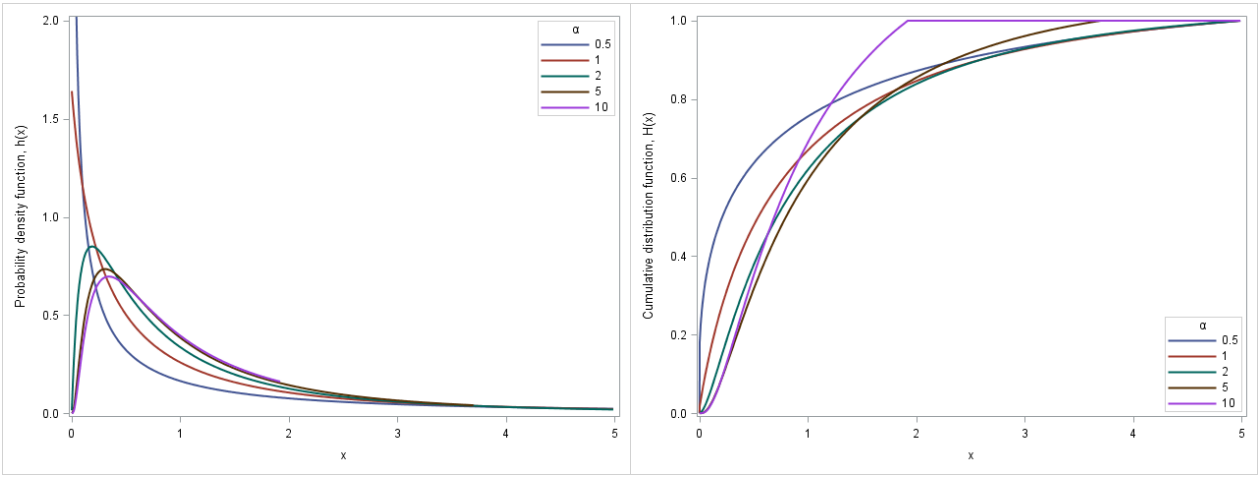


Figure 6.10: PDF (6.16) and CDF, respectively, with $\mu = 0$, $\sigma = 1$ and varying α

6.4.2 MGF of the composite Weibull/lognormal distribution

By substituting (4.12) and (5.1) into (3.5) and by (5.2), the MGF is given by:

$$\begin{aligned} M_X(t) &= \int_\lambda M_{X|\lambda}(t)\, g(\lambda)\, d\lambda \\ &= \int_0^\infty \left\{ \sum_{i=0}^\infty \frac{t^i \lambda^i \Gamma\left(1+\frac{i}{a}\right)}{i!} \right\} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2} \right) d\lambda \\ &= \sum_{i=0}^\infty \frac{t^i \lambda^i \Gamma\left(1+\frac{i}{a}\right)}{i!} \int_0^\infty \lambda^i \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2} \right) d\lambda \\ &= \sum_{i=0}^\infty \frac{t^i \lambda^i \Gamma\left(1+\frac{i}{a}\right)}{i!} \mathbb{E}\left[\lambda^i\right] \\ &= \sum_{i=0}^\infty \frac{t^i \lambda^i \Gamma\left(1+\frac{i}{a}\right)}{i!} \exp\left(i\mu + \frac{i^2\sigma^2}{2} \right), \quad t \geq 0. \end{aligned} \tag{6.17}$$

6.4.3 Moments of the composite Weibull/lognormal distribution

By substituting (4.11) and (5.1) into (3.5) and by (5.2), the r^{th} moment is given by:

$$\begin{aligned} m_r &= \int_\lambda m_{r|\lambda} g(\lambda)\, d\lambda \\ &= \int_0^\infty \lambda^r \Gamma\left(1+\frac{r}{\alpha}\right) \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2} \right) d\lambda \\ &= \Gamma\left(1+\frac{r}{\alpha}\right) \int_0^\infty \lambda^r \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2} \right) d\lambda \\ &= \Gamma\left(1+\frac{r}{\alpha}\right) \mathbb{E}\left[\lambda^r\right] \\ &= \Gamma\left(1+\frac{r}{\alpha}\right) \exp\left(r\mu + \frac{r^2\sigma^2}{2} \right), \quad r \in \mathbb{Z}^+. \end{aligned} \tag{6.18}$$

By (6.18), the mean and variance can be derived as follows:

$$\begin{aligned}\mathbb{E}[X] &= m_1 \\ &= \Gamma\left(1+\frac{1}{\alpha}\right)\exp\left(1\times\mu+\frac{1^2\times\sigma^2}{2}\right) \\ &= \Gamma\left(1+\frac{1}{\alpha}\right)\exp\left(\mu+\frac{\sigma^2}{2}\right), \end{aligned} \tag{6.19}$$

and

$$\begin{aligned} var(X) &= m_2 - m_1^2 \\ &= \Gamma\left(1+\frac{2}{\alpha}\right)\exp\left(2\times\mu+\frac{2^2\times\sigma^2}{2}\right) - \left[\Gamma\left(1+\frac{1}{\alpha}\right)\exp\left(\mu+\frac{\sigma^2}{2}\right)\right]^2 \\ &= \Gamma\left(1+\frac{2}{\alpha}\right)\exp\left(2\mu+2\sigma^2\right) - \Gamma^2\left(1+\frac{1}{\alpha}\right)\exp\left(2\mu+\sigma^2\right) \\ &= \exp\left(2\mu+\sigma^2\right)\left[\Gamma\left(1+\frac{2}{\alpha}\right)\exp\left(\sigma^2\right) - \Gamma^2\left(1+\frac{1}{\alpha}\right)\right]. \end{aligned} \tag{6.20}$$

6.5 Nakagami-m fading with lognormal shadowing

6.5.1 PDF of the composite Nakagami-m/lognormal distribution

The PDF of the composite Nakagami-m/lognormal distribution with *shape* parameters $m \geq \frac{1}{2}$, $\mu \in \mathbb{R}$ and $\sigma > 0$ is given by substituting (4.13) and (5.1) into (3.1):

$$\begin{aligned} h(x) &= \int_\lambda f(x|\lambda)\, g(\lambda)\, d\lambda \\ &= \int_0^\infty \frac{2m^m x^{2m-1}}{\Gamma(m)\lambda^m}\exp\left(-\frac{mx^2}{\lambda}\right)\frac{1}{\sqrt{2\pi}\lambda\sigma}\exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right)d\lambda \\ &= \int_0^\infty \frac{2m^m x^{2m-1}}{\sqrt{2\pi}\Gamma(m)\lambda^{m+1}\sigma}\exp\left(-\frac{mx^2}{\lambda}-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right)d\lambda, \quad x>0, \end{aligned} \tag{6.21}$$

which is the PDF of the composite Nakagami-m/lognormal channel in [6, 13]. Figures 6.11, 6.12 and 6.13 show the PDF and CDF with different parameter values.

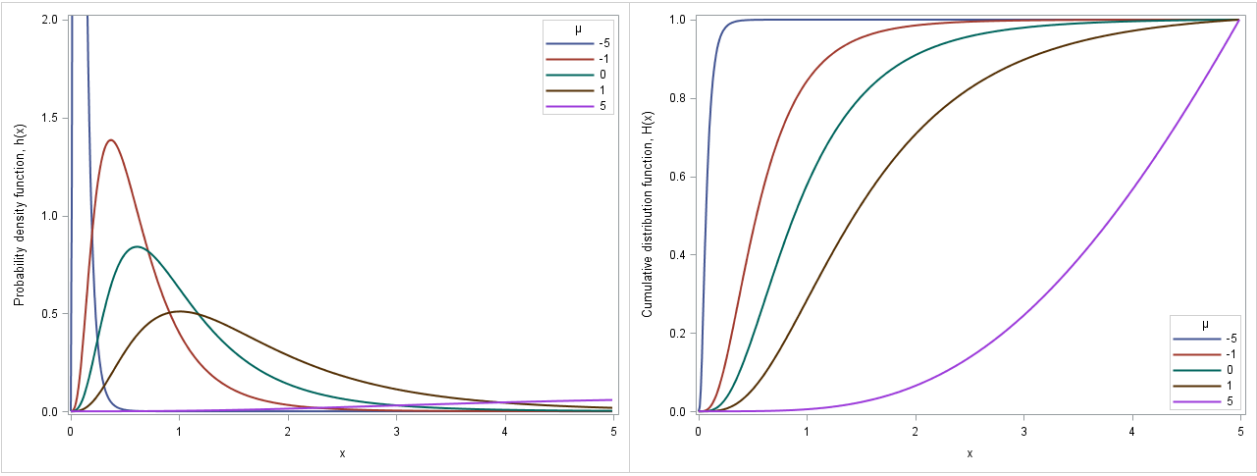


Figure 6.11: PDF (6.21) and CDF, respectively, with $m = 2$, $\sigma = 1$ and varying μ

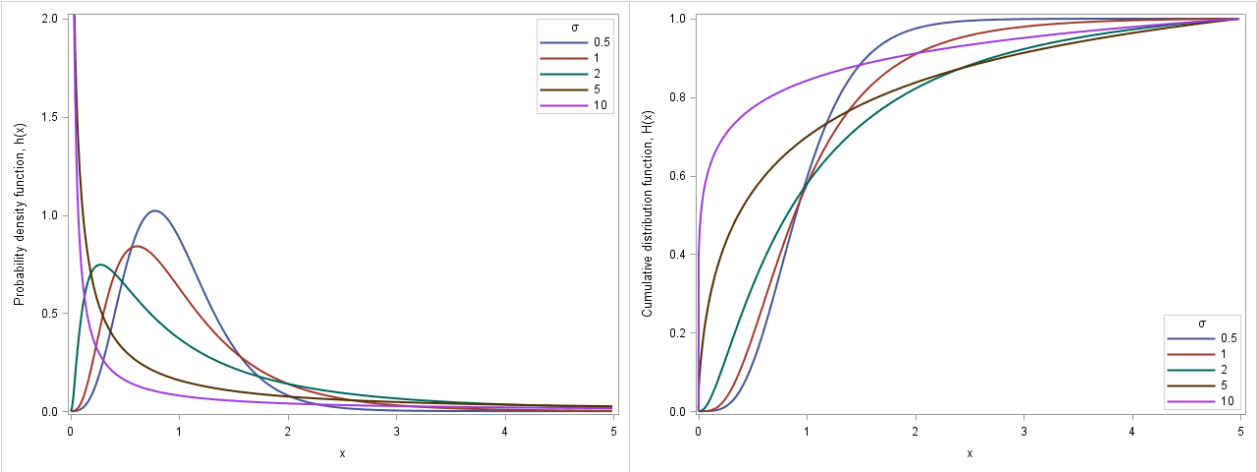


Figure 6.12: PDF (6.21) and CDF, respectively, with $m = 2$, $\mu = 0$ and varying σ

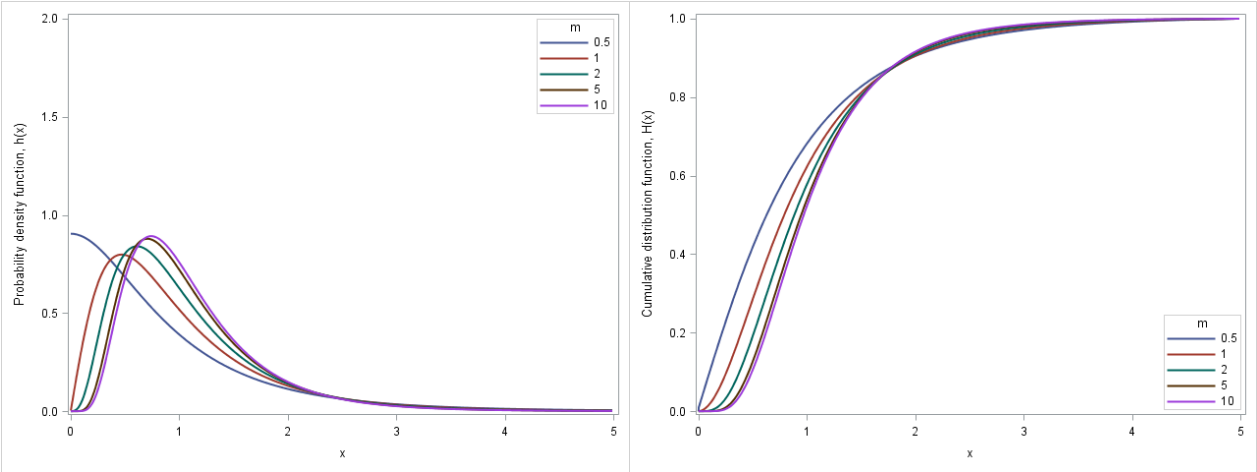


Figure 6.13: PDF (6.21) and CDF, respectively, with $\mu = 0$, $\sigma = 1$ and varying m

6.5.2 MGF of the composite Nakagami-m/lognormal distribution

By substituting (4.15) and (5.1) into (3.5) and by (5.2), the MGF is given by:

$$\begin{aligned} M_X(t) &= \int_\lambda M_{X|\lambda}(t)\, g(\lambda)\, d\lambda \\ &= \int_0^\infty \left\{ \sum_{i=0}^\infty \frac{t^i \lambda^{\frac{i}{2}} \Gamma\left(m+\frac{i}{2}\right)}{i! m^{\frac{i}{2}} \Gamma(m)} \right\} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2} \right) d\lambda \\ &= \sum_{i=0}^\infty \frac{t^i \Gamma\left(m+\frac{i}{2}\right)}{i! m^{\frac{i}{2}} \Gamma(m)} \int_0^\infty \lambda^{\frac{i}{2}} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2} \right) d\lambda \\ &= \sum_{i=0}^\infty \frac{t^i \Gamma\left(m+\frac{i}{2}\right)}{i! m^{\frac{i}{2}} \Gamma(m)} \mathbb{E}\left[\lambda^{\frac{i}{2}}\right] \\ &= \sum_{i=0}^\infty \frac{t^i \Gamma\left(m+\frac{i}{2}\right)}{i! m^{\frac{i}{2}} \Gamma(m)} \exp\left(\frac{i\mu}{2} + \frac{i^2\sigma^2}{8} \right), \quad t \geq 0. \end{aligned} \tag{6.22}$$

6.5.3 Moments of the composite Nakagami-m/lognormal distribution

By substituting (4.14) and (5.1) into (3.5) and by (5.2), the r^{th} moment is given by:

$$\begin{aligned} m_r &= \int_{\lambda} m_{r|\lambda} g(\lambda)\, d\lambda \\ &= \int_0^{\infty} \frac{\lambda^{\frac{r}{2}}\Gamma\left(m+\frac{r}{2}\right)}{m^{\frac{r}{2}}\Gamma(m)} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right) d\lambda \\ &= \frac{\Gamma\left(m+\frac{r}{2}\right)}{m^{\frac{r}{2}}\Gamma(m)} \int_0^{\infty} \lambda^{\frac{r}{2}} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right) d\lambda \\ &= \frac{\Gamma\left(m+\frac{r}{2}\right)}{m^{\frac{r}{2}}\Gamma(m)} \mathbb{E}\left[\lambda^{\frac{r}{2}}\right] \\ &= \frac{\Gamma\left(m+\frac{r}{2}\right)}{m^{\frac{r}{2}}\Gamma(m)} \exp\left(\frac{r\mu}{2}+\frac{r^2\sigma^2}{8}\right), \ r\in\mathbb{Z}^+. \end{aligned} \tag{6.23}$$

By (6.23), the mean and variance can be derived as follows:

$$\begin{aligned} \mathbb{E}[X] &= m_1 \\ &= \frac{\Gamma\left(m+\frac{1}{2}\right)}{\sqrt{m}\Gamma(m)} \exp\left(\frac{1\times\mu}{2}+\frac{1^2\times\sigma^2}{8}\right) \\ &= \frac{\Gamma\left(m+\frac{1}{2}\right)}{\sqrt{m}\Gamma(m)} \exp\left(\frac{\mu}{2}+\frac{\sigma^2}{8}\right), \end{aligned} \tag{6.24}$$

and

$$\begin{aligned} var(X) &= m_2 - m_1^2 \\ &= \frac{\Gamma\left(m+\frac{2}{2}\right)}{m\Gamma(m)} \exp\left(\frac{2\times\mu}{2}+\frac{2^2\times\sigma^2}{8}\right) - \left[\frac{\Gamma\left(m+\frac{1}{2}\right)}{\sqrt{m}\Gamma(m)} \exp\left(\frac{\mu}{2}+\frac{\sigma^2}{8}\right)\right]^2 \\ &= \frac{m!}{m\times(m-1)!} \exp\left(\mu+\frac{\sigma^2}{2}\right) - \frac{\Gamma^2\left(m+\frac{1}{2}\right)}{m\Gamma^2(m)} \exp\left(\mu+\frac{\sigma^2}{4}\right) \\ &= \exp\left(\mu+\frac{\sigma^2}{4}\right)\left[\exp\left(\frac{\sigma^2}{4}\right) - \frac{\Gamma^2\left(m+\frac{1}{2}\right)}{m\Gamma^2(m)}\right]. \end{aligned} \tag{6.25}$$

6.6 Compound-Weibull fading with lognormal shadowing

6.6.1 PDF of the composite compound-Weibull/lognormal distribution

The PDF of the composite compound-Weibull/lognormal distribution with *shape* parameters $\alpha, k > 0$, $\mu \in \mathbb{R}$ and $\sigma > 0$ is given by substituting (4.16) and (5.1) into (3.1):

$$\begin{aligned} h(x) &= \int_{\lambda} f(x|\lambda)\, g(\lambda)\, d\lambda \\ &= \int_0^{\infty} \alpha k \lambda^k x^{\alpha-1} \left(\lambda + x^{\alpha}\right)^{-(k+1)} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right) d\lambda \\ &= \int_0^{\infty} \frac{\alpha k \lambda^{k-1} x \left(\lambda + x^2\right)^{-(k+1)}}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right) d\lambda, \ \ x \geq 0. \end{aligned} \tag{6.26}$$

Figures 6.14, 6.15 and 6.16 show the PDF and CDF with different parameter values.

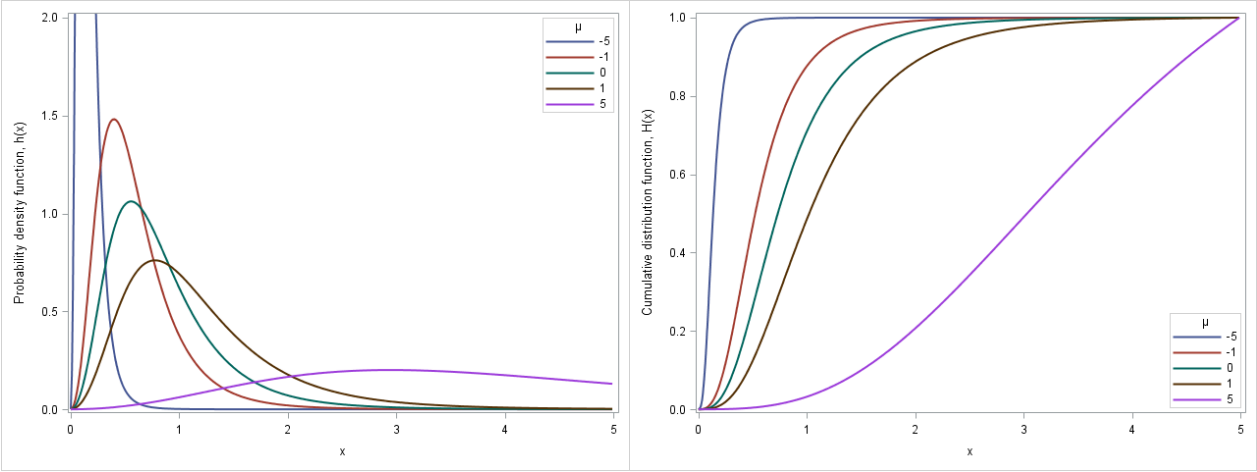


Figure 6.14: PDF (6.26) and CDF, respectively, with $\alpha = 3$, $k = 2$, $\sigma = 1$ and varying μ

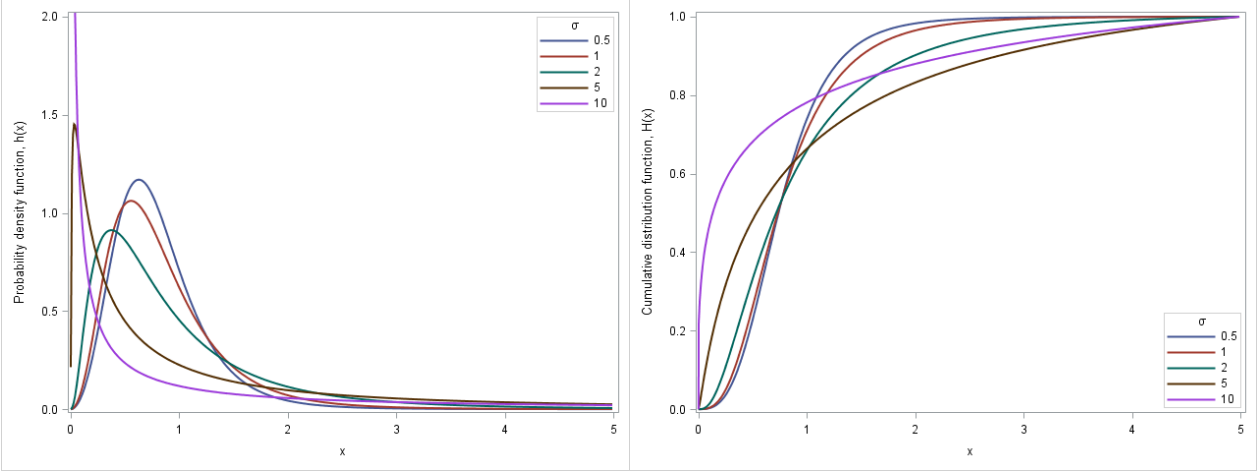


Figure 6.15: PDF (6.26) and CDF, respectively, with $\alpha = 3$, $k = 2$, $\mu = 0$ and varying σ

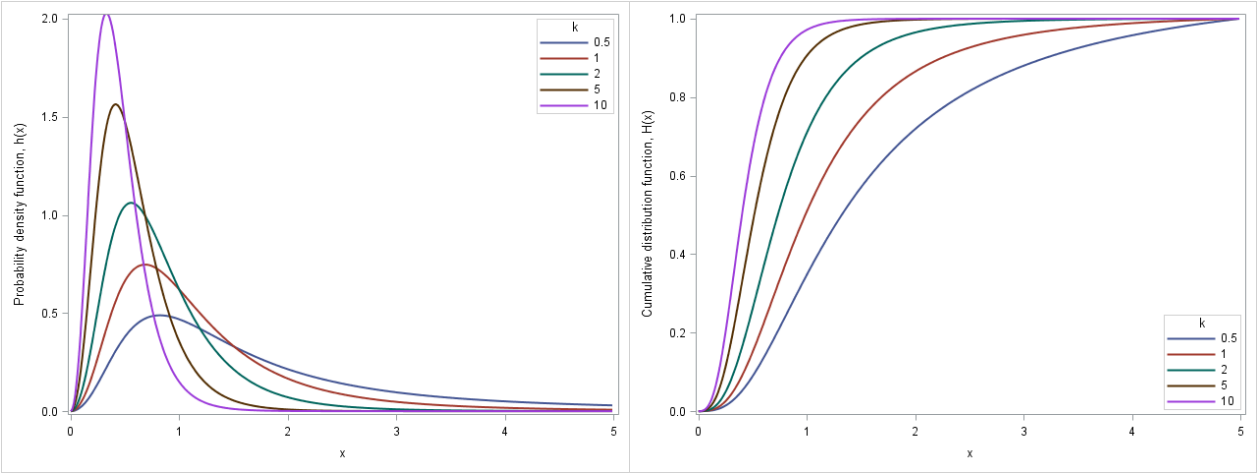


Figure 6.16: PDF (6.26) and CDF, respectively, with $\alpha = 3$, $\mu = 0$, $\sigma = 1$ and varying k

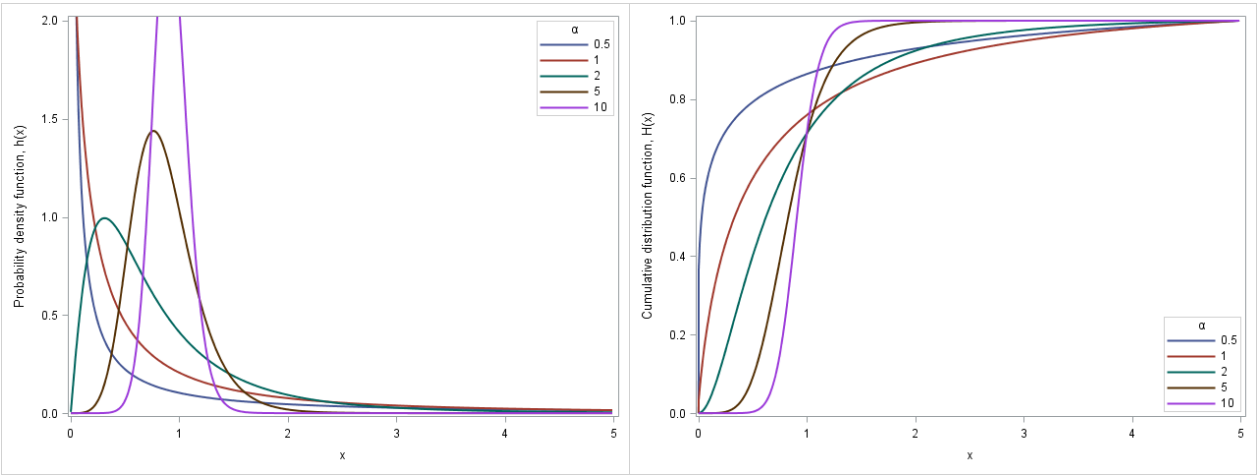


Figure 6.17: PDF (6.26) and CDF, respectively, with $k = 2$, $\mu = 0$, $\sigma = 1$ and varying α

6.6.2 MGF of the composite compound-Weibull/lognormal distribution

By substituting (4.17) and (5.1) into (3.5) and by (5.2), the MGF is given by:

$$\begin{aligned}
M_X(t) &= \int_\lambda M_{X|\lambda}(t)\, g(\lambda)\, d\lambda \\
&= \int_0^\infty \left\{ \sum_{i=0}^{\infty} \frac{t^i \lambda^i \Gamma\left(1+\frac{i}{\alpha}\right)\Gamma(k+i)}{i!\Gamma(k)} \right\} \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2} \right) d\lambda \\
&= \sum_{i=0}^{\infty} \frac{t^i \Gamma\left(1+\frac{i}{\alpha}\right)\Gamma(k+i)}{i!\Gamma(k)} \int_0^\infty \lambda^i \frac{1}{\sqrt{2\pi}\lambda\sigma} \exp\left(-\frac{(\log\lambda - \mu)^2}{2\sigma^2} \right) d\lambda \\
&= \sum_{i=0}^{\infty} \frac{t^i \Gamma\left(1+\frac{i}{\alpha}\right)\Gamma(k+i)}{i!\Gamma(k)} \mathbb{E}\left[\lambda^i\right] \\
&= \sum_{i=0}^{\infty} \frac{t^i \Gamma\left(1+\frac{i}{\alpha}\right)\Gamma(k+i)}{i!\Gamma(k)} \exp\left(i\mu + \frac{i^2\sigma^2}{2} \right), \quad t \geq 0.
\end{aligned} \tag{6.27}$$

6.6.3 Moments of the composite compound-Weibull/lognormal distribution

By substituting (4.18) and (5.1) into (3.5) and by (5.2), the r^{th} moment is given by:

$$\begin{aligned} m_r &= \int_\lambda m_{r|\lambda} g(\lambda)\, d\lambda \\ &= \int_\lambda \Gamma\left(1+\frac{r}{\alpha}\right)\frac{\lambda^r \Gamma(k+r)}{\Gamma(k)}\frac{1}{\sqrt{2\pi}\lambda\sigma}\exp\left(-\frac{(log\lambda-\mu)^2}{2\sigma^2}\right)d\lambda \\ &= \Gamma\left(1+\frac{r}{\alpha}\right)\frac{\Gamma(k+r)}{\Gamma(k)}\int_\lambda \lambda^r \frac{1}{\sqrt{2\pi}\lambda\sigma}\exp\left(-\frac{(log\lambda-\mu)^2}{2\sigma^2}\right)d\lambda \\ &= \Gamma\left(1+\frac{r}{\alpha}\right)\frac{\Gamma(k+r)}{\Gamma(k)}\mathbb{E}\left[\lambda^r\right] \\ &= \Gamma\left(1+\frac{r}{\alpha}\right)\frac{\Gamma(k+r)}{\Gamma(k)}\exp\left(r\mu+\frac{r^2\sigma^2}{2}\right),\ \ r\in\mathbb{Z}^+. \end{aligned} \tag{6.28}$$

By (6.28), the mean and variance can be derived as follows:

$$\begin{aligned} \mathbb{E}[X] &= m_1 \\ &= \Gamma\left(1+\frac{1}{\alpha}\right)\frac{\Gamma(k+1)}{\Gamma(k)}\exp\left(1\times\mu+\frac{1^2\times\sigma^2}{2}\right) \\ &= k\Gamma\left(1+\frac{1}{\alpha}\right)\exp\left(\mu+\frac{\sigma^2}{2}\right), \end{aligned} \tag{6.29}$$

and

$$\begin{aligned} var(X) &= m_2 - m_1^2 \\ &= \Gamma\left(1+\frac{2}{\alpha}\right)\frac{\Gamma(k+2)}{\Gamma(k)}\exp\left(2\times\mu+\frac{2^2\times\sigma^2}{2}\right) - \left[k\Gamma\left(1+\frac{1}{\alpha}\right)\exp\left(\mu+\frac{\sigma^2}{2}\right)\right]^2 \\ &= k(k+1)\Gamma\left(1+\frac{2}{\alpha}\right)\exp\left(2\mu+2\sigma^2\right) - k^2\Gamma^2\left(1+\frac{1}{\alpha}\right)\exp\left(2\mu+\sigma^2\right) \\ &= \exp\left(2\mu+\sigma^2\right)\left[\left(k^2+k\right)\Gamma\left(1+\frac{2}{\alpha}\right)\exp\left(\sigma^2\right) - k^2\Gamma^2\left(1+\frac{1}{\alpha}\right)\right]. \end{aligned} \tag{6.30}$$

6.7 Compound-Rayleigh fading with lognormal shadowing

6.7.1 PDF of the composite compound-Rayleigh/lognormal distribution

The PDF of the composite compound-Rayleigh/lognormal distribution with *shape* parameters $k>0$, $\mu\in\mathbb{R}$ and $\sigma>0$ is given by substituting (4.19) and (5.1) into (3.1):

$$\begin{aligned} h(x) &= \int_\lambda f(x|\lambda)\, g(\lambda)\, d\lambda \\ &= \int_0^\infty 2k\lambda^k x\left(\lambda+x^2\right)^{-(k+1)}\frac{1}{\sqrt{2\pi}\lambda\sigma}\exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right)d\lambda \\ &= \int_0^\infty \frac{\sqrt{2}k\lambda^{k-1}x\left(\lambda+x^2\right)^{-(k+1)}}{\sqrt{\pi}\sigma}\exp\left(-\frac{(\log\lambda-\mu)^2}{2\sigma^2}\right)d\lambda,\ \ x\geq 0, \end{aligned} \tag{6.31}$$

which is the PDF of the composite compound-Weibull/lognormal distribution with $\alpha=2$. Figures 6.18, 6.19 and 6.20 show the PDF and CDF with different parameter values.

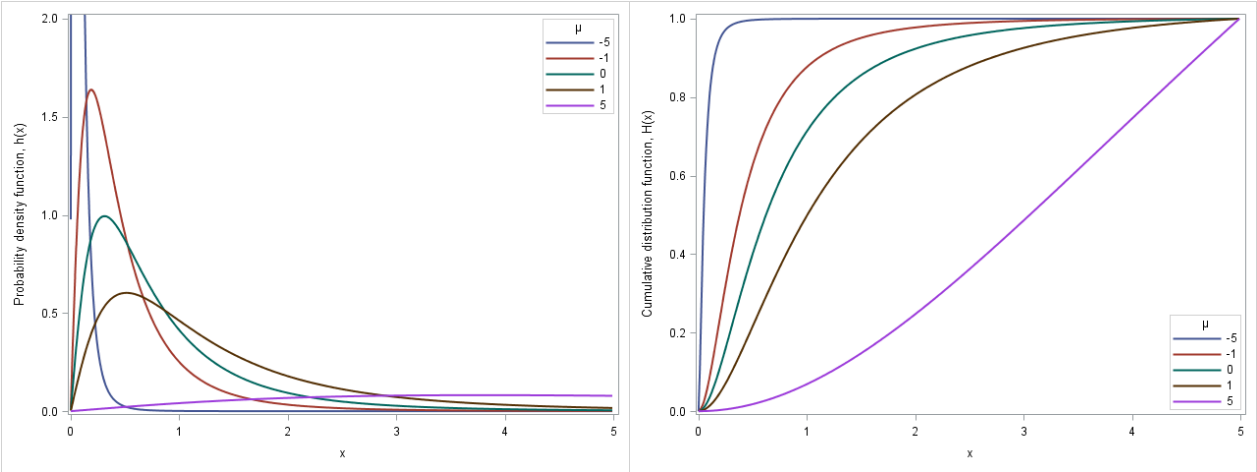


Figure 6.18: PDF (6.31) and CDF, respectively, with $k = 2$, $\sigma = 1$ and varying μ

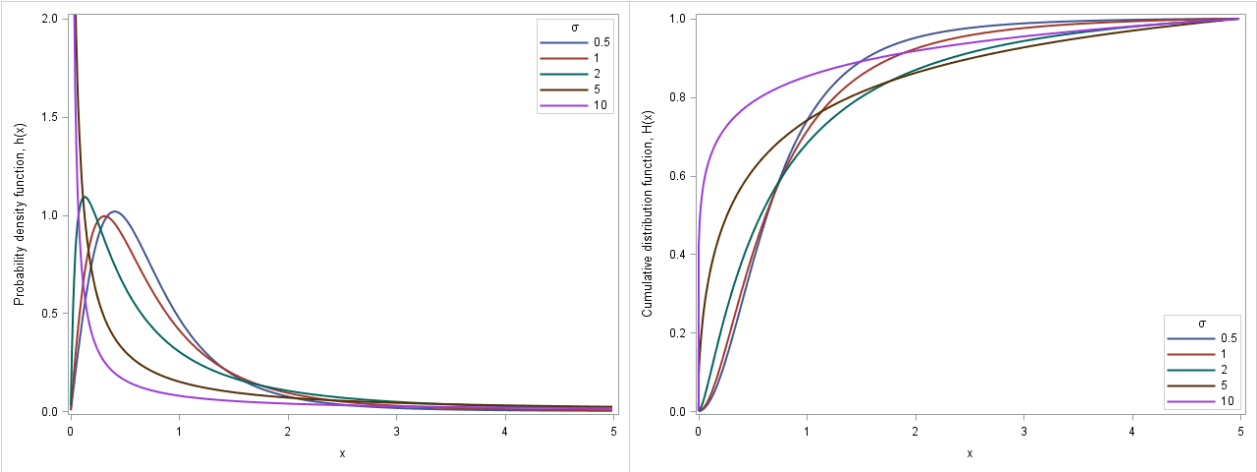


Figure 6.19: PDF (6.31) and CDF, respectively, with $k = 2$, $\mu = 0$ and varying σ

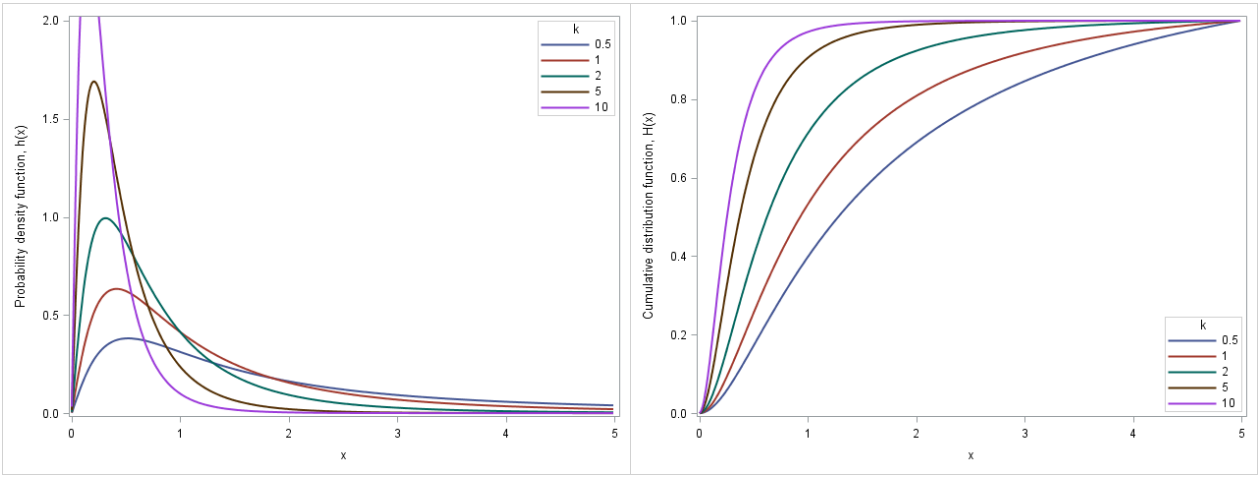


Figure 6.20: PDF (6.31) and CDF, respectively, with $\mu = 0$, $\sigma = 1$ and varying k

6.7.2 MGF of the composite compound-Rayleigh/lognormal distribution

The MGF is given by letting $\alpha = 2$ in equation (6.27):

$$M_X(t) = \sum_{i=0}^{\infty} \frac{t^i \Gamma\left(1+\frac{i}{2}\right)\Gamma(k+i)}{i!\Gamma(k)} \exp\left(i\mu + \frac{i^2\sigma^2}{2}\right), \quad t \geq 0. \tag{6.32}$$

6.7.3 Moments of the composite compound-Rayleigh/lognormal distribution

The r^{th} moment is given by letting $\alpha = 2$ in equation (6.28):

$$m_r = \Gamma\left(1+\frac{r}{2}\right)\frac{\Gamma(k+r)}{\Gamma(k)} \exp\left(r\mu + \frac{r^2\sigma^2}{2}\right), \quad r \in \mathbb{Z}^+. \tag{6.33}$$

By (6.33), the mean and variance can be derived as follows:

$$\begin{aligned} \mathbb{E}[X] &= m_1 \\ &= \Gamma\left(1+\frac{1}{2}\right)\frac{\Gamma(k+1)}{\Gamma(k)} \exp\left(1\times\mu + \frac{1^2\times\sigma^2}{2}\right) \\ &= \sqrt{\frac{\pi}{4}}k \exp\left(\mu + \frac{\sigma^2}{2}\right), \end{aligned} \tag{6.34}$$

and

$$\begin{aligned} var(X) &= m_2 - m_1^2 \\ &= \Gamma\left(1+\frac{2}{2}\right)\frac{\Gamma(k+2)}{\Gamma(k)} \exp\left(2\times\mu + \frac{2^2\times\sigma^2}{2}\right) - \left[\sqrt{\frac{\pi}{4}}k \exp\left(\mu + \frac{\sigma^2}{2}\right)\right]^2 \\ &= k(k+1)\exp\left(2\mu + 2\sigma^2\right) - \frac{\pi}{4}k^2\exp\left(2\mu+\sigma^2\right) \\ &= \exp\left(2\mu+\sigma^2\right)\left[\left(k^2+k\right)\exp\left(\sigma^2\right) - \frac{\pi}{4}k^2\right]. \end{aligned} \tag{6.35}$$

7 Application to wireless communication systems

7.1 Amplitude of of composite fading/shadowing channels

In the context of wireless communications, amplitude is defined as the wave deviation from the mean level of a signal [4]. The distributions discussed in section 4 are used to model the amplitude of the signal, exposed to fading, in a wireless communication system [13]. In a system exposed to fading and shadowing, amplitude is modelled using the composite distributions discussed in section 6.

7.2 Signal-to-noise ratio of composite fading/shadowing channels

The SNR of a fading signal in a shadowed environment is modelled by the composite fading/shadowing distributions discussed in section 6, whereby SNR, denoted by ω, has the fading distribution and average SNR, denoted by $\bar{\omega}$, has the shadowing distribution [13]. Table 7.2 shows how the SNR distribution of the composite channel is formulated.

7.2.1 Appropriateness of SNR distribution

In order to assess whether the distributions discussed in this section are good SNR distributions, each must meet the following requirement: the fading (scale) parameter λ must be equal to the mean-square value of the amplitude, i.e. $\lambda = \mathbb{E}\left[X^2\right] = m_2$, where X is the fading amplitude with PDF $f(x)$ [13]. Table 7.1 gives the results of this test. The m_2 for each fading distribution was obtained using the r^{th} moments defined in section 4.

Fading distribution	m_2	SNR distribution
Exponential	$2\lambda^2$	No
Gamma	$\left(k^2+k\right)\lambda^2$	No
Rayleigh	λ	Yes
Weibull	$\Gamma\left(1+\frac{2}{\alpha}\right)\lambda^2$	No
Nakagami-m	λ	Yes
Compound-Weibull	$\left(k^2+k\right)\Gamma\left(1+\frac{2}{\alpha}\right)\lambda^2$	No
Compound-Rayleigh	$\left(k^2+k\right)\lambda^2$ or $k\lambda$ (reparameterised)	Possibly

Table 7.1: Testing appropriateness of SNR distributions

The Rayleigh and Nakagami-m distributions fit the requirement (as suggested in [13]) and the result for the compound-Rayleigh distribution gives speculative suggestion that this may be a suitable fading distribution. Therefore, only the SNR distributions of the Rayleigh/lognormal, Nakagami-m/lognormal and compound-Rayleigh/lognormal distributions are investigated.

7.2.2 Mean SNR ($\bar{\omega}$) distribution

$s(\omega\|\bar{\omega})$	PDF of the conditional *fading* SNR distribution of $\omega\|\bar{\omega}$
$g(\bar{\omega})$	PDF of the *shadowing* SNR distribution of $\bar{\omega}$
$p(\omega)$	PDF of the *composite* SNR distribution of ω

Table 7.2: Construction of the composite SNR distribution

The SNR is measured in decibels, therefore the lognormal shadowing distribution must be formulated in terms of decibels. The PDF of the average SNR distribution is given by [13]:

$$g(\bar{\omega}) = \frac{\xi}{\sqrt{2\pi}\bar{\omega}\sigma}\exp\left(-\frac{\left(10\log_{10}\bar{\omega}-\mu\right)^2}{2\sigma^2}\right),\ \bar{\omega}>0, \tag{7.1}$$

where $\xi = \frac{10}{\log 10} = 4.3429$, and $\mu \in \mathbb{R}$ and $\sigma^2 > 0$ are the *mean* and *variance* of $10 \log_{10} \bar{\omega}$, respectively [13]. The r^{th} moment of this lognormal distribution is given by [13]:

$$\mathbb{E}\left[\bar{\omega}^r\right] = \exp\left(\frac{r\mu}{\xi} + \frac{r^2\sigma^2}{2\xi^2}\right), \ r \in \mathbb{Z}^+,$$

and therefore with mean and variance (by substitution of r and since $var\left(\bar{\omega}\right) = \mathbb{E}\left[\bar{\omega}^2\right] - \left(\mathbb{E}\left[\bar{\omega}\right]\right)^2$):

$$\mathbb{E}\left[\bar{\omega}\right] = \exp\left(\frac{\mu}{\xi} + \frac{\sigma^2}{2\xi^2}\right), \tag{7.2}$$

and

$$var\left(\bar{\omega}\right) = \exp\left(\frac{2\mu}{\xi} + \frac{\sigma^2}{\xi^2}\right)\left[\exp\left(\frac{\sigma^2}{\xi^2}\right) - 1\right]. \tag{7.3}$$

The PDF of the SNR, ω, distribution of any fading channel X with PDF $f\left(x\right)$ is derived using the transformation, $\omega = \frac{\bar{\omega}X^2}{\lambda}$ [13]. Therefore:

$$s\left(\omega\right) = \frac{f_X\left(\sqrt{\frac{\lambda\omega}{\bar{\omega}}}\right)}{2\sqrt{\frac{\omega\bar{\omega}}{\lambda}}} \tag{7.4}$$

The mean and variance of the SNR can also be derived using this transformation and the following theorem:

Theorem 7

Let X be a random variable with PDF $h\left(x\right)$ and let ω be SNR random variable with PDF $h\left(\omega\right)$. Let $\omega = \frac{\bar{\omega}X^2}{\lambda}$. Then

$$\mathbb{E}\left[\omega\right] = \frac{m_2}{\lambda}\exp\left(\frac{\mu}{\xi} + \frac{\sigma^2}{2\xi^2}\right) \tag{7.5}$$

and

$$var\left(\omega\right) = \frac{1}{\lambda^2}\exp\left(\frac{2\mu}{\xi} + \frac{\sigma^2}{\xi^2}\right)\left[m_4\exp\left(\frac{2\mu}{\xi} + \frac{\sigma^2}{\xi^2}\right) - m_2^2\right] \tag{7.6}$$

Proof. The expected value of ω is

$$\begin{aligned}
\mathbb{E}\left[\omega\right] &= \mathbb{E}\left[\frac{\bar{\omega}X^2}{\lambda}\right] \\
&= \int_{\bar{\omega}} \mathbb{E}\left[\frac{\bar{\omega}X^2}{\lambda}|\bar{\omega}\right] g\left(\bar{\omega}\right) d\bar{\omega}, \quad \text{by Theorem 1,} \\
&= \int_{\bar{\omega}} \frac{1}{\lambda}\mathbb{E}\left[X^2\right] \bar{\omega} g\left(\bar{\omega}\right) d\bar{\omega} \\
&= \frac{m_2}{\lambda}\int_{\bar{\omega}} \bar{\omega} g\left(\bar{\omega}\right) d\bar{\omega} \\
&= \frac{m_2}{\lambda}\mathbb{E}\left[\bar{\omega}\right] \\
&= \frac{m_2}{\lambda}\exp\left(\frac{\mu}{\xi} + \frac{\sigma^2}{2\xi^2}\right), \quad \text{by equation 7.2,}
\end{aligned}$$

and the variance of ω is

$$\begin{aligned}
var\left(\omega\right) &= \mathbb{E}\left[\omega^2\right] - \left(\mathbb{E}\left[\omega\right]\right)^2 \\
&= \mathbb{E}\left[\left(\frac{\bar{\omega}X^2}{\lambda}\right)^2\right] - \left(\frac{m_2}{\lambda}\exp\left(\frac{\mu}{\xi}+\frac{\sigma^2}{2\xi^2}\right)\right)^2 \\
&= \int_{\bar{\omega}} \mathbb{E}\left[\left(\frac{\bar{\omega}X^2}{\lambda}\right)^2 |\bar{\omega}\right] g\left(\bar{\omega}\right) d\bar{\omega} - \frac{m_2^2}{\lambda^2}\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right), \quad \text{by Theorem 1,} \\
&= \int_{\bar{\omega}} \frac{m_4}{\lambda^2}\bar{\omega}^2 g\left(\bar{\omega}\right) d\bar{\omega} - \frac{m_2^2}{\lambda^2}\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right) \\
&= \frac{m_4}{\lambda^2}\int_{\bar{\omega}} \bar{\omega}^2 g\left(\bar{\omega}\right) d\bar{\omega} - \frac{m_2^2}{\lambda^2}\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right) \\
&= \frac{1}{\lambda^2}\left[m_4\mathbb{E}\left[\bar{\omega}^2\right] - m_2^2\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\right] \\
&= \frac{1}{\lambda^2}\left[m_4\exp\left(\frac{2\mu}{\xi}+\frac{2\sigma^2}{\xi^2}\right) - m_2^2\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\right], \quad \text{by equation 7.3,} \\
&= \frac{1}{\lambda^2}\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[m_4\exp\left(\frac{\sigma^2}{\xi^2}\right) - m_2^2\right],
\end{aligned}$$

which proves the result. ■

7.2.3 SNR distribution of the composite Rayleigh/lognormal channel

By (7.4), the PDF of the SNR of the Rayleigh fading component is given by:

$$\begin{aligned}
s\left(\omega\right) &= \frac{f_X\left(\sqrt{\frac{\lambda\omega}{\bar{\omega}}}\right)}{2\sqrt{\frac{\omega\bar{\omega}}{\lambda}}} \\
&= \frac{2\sqrt{\frac{\lambda\omega}{\bar{\omega}}}}{\lambda}\exp\left[-\frac{\left(\sqrt{\frac{\lambda\omega}{\bar{\omega}}}\right)^2}{\lambda}\right]\frac{\sqrt{\lambda}}{2\sqrt{\omega\bar{\omega}}} \\
&= \frac{2\sqrt{\omega}}{\sqrt{\lambda\bar{\omega}}}\exp\left[-\frac{\left(\sqrt{\frac{\lambda\omega}{\bar{\omega}}}\right)^2}{\lambda}\right]\frac{\sqrt{\lambda}}{2\sqrt{\omega\bar{\omega}}} \\
&= \frac{1}{\bar{\omega}}\exp\left(-\frac{\omega}{\bar{\omega}}\right), \ \omega > 0. \qquad (7.7)
\end{aligned}$$

Therefore the SNR distribution of Rayleigh fading is exponential with *scale* parameter $\bar{\omega} > 0$. This is the result obtained in [13].

The PDF of the composite Rayleigh/lognormal SNR distribution is given by substituting (7.7) and (7.1) into (3.1):

$$
\begin{aligned}
p(\omega) &= \int_{\bar{\omega}} f(\omega|\bar{\omega})\, g(\bar{\omega})\, d\bar{\omega} \\
&= \int_0^\infty \frac{1}{\bar{\omega}} \exp\left(-\frac{\omega}{\bar{\omega}}\right) \frac{\xi}{\sqrt{2\pi}\bar{\omega}\sigma} \exp\left(-\frac{(10\log_{10}\bar{\omega}-\mu)^2}{2\sigma^2}\right) d\bar{\omega} \\
&= \int_0^\infty \frac{\xi}{\sqrt{2\pi}\bar{\omega}^2\sigma} \exp\left(-\frac{\omega}{\bar{\omega}} - \frac{(10\log_{10}\bar{\omega}-\mu)^2}{2\sigma^2}\right) d\bar{\omega},\ \ \bar{\omega} > 0, \qquad (7.8)
\end{aligned}
$$

which shows that the SNR distribution is composite exponential/lognormal with *shape* parameters $\mu \in \mathbb{R}$ and $\sigma > 0$ (in decibels). This distribution is described in [13].

The mean and variance of the SNR is given by substituting (4.8) into (7.5) and (7.6), respectively:

$$
\begin{aligned}
\mathbb{E}[\omega] &= \frac{m_2}{\lambda} \exp\left(\frac{\mu}{\xi} + \frac{\sigma^2}{2\xi^2}\right) \\
&= \frac{1}{\lambda}\lambda^{\frac{2}{2}}\Gamma\left(1+\frac{2}{2}\right) \exp\left(\frac{\mu}{\xi} + \frac{\sigma^2}{2\xi^2}\right) \\
&= \exp\left(\frac{\mu}{\xi} + \frac{\sigma^2}{2\xi^2}\right),
\end{aligned}
$$

and

$$
\begin{aligned}
var(\omega) &= \frac{1}{\lambda^2} \exp\left(\frac{2\mu}{\xi} + \frac{\sigma^2}{\xi^2}\right) \left[m_4 \exp\left(\frac{\sigma^2}{\xi^2}\right) - m_2^2\right] \\
&= \frac{1}{\lambda^2} \exp\left(\frac{2\mu}{\xi} + \frac{\sigma^2}{\xi^2}\right) \left[\lambda^{\frac{4}{2}}\Gamma\left(1+\frac{4}{2}\right) \exp\left(\frac{\sigma^2}{\xi^2}\right) - \left(\lambda^{\frac{2}{2}}\Gamma\left(1+\frac{2}{2}\right)\right)^2\right] \\
&= \exp\left(\frac{2\mu}{\xi} + \frac{\sigma^2}{\xi^2}\right) \left[2\exp\left(\frac{\sigma^2}{\xi^2}\right) - 1\right].
\end{aligned}
$$

7.2.4 SNR distribution of the composite Nakagami-m/lognormal channel

By (7.4), the PDF of the SNR of the Nakagami-m fading component is given by:

$$
\begin{aligned}
s(\omega) &= \frac{f_X\left(\sqrt{\frac{\lambda\omega}{\bar{\omega}}}\right)}{2\sqrt{\frac{\omega\bar{\omega}}{\lambda}}} \\
&= \frac{2m^m\left(\sqrt{\frac{\lambda\omega}{\bar{\omega}}}\right)^{2m-1}}{\Gamma(m)\lambda^m}\exp\left(-\frac{m\left(\sqrt{\frac{\lambda\omega}{\bar{\omega}}}\right)^2}{\lambda}\right)\frac{\sqrt{\lambda}}{2\sqrt{\omega\bar{\omega}}} \\
&= \frac{2m^m\lambda^m\omega^m\sqrt{\bar{\omega}}}{\Gamma(m)\lambda^m\bar{\omega}^m\sqrt{\lambda\omega}}\exp\left(-\frac{m\left(\sqrt{\frac{\lambda\omega}{\bar{\omega}}}\right)^2}{\lambda}\right)\frac{\sqrt{\lambda}}{2\sqrt{\omega\bar{\omega}}} \\
&= \frac{2m^m\omega^m}{2\Gamma(m)\bar{\omega}^m\omega}\exp\left(-\frac{m\left(\sqrt{\frac{\lambda\omega}{\bar{\omega}}}\right)^2}{\lambda}\right) \\
&= \frac{m^m\omega^{m-1}}{\Gamma(m)\bar{\omega}^m}\exp\left(-\frac{m\omega}{\bar{\omega}}\right), \ \omega > 0. \qquad (7.9)
\end{aligned}
$$

Therefore the SNR distribution of Nakagami-m fading is gamma with *shape* and *scale* parameter $m > 0$ and $\frac{\bar{\omega}}{m} > 0$. This is the result obtained in [13].

The PDF of the composite Nakagami-m/lognormal SNR distribution is given by substituting (7.9) and (7.1) into (3.1):

$$
\begin{aligned}
p(\omega) &= \int_{\bar{\omega}} f(\omega|\bar{\omega})\, g(\bar{\omega})\, d\bar{\omega} \\
&= \int_0^\infty \frac{m^m\omega^{m-1}}{\Gamma(m)\bar{\omega}^m}\exp\left(-\frac{m\omega}{\bar{\omega}}\right)\frac{\xi}{\sqrt{2\pi}\bar{\omega}\sigma}\exp\left(-\frac{(10\log_{10}\bar{\omega}-\mu)^2}{2\sigma^2}\right)d\bar{\omega} \\
&= \int_0^\infty \frac{\xi m^m\omega^{m-1}}{\sqrt{2\pi}\Gamma(m)\bar{\omega}^{m+1}\sigma}\exp\left(-\frac{m\omega}{\bar{\omega}}-\frac{(10\log_{10}\bar{\omega}-\mu)^2}{2\sigma^2}\right)d\bar{\omega}, \ \bar{\omega} > 0, \qquad (7.10)
\end{aligned}
$$

which shows that the SNR distribution is composite gamma/lognormal with *shape* parameters $m > 0$, $\mu \in \mathbb{R}$ and $\sigma > 0$ (in decibels). This distribution is described in [13].

The mean and variance of the SNR is given by substituting (4.14) into (7.5) and (7.6), respectively:

$$
\begin{aligned}
\mathbb{E}[\omega] &= \frac{m_2}{\lambda}\exp\left(\frac{\mu}{\xi}+\frac{\sigma^2}{2\xi^2}\right) \\
&= \frac{1}{\lambda}\frac{\lambda^{\frac{2}{2}}\Gamma\left(m+\frac{2}{2}\right)}{m^{\frac{2}{2}}\Gamma(m)}\exp\left(\frac{\mu}{\xi}+\frac{\sigma^2}{2\xi^2}\right) \\
&= \frac{m!}{m(m-1)!}\exp\left(\frac{\mu}{\xi}+\frac{\sigma^2}{2\xi^2}\right) \\
&= \exp\left(\frac{\mu}{\xi}+\frac{\sigma^2}{2\xi^2}\right),
\end{aligned}
$$

and

$$\begin{aligned} var\left(\omega\right) &= \frac{1}{\lambda^2}\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[m_4\exp\left(\frac{\sigma^2}{\xi^2}\right)-m_2^2\right] \\ &= \frac{1}{\lambda^2}\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[\frac{\lambda^{\frac{4}{2}}\Gamma\left(m+\frac{4}{2}\right)}{m^{\frac{4}{2}}\Gamma\left(m\right)}\exp\left(\frac{\sigma^2}{\xi^2}\right)-\left(\frac{\lambda^{\frac{2}{2}}\Gamma\left(m+\frac{2}{2}\right)}{m^{\frac{2}{2}}\Gamma\left(m\right)}\right)^2\right] \\ &= \exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[\frac{(m+1)!}{m^2\,(m-1)!}\exp\left(\frac{\sigma^2}{\xi^2}\right)-1\right] \\ &= \exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[\frac{m+1}{m}\exp\left(\frac{\sigma^2}{\xi^2}\right)-1\right]. \end{aligned}$$

7.2.5 SNR distribution of the composite compound-Rayleigh/lognormal channel

By (7.4), the PDF of the SNR of the compound-Rayleigh fading component is given by:

$$\begin{aligned} s\left(\omega\right) &= \frac{f_X\left(\sqrt{\frac{\lambda\omega}{\bar{\omega}}}\right)}{2\sqrt{\frac{\omega\bar{\omega}}{\lambda}}} \\ &= 2k\lambda^k\sqrt{\frac{\lambda\omega}{\bar{\omega}}}\left[\lambda+\left(\sqrt{\frac{\lambda\omega}{\bar{\omega}}}\right)^2\right]^{-(k+1)}\frac{\sqrt{\lambda}}{2\sqrt{\omega\bar{\omega}}} \\ &= \frac{k}{\bar{\omega}}\left(1+\frac{\omega}{\bar{\omega}}\right)^{-(k+1)},\ \ \omega>0, \end{aligned} \tag{7.11}$$

with *shape* and *scale* parameters $k>0$ and $\bar{\omega}>0$, respectively.

The PDF of the composite compound-Rayleigh/lognormal SNR distribution is given by substituting (7.11) and (7.1) into (3.1):

$$\begin{aligned} p\left(\omega\right) &= \int_{\bar{\omega}} f\left(\omega|\bar{\omega}\right) g\left(\bar{\omega}\right) d\bar{\omega} \\ &= \int_0^\infty \frac{k}{\bar{\omega}}\left(1+\frac{\omega}{\bar{\omega}}\right)^{-(k+1)}\frac{\xi}{\sqrt{2\pi}\bar{\omega}\sigma}\exp\left(-\frac{\left(10\log_{10}\bar{\omega}-\mu\right)^2}{2\sigma^2}\right)d\bar{\omega} \\ &= \int_0^\infty \frac{\xi k}{\sqrt{2\pi}\bar{\omega}^2\sigma}\left(1+\frac{\omega}{\bar{\omega}}\right)^{-(k+1)}\exp\left(-\frac{\left(10\log_{10}\bar{\omega}-\mu\right)^2}{2\sigma^2}\right)d\bar{\omega},\ \ \bar{\omega}>0, \end{aligned} \tag{7.12}$$

with *shape* parameters $k>0$, $\mu\in\mathbb{R}$ and $\sigma>0$ (in decibels).

The mean and variance of the SNR is given by substituting (4.21) into (7.5) and (7.6), respectively:

$$\begin{aligned}
\mathbb{E}[\omega] &= \frac{m_2}{\lambda}\exp\left(\frac{\mu}{\xi}+\frac{\sigma^2}{2\xi^2}\right) \\
&= \frac{1}{\lambda}\Gamma\left(1+\frac{2}{2}\right)\frac{\lambda^2\Gamma(k+2)}{\Gamma(k)}\exp\left(\frac{\mu}{\xi}+\frac{\sigma^2}{2\xi^2}\right) \\
&= \lambda k(k+1)\exp\left(\frac{\mu}{\xi}+\frac{\sigma^2}{2\xi^2}\right) \\
&= \lambda\left(k^2+k\right)\exp\left(\frac{\mu}{\xi}+\frac{\sigma^2}{2\xi^2}\right), \qquad (7.13)
\end{aligned}$$

and

$$\begin{aligned}
var(\omega) &= \frac{1}{\lambda^2}\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[m_4\exp\left(\frac{\sigma^2}{\xi^2}\right)-m_2^2\right] \\
&= \frac{1}{\lambda^2}\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[\Gamma\left(1+\frac{4}{2}\right)\frac{\lambda^4\Gamma(k+4)}{\Gamma(k)}\exp\left(\frac{\sigma^2}{\xi^2}\right)-\mathbb{E}\left[\Gamma\left(1+\frac{2}{2}\right)\frac{\lambda^2\Gamma(k+2)}{\Gamma(k)}\right]^2\right] \\
&= \lambda^2\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[2!\times\frac{\Gamma(k+4)}{\Gamma(k)}\exp\left(\frac{\sigma^2}{\xi^2}\right)-\left(k^2+k\right)^2\right] \\
&= \lambda^2\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[\frac{2\Gamma(k+4)}{\Gamma(k)}\exp\left(\frac{\sigma^2}{\xi^2}\right)-\left(k^2+k\right)^2\right]. \qquad (7.14)
\end{aligned}$$

The expressions in (7.13) and (7.14) contradict the requirement for this to be a good SNR distribution. After reparameterisation the r^{th} moment of the compound-Rayleigh is derived similarly to (4.21), using (4.8):

$$\begin{aligned}
m_r &= \int_\theta m_{r|\theta}g(\theta)\,d\theta \\
&= \int_0^\infty \theta^{\frac{r}{2}}\Gamma\left(1+\frac{r}{2}\right)\frac{1}{\Gamma(k)\lambda^k}\theta^{k-1}\exp\left(-\frac{\theta}{\lambda}\right)d\theta \\
&= \Gamma\left(1+\frac{r}{2}\right)\int_0^\infty \theta^{\frac{r}{2}}\frac{1}{\Gamma(k)\lambda^k}\theta^{k-1}\exp\left(-\frac{\theta}{\lambda}\right)d\theta \\
&= \Gamma\left(1+\frac{r}{2}\right)\mathbb{E}\left[\theta^{\frac{r}{2}}\right] \\
&= \Gamma\left(1+\frac{r}{2}\right)\frac{\lambda^{\frac{r}{2}}\Gamma\left(k+\frac{r}{2}\right)}{\Gamma(k)}, \;\; r\in\mathbb{Z}^+. \qquad (7.15)
\end{aligned}$$

The mean and variance of the SNR of the reparameterised compound-Rayleigh/lognormal are derived similarly to (7.13) and (7.14):

$$\begin{aligned}
\mathbb{E}[\omega] &= \frac{m_2}{\lambda}\exp\left(\frac{\mu}{\xi}+\frac{\sigma^2}{2\xi^2}\right) \\
&= \frac{1}{\lambda}\Gamma\left(1+\frac{2}{2}\right)\frac{\lambda^{\frac{2}{2}}\Gamma\left(k+\frac{2}{2}\right)}{\Gamma(k)}\exp\left(\frac{\mu}{\xi}+\frac{\sigma^2}{2\xi^2}\right) \\
&= k\exp\left(\frac{\mu}{\xi}+\frac{\sigma^2}{2\xi^2}\right),
\end{aligned}$$

and

$$\begin{aligned} var(\omega) &= \frac{1}{\lambda^2}\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[m_4\exp\left(\frac{\sigma^2}{\xi^2}\right)-m_2^2\right] \\ &= \frac{1}{\lambda^2}\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[\Gamma\left(1+\frac{4}{2}\right)\frac{\lambda^{\frac{4}{2}}\Gamma\left(k+\frac{4}{2}\right)}{\Gamma(k)}\exp\left(\frac{\sigma^2}{\xi^2}\right)-\left[\Gamma\left(1+\frac{2}{2}\right)\frac{\lambda^{\frac{2}{2}}\Gamma\left(k+\frac{2}{2}\right)}{\Gamma(k)}\right]^2\right] \\ &= \frac{1}{\lambda^2}\exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[2k(k+1)\lambda^2\exp\left(\frac{\sigma^2}{\xi^2}\right)-(k\lambda)^2\right] \\ &= \exp\left(\frac{2\mu}{\xi}+\frac{\sigma^2}{\xi^2}\right)\left[2k(k+1)\exp\left(\frac{\sigma^2}{\xi^2}\right)-k^2\right]. \end{aligned}$$

7.3 Outage probability (P_{out})

The SNR distribution of the composite distribution is used to find the outage probability of the corresponding channel. The outage probability is equal to the CDF of the target SNR [11, 13, 18, 19]:

$$P_{out} = P(\omega_0)$$

where ω_0 is the threshold SNR value.

The rest of this section shows the behaviour of the SNR distributions in section 7.2 from the outage probability. For a given threshold ω_0, the parameter choice which results in a lower outage probability is deemed preferable.

7.3.1 P_{out} of the composite Rayleigh/lognormal channel (7.8)

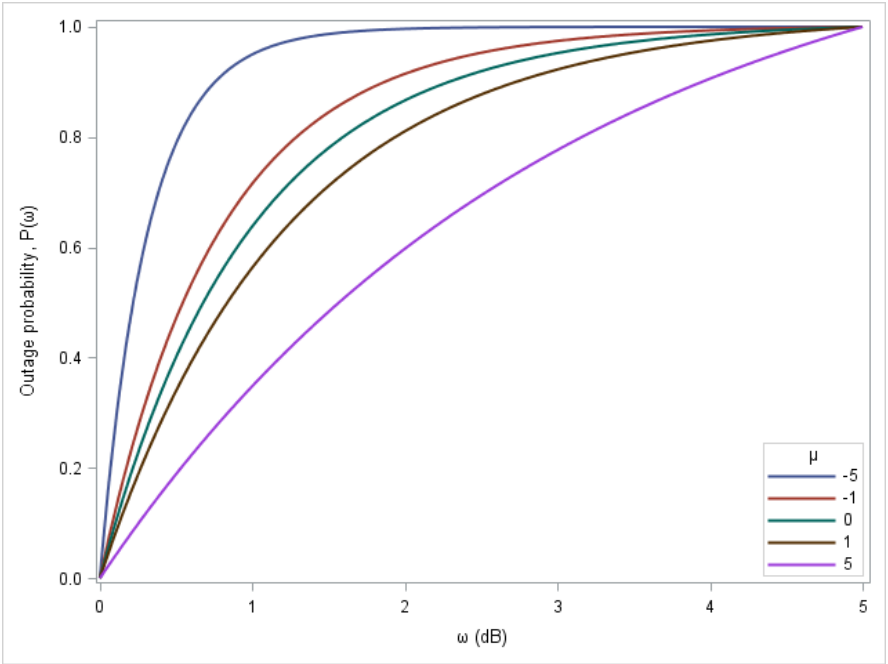


Figure 7.1: Outage probability with $\sigma = 1$ and varying μ

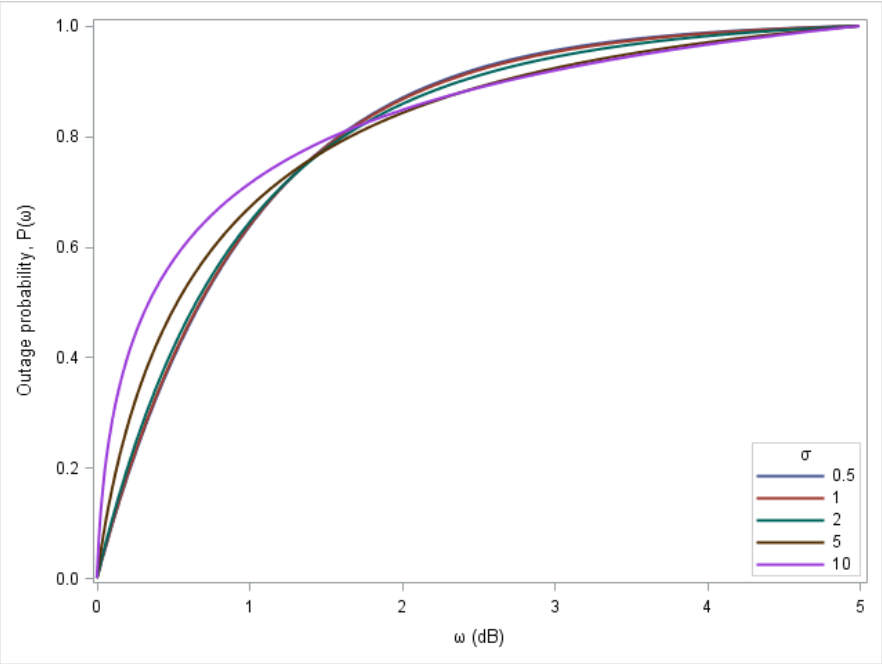


Figure 7.2: Outage probability with $\mu = 0$ and varying σ

7.3.2 P_{out} of the composite Nakagami-m/lognormal channel (7.10)

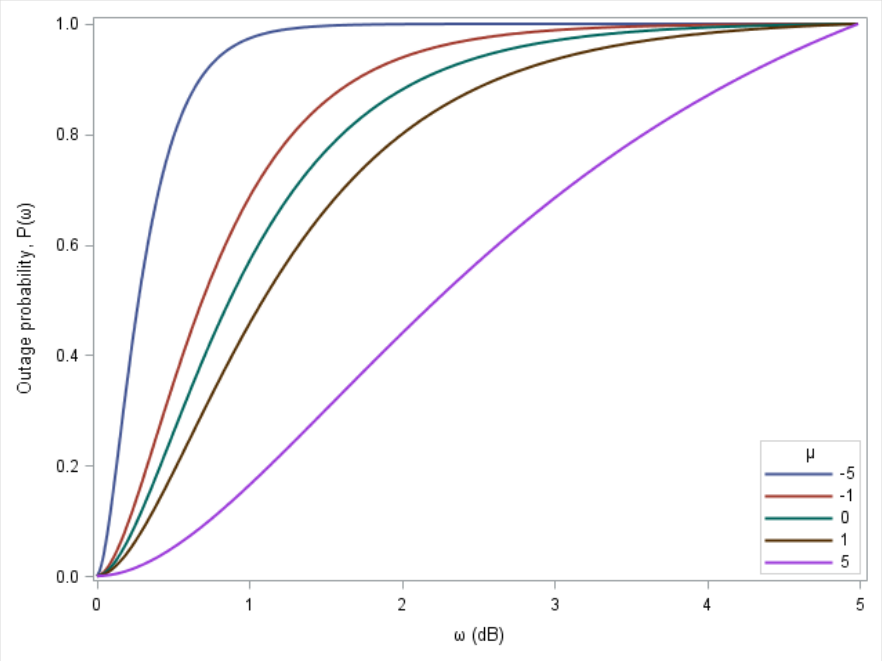


Figure 7.3: Outage probability with $m = 2$, $\sigma = 1$ and varying μ

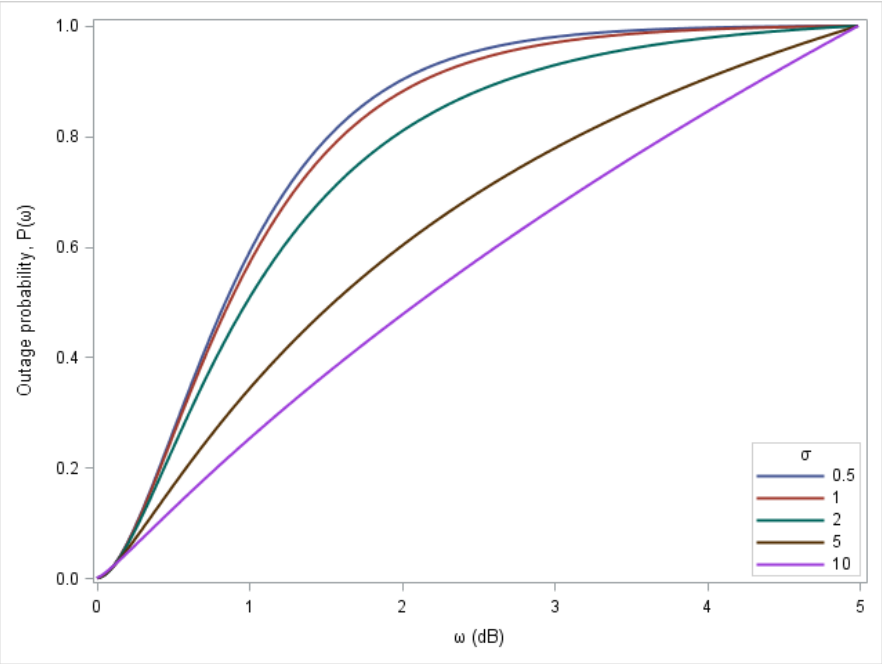


Figure 7.4: Outage probability with $m = 2$, $\mu = 0$ and varying σ

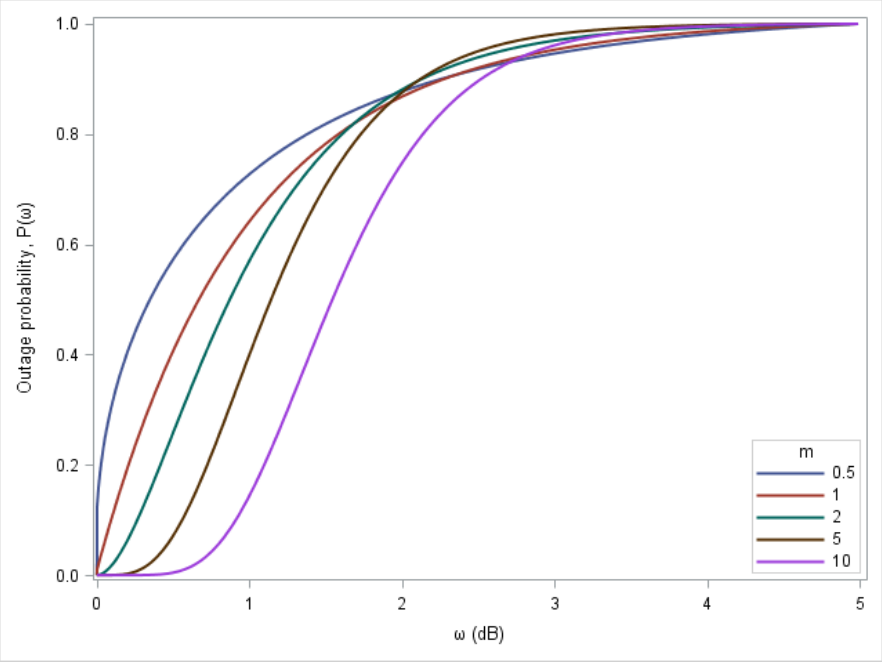


Figure 7.5: Outage probability with $\mu = 0$, $\sigma = 1$ and varying m

7.3.3 P_{out} of the composite compound-Rayleigh/lognormal channel (7.12)

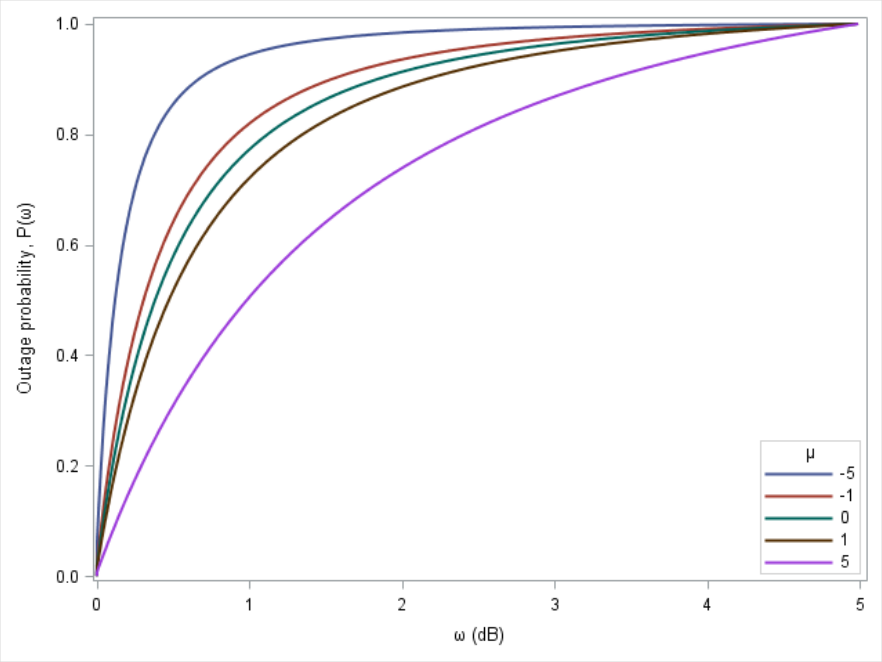


Figure 7.6: Outage probability with $k = 2$, $\sigma = 1$ and varying μ

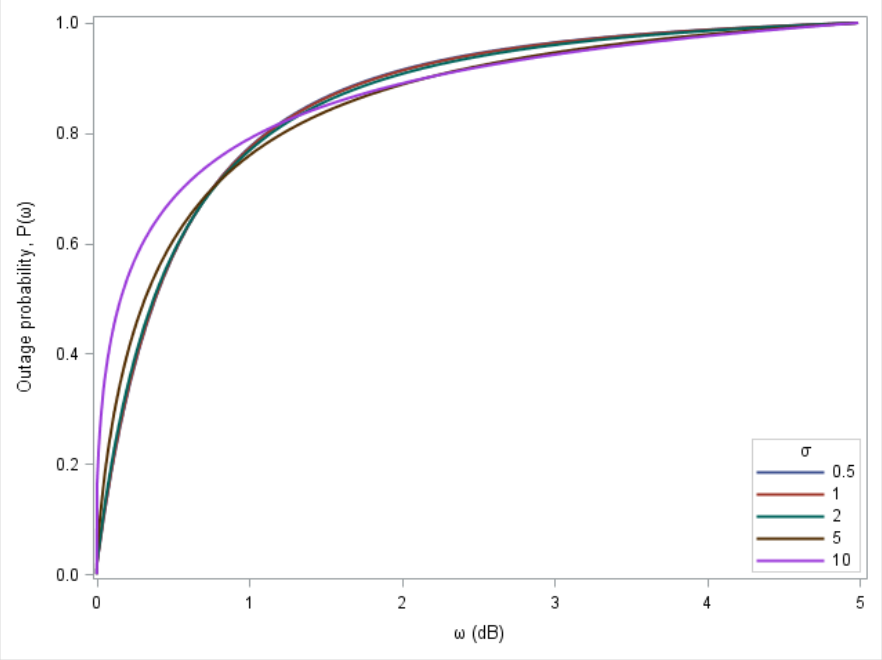


Figure 7.7: Outage probability with $k = 2$, $\mu = 0$ and varying σ

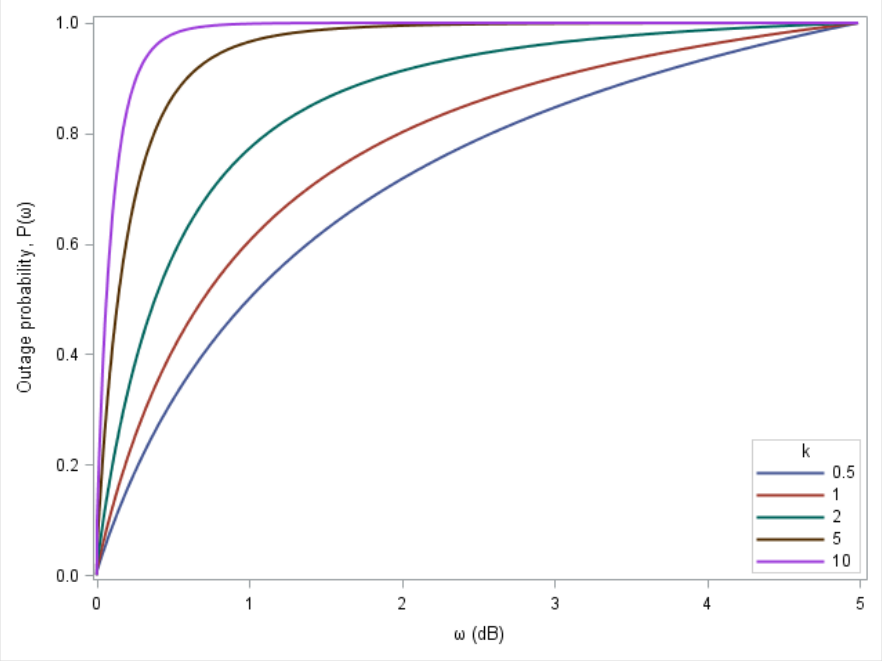


Figure 7.8: Outage probability with $\mu = 0$, $\sigma = 1$ and varying k

7.3.4 Appropriateness of the compound-Rayleigh distribution to assess outage probability

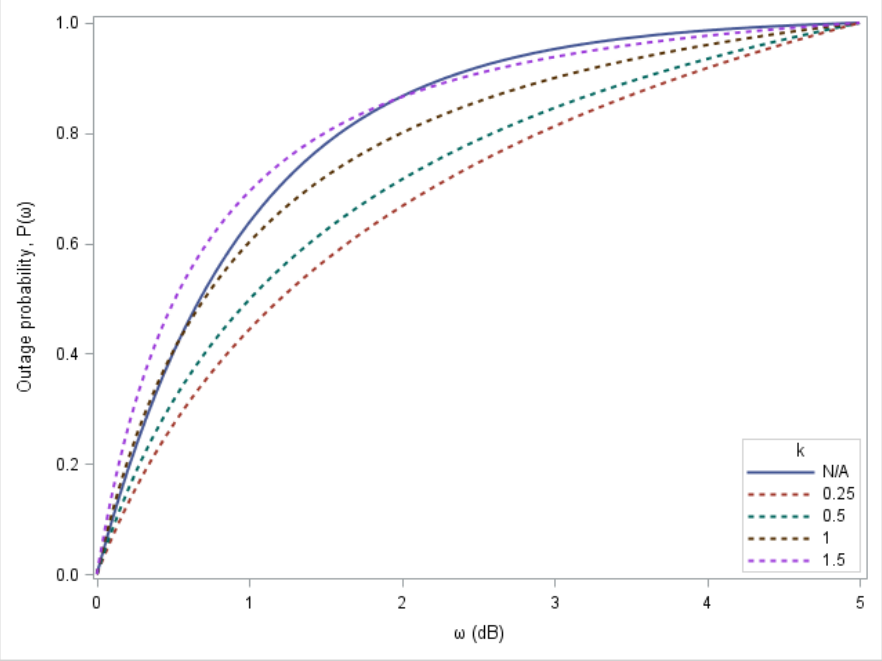


Figure 7.9: Outage probability with $\mu = 0$, $\sigma = 1$ and varying k

The outage probability of the compound-Rayleigh/lognormal distribution is lower than that of the Rayleigh/lognormal distribution for values of $k < 1.5$. This distribution may be suitable with an appropriate choice of k.

8 Conclusion

This study gave a full breakdown of the construction of composite fading/shadowing distributions. Distributions belonging to the regular exponential class and two new distributions were described and construction in the composite paradigm were motivated. The PDF, MGF and moments were derived for each of these composite fading/shadowing distributions. The theory and application of these new distributions have made contributions to the wireless communications arena. The literature was further enriched with derivation of the corresponding signal-to-noise ratio distributions for some of these distributions, namely the composite Rayleigh/lognormal, Nakagami-m/lognormal and compound-Rayleigh/lognormal distributions. These distributions have been comparatively investigated in terms of their outage probability. It is recommended that the compound-Rayleigh/lognormal distribution can be investigated further as a viable composite fading/shadowing distribution to model wireless channels and evaluate their performance through outage probability.

References

[1] S. Al-Ahmadi. The gamma-gamma signal fading model: A survey. *IEEE Antennas and Propagation Magazine*, 56(5):245–260, 2014.

[2] D. Aleksić, Z. Stefanović, M.and Popović, D. Radenković, and J. D. Ristić. On the K and K_G fading channels. *Serbian Journal of Electrical Engineering*, 6(1):187–201, 2009.

[3] L. J. Bain and M. Engelhardt. *Introduction to Probability and Mathematical Statistics*. Brooks/Cole, 1987.

[4] W. K. Chen. *The Electrical Engineering Handbook*. Academic Press, 2004.

[5] R. Haggarty. *Fundamentals of Mathematical Analysis*. Addison-Wesley New York, 2nd edition, 1993.

[6] M. J. Ho and G. L. Stüber. Co-channel interference of microcellular systems on shadowed Nakagami fading channels. In *Vehicular Technology Conference, 1993., 43rd IEEE*, pages 568–571. IEEE, 1993.

[7] N. L. Johnson and S. Kotz. *Distributions in Statistics: Continuous Univariate Distributions*. Houghton Mifflin, 1995.

[8] P. Karadimas and S. A. Kotsopoulos. The Weibull-lognormal fading channel: analysis, simulation, and validation. *IEEE Transactions on Vehicular Technology*, 58(7):3808–3813, 2009.

[9] P. J. Mostert, J. J. J. Roux, and A. Bekker. *A Bayesian method to analyse cancer lifetimes using Rayleigh models*. PhD thesis, Department of Statistics, University of Pretoria, November 1999.

[10] E.W. Ng and M. Geller. A table of integrals of the error functions. *Journal of Research of the National Bureau of Standards - B. Mathematical Sciences*, 73(1):1–20, March 1969.

[11] W. Roh and A. Paulraj. Outage performance of the distributed antenna systems in a composite fading channel. In *Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th*, volume 3, pages 1520–1524. IEEE, 2002.

[12] H. Samimi. Performance analysis of lognormally shadowed generalized gamma fading channels. *International Journal of Communication Systems*, 24(1):14–26, 2011.

[13] M. K. Simon and M. S. Alouini. *Digital Communication Over Fading Channels*, volume 95. John Wiley & Sons, 2005.

[14] P. C. Sofotasios and S. Freear. On the κ-μ/gamma composite distribution: A generalized multipath/shadowing fading model. In *Microwave & Optoelectronics Conference (IMOC), 2011 SBMO/IEEE MTT-S International*, pages 390–394. IEEE, 2011.

[15] H. Suzuki. A statistical model for urban radio propogation. *Communications, IEEE Transactions on*, 25(7):673–680, 1977.

[16] G. L. Turin, F. D. Clapp, T. L. Johnston, S. B. Fine, and D. Lavry. A statistical model of urban multipath propagation. *Vehicular Technology, IEEE Transactions on*, 21(1):1–9, 1972.

[17] N. Wang, X. Song, and J. Cheng. Generalized method of moments estimation of the Nakagami-m fading parameter. *Wireless Communications, IEEE Transactions on*, 11(9):3316–3325, 2012.

[18] Q. T. Zhang. Outage probability in cellular mobile radio due to Nakagami signal and interferers with arbitrary parameters. *IEEE Transactions on Vehicular Technology*, 45(2):364–372, 1996.

[19] R. Zhang, J. B. Wei, D. G. Michelson, and V. Leung. Outage probability of MRC diversity over correlated shadowed fading channels. *Wireless Communications Letters, IEEE*, 1(5):516–519, 2012.

[20] R. Zhang, J. B. Wei, J. B. Wei, and G. S. Li. Approximating bivariate Suzuki distribution with bivariate lognormal distribution. *IEICE Communications Express*, 2(11):470–477, 2013.

Appendix

Formulas

Error function [10]

$$\operatorname{erf}(y) = \frac{2}{\sqrt{\pi}} \int_0^y \exp\left(-u^2\right) du$$

Taylor series for exponential function [5]

$$\begin{aligned} \exp(tx) &= 1 + tx + \frac{(tx)^2}{2!} + \frac{(tx)^3}{3!} + \cdots \\ &= \sum_{i=0}^{\infty} \frac{(tx)^i}{i!} \end{aligned}$$

SAS Code

This appendix gives the code for graphing the PDFs and CDFs of the *compound-Weibull* and *compound-Weibull/lognormal* distributions since these have the largest number of parameters out of all the fading and composite distributions, respectively. This code can be adapted to all the distributions in this study. The code is for varying values of λ and μ, respectively, and can be adapted to all parameters.

Compound-Weibull distribution

```
*Varying values of lambda;

proc iml;

start f(x, alpha, k, lambda); *Define function;
v = alpha*k*lambda**k*x**(alpha-1)*(lambda+x**alpha)**(-k-1); *Fading distribution PDF;
return(v);
finish;

alpha = 3; *Set parameter values;
k = 2;
lambda_v = {0.5,1,2,5,10}; *Varying parameter values;
do i = 1 to nrow(lambda_v);
lambda = lambda_v[i];
do x = 0 to 5 by 0.01;
z = f(x, alpha, k, lambda);
z_vec = z_vec//z; *Creates PDF values;
x_vec = x_vec//x; *Vector of X values;
end;
cdf = cusum(z_vec)/max(cusum(z_vec)); *Creates CDF values;
density = x_vec||z_vec||cdf; *Creates matrix with X, PDF and CDF values;
density1 = density1||density; *Creates matrix with X, PDF and CDF values for all lambda;
free density;
free z_vec;
free x_vec;
free cdf;
end;

create plot1 from density1[colname = {'x1' 'fx1' 'CDF1' 'x2' 'fx2' 'CDF2'
'x3' 'fx3' 'CDF3' 'x4' 'fx4' 'CDF4' 'x5' 'fx5' 'CDF5'}];
append from density1;
close;
quit;

goptions reset = all;
ods escapechar="~";
proc sgplot data = plot1; *Plot of PDF;
series x=x1 y=fx1 / lineattrs=(thickness=2) legendlabel="0.5";
series x=x2 y=fx2 / lineattrs=(thickness=2) legendlabel="1";
series x=x3 y=fx3 / lineattrs=(thickness=2) legendlabel="2";
series x=x4 y=fx4 / lineattrs=(thickness=2) legendlabel="5";
series x=x5 y=fx5 / lineattrs=(thickness=2) legendlabel="10";
xaxis min=0 max=5 label="x";
yaxis min=0 max=2 label="Probability density function, f(x)";
keylegend /
```

```
across=1 border location=inside position=topright title="~{unicode lambda}";
run;
proc sgplot data = plot1; *Plot of CDF;
series x=x1 y=cdf1 / lineattrs=(thickness=2) legendlabel="0.5";
series x=x2 y=cdf2 / lineattrs=(thickness=2) legendlabel="1";
series x=x3 y=cdf3 / lineattrs=(thickness=2) legendlabel="2";
series x=x4 y=cdf4 / lineattrs=(thickness=2) legendlabel="5";
series x=x5 y=cdf5 / lineattrs=(thickness=2) legendlabel="10";
xaxis min=0 max=5 label="x";
yaxis min=0 max=1 label="Cumulative distribution function, F(x)";
keylegend / across=1 border location=inside position=bottomright title="~{unicode lambda}";
run;
```

Compound-Weibull/lognormal distribution

```
*Varying values of mu;

proc iml;
*Define function with global variable since;
start f(lambda) global(x_, alpha_, k_, mu_, sigma_);
p = constant("Pi");
x = x_;
alpha = alpha_;
k = k_;
mu = mu_;
sigma = sigma_;
v = alpha*k*lambda**k*x**(alpha-1)*(lambda+x**alpha)**(-k-1)/(sqrt(2*p)*sigma*lambda)
*exp(-0.5*((log(lambda)-mu)/sigma)**2); *Composite PDF without integral;
return (v);
finish;

alpha_ = 3; *Set parameter values;
k_ = 2;
sigma_ = 1;
mu = {-5,-1,0,1,5}; *Varying parameter values;
do i = 1 to nrow(mu);
mu_ = mu[i];
a = {0  .P}; *Set integral bounds;
z_vec = J(1,1,.);
x_vec = J(1,1,0);
do x_ = 0.001 to 5 by 0.01;
call quad(z, "f", a); *Integrate over lambda to obtain the composite PDF;
z_vec = z_vec//z; *Creates PDF values;
x_vec = x_vec//x_; *Vector of X values;
end;
cdf = cusum(z_vec)/max(cusum(z_vec)); *Creates CDF values;
density = x_vec||z_vec||cdf; *Creates matrix with X, PDF and CDF values;
density1 = density1||density; *Creates matrix with X, PDF and CDF values for all mu;
free density;
free z_vec;
free x_vec;
free cdf;
end;
```

```
create plot1 from density1[colname = {'x1' 'fx1' 'CDF1' 'x2' 'fx2' 'CDF2'
'x3' 'fx3' 'CDF3' 'x4' 'fx4' 'CDF4' 'x5' 'fx5' 'CDF5'}];
append from density1;
close;
quit;

goptions reset = all;
ods escapechar='~';
proc sgplot data = plot1; *Plot of PDF;
series x=x1 y=fx1 / lineattrs=(thickness=2) legendlabel="-5";
series x=x2 y=fx2 / lineattrs=(thickness=2) legendlabel="-1";
series x=x3 y=fx3 / lineattrs=(thickness=2) legendlabel="0";
series x=x4 y=fx4 / lineattrs=(thickness=2) legendlabel="1";
series x=x5 y=fx5 / lineattrs=(thickness=2) legendlabel="5";
xaxis min=0 max=5 label="x";
yaxis min=0 max=2 label="Probability density function, f(x)";
keylegend /
across=1 border location=inside position=topright title="~{unicode mu}";
run;
proc sgplot data = plot1; *Plot of CDF;
series x=x1 y=cdf1 / lineattrs=(thickness=2) legendlabel="-5";
series x=x2 y=cdf2 / lineattrs=(thickness=2) legendlabel="-1";
series x=x3 y=cdf3 / lineattrs=(thickness=2) legendlabel="0";
series x=x4 y=cdf4 / lineattrs=(thickness=2) legendlabel="1";
series x=x5 y=cdf5 / lineattrs=(thickness=2) legendlabel="5";
xaxis min=0 max=5 label="x";
yaxis min=0 max=1 label="Cumulative distribution function, F(x)";
keylegend / across=1 border location=inside position=bottomright title="~{unicode mu}";
run;
```