Approximate Bayesian computation for a spatial susceptible-exposed-infectious-removed model

by

Arminn Potgieter

Submitted in partial fulfillment of the requirements for the degree

Magister Scientiae

In the Department of Statistics

In the Faculty of Natural and Agricultural Sciences

University of Pretoria

October 2021

Declaration

I, *Arminn Potgieter,* declare that this mini-dissertation (100 credits), which I hereby submit for the degree Magister Scientiae in Mathematical Statistics at the University of Pretoria, is my own work and has not previously been submitted by me for a degree at this or any other tertiary institution.

Signature: A.Potgieter

Date: 13-01-2022

Summary

In this mini-dissertation we utilize population mobility data and COVID-19 case data in a variety of formats, from a variety of sources, in order to formulate a model for the spatial spread of COVID-19. The study region for this mini-dissertation is the Western Cape province of South Africa. Appropriate spatial structures are formulated using both standard and novel approaches, and the effect of these different conceptualisations of spatial association are illustrated, compared and discussed.

The spatial spread of COVID-19 is modelled using a susceptible-exposed-infectious-removed (SEIR) model that describes the progression of the disease. The model is stochastic in nature in order to incorporate the inherent uncertainty present in pandemic parameters. The stochastic nature of the model allows for greater inferential capabilities than deterministic models. Pandemic characteristics such as the spatial autocorrelation of COVID-19 cases and the reproductive number of the disease are determined and discussed.

Model fitting and inference are achieved through the use of approximate Bayesian computation (ABC) techniques for likelihood-free inference. This computational framework extends naturally to stochastic pandemic models, since the potentially complex disease system results in computationally infeasible likelihood expressions. The use of artificial neural networks for the purpose of improving the computational efficiency of this computational framework is evaluated and discussed.

Acknowledgements

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

Thank you to STATOMET for additional funding received in the form of the Internal Statistical Consultation Service (ISCS) HUB internship program of the Department of Statistics, University of Pretoria.

The mobile network data used in this mini-dissertation is not directly available without approval and so cannot be shared directly. We express gratitude to Zaid Kimmie for serving as custodian and overseer of this data[1].

I would like to thank my supervisor, Professor Inger Fabris-Rotelli, for her great deal of support and assistance in the conceptualization, execution and writing of this mini-dissertation. Thank you for providing me with opportunities to obtain more experience in the field of academic research. Your compassion and work ethic is an inspiration to each and every one of those fortunate enough to have you as their supervisor.

Finally, thank you to my colleagues from the COVID-19 collaborative research team for sharing their ideas, opinions and insights which contributed to the conceptualization and writing of this mini-dissertation. A special thanks to Ms Renate Thiede for invaluable assistance with the writing of this mini-dissertation as well as the papers we have collaborated on.

[1] For further enquiries: zaid.kimmie@gmail.com

Contents

List of Figures

List of Tables

List of Abbreviations

ABC	Approximate Bayesian computation
ANN	Artificial neural network
API	Application Programming Interface
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CFAA	Computer Fraud and Abuse Act
COVID-19	Coronavirus disease of 2019
CNN	Convolutional neural network
CSS	Cascading Style Sheets
DNN	Deep neural network
ENN	Exchangeable neural network
HTML	Hypertext Markup Language
MCMC ABC	Markov chain Monte Carlo approximate Bayesian computation
NICD	National institute for communicable diseases
PEN	Partially exchangeable network
SARS	Severe Acute Respiratory Syndrome
SEIR	Susceptible-exposed-infectious-removed
SMC ABC	Sequential Monte Carlo approximate Bayesian computation
URL	Uniform Resource Locator
WWW	World Wide Web
XML	Extensible Markup Language

Chapter 1

Introduction

The onset of the COVID-19 pandemic at the start of 2020 had a profound effect on every country in the world. COVID-19, which refers to the Coronavirus disease of 2019, is an infectious respiratory disease[1] caused by the SARS-CoV-2 virus. Most individuals infected with the disease develop mild flu-like symptoms, however there have been many cases where infected individuals become seriously ill and die unless they receive specialized treatment. In some cases, those that recover from the disease exhibit symptoms months after recovery. The disease is transmitted through small liquid particles from an infectious individual's nose and/or mouth[2]. Other potential means of transmission include physical contact with surfaces contaminated with the virus, however the probability of this occuring is considered to be low[3].

The disease was first identified in the city of Wuhan within the Hubei province of China on 31 December 2019[4]. This followed a string of cases initially believed to be viral pneumonia of unknown cause. On 9 January 2020 it was confirmed that these cases were not pneumonia but rather caused by a novel coronavirus. The first death due to the disease caused by this virus was documented shortly thereafter. Over the course of the next two months, cases of the disease were identified across the globe. The first confirmed case in South Africa was confirmed on 5 March 2020[5]. With the spread of the disease proving difficult to control, governments around the globe instilled non-pharmaceutical interventions. These interventions aim to limit the spread of the disease by limiting the mobility and interaction of the population. Due to their restrictive nature, these interventions are often referred to as "lockdowns".

[1] WHO COVID-19 characteristics: https://www.who.int/health-topics/coronavirus#tab=tab_1 (Accessed October 2021)
[2] WHO COVID-19 transmission: https://1e.to/f2NA6v (Accessed October 2021)
[3] Centers for Disease Control and Prevention: https://1e.to/HI7YyG (Accessed October 2021)
[4] WHO COVID-19 timeline: https://www.who.int/news/item/29-06-2020-covidtimeline (Accessed October 2021)
[5] NICD report on first case in South Africa: https://1e.to/eJlHBd (Accessed October 2021)

The South African government implemented one of the most stringent lockdowns in the world starting on the 27th of March 2020. This lockdown strategy revolves around five different "levels" of lockdown, with level 1 and 5 being the least and most stringent levels respectively. The various levels as well as the dates for which they were active (during 2020) are given in Table 1.1. Note that for this mini-dissertation we only consider the lockdown until the end of September 2020 due to data availability only extending over this period[6]. As of 1 October 2021, South Africa is operating under an adjusted version of the level 1 lockdown regulations[7].

As is the case with almost every single country in the world, South Africa has seen a rather large amount of research published studying COVID-19 in a very short time. The primary focus of the majority of these studies is on the economic impact of the imposed lockdown regulations [9, 10, 123, 89, 115] while others discuss the impact of COVID-19 on education in the country in particular [85, 69]. While studies discussing the economic and social impact of COVID-19 assist in providing context for the pandemic in South Africa, they lack any quantitative means of predicting the future impact and spread of the virus. In order to meet this demand some authors have proposed mathematical models for modelling the spread of the disease across the country [49, 141, 87, 79, 118, 59, 90]. By reviewing the available literature we will aim to identify regions where additional contributions can be made to the mathematical modelling of COVID-19 within the South African setting.

The mathematical modelling and prediction of infectious diseases represents an important aspect of epidemiology. Epidemiological models assist researchers and policymakers in understanding and planning for the presence, spread and negative effects of disease outbreaks. The development of a mathematical framework for modelling such diseases originated in the first half of the 20th century (as early as 1915) due to the work of researchers such as Ross [108, 109], McKendrick [86] and Kermack [62], Bartlett [12] and Kendall [61]. The result of these early ventures into disease modelling was a family of compartmental disease models that would proceed to be adjusted and modified to see a large degree of popularity over the course of the next century. These compartmental models, known as susceptible-exposed-infectious-removed (SEIR) models, operate by segregating the study population into various compartments based on the progression of the disease being studied [108, 109, 86, 62, 12].

The original formulation of SEIR models as proposed by Ross [108, 109] as well as Kermack and McKendrick [86, 62] were deterministic in nature, containing no stochastic element or allowance for uncertainty. Subsequent adaptations of SEIR models would rectify this to include stochastic components in order to

[6] For further details regarding lockdown levels: https://1e.to/9xbQIo

[7] Further details on South Africa's adjusted alert level 1: https://1e.to/Rb7ARt

Table 1.1: South African lockdown levels and dates [100].

Level	Date	Restrictions
Business as usual	1 March 2020 - 26 March 2020	No restrictions.
Level 5	27 March 2020 - 30 April 2020	All individuals confined to place of residence aside from essential services. No inter-provincial movement, except for transportation of goods and exceptional circumstances e.g. funerals. Public and private transport restricted to certain times of the day, with limitations on vehicle capacity.
Level 4	1 May 2020 - 31 May 2020	More sectors permitted to operate with restrictions, including mining, and partial e-Commerce. Public spaces and the tourism sector remain closed and gatherings prohibited. All confined to place of residence from 8pm-5am. No local or inter-provincial movement of people, except for permitted reasons. All borders remain closed except for designated ports of entry for restricted home affairs operations and for the transportation of fuel, cargo and goods. Public and private transport may operate at all times of the day, with limitations on vehicle capacity.
Level 3	1 June 2020 - 17 August 2020	More sectors permitted to operate including take away restaurants, e-commerce and delivery services and global business services. Public spaces and tourism opened. Gatherings and sporting activities permitted subject to restrictions. All confined to place of residence from 11pm-4am. No inter-provincial movement of people, except for transportation of goods, exceptional circumstances and other permitted reasons. Public and private transport may operate at all times of the day, with limitations on vehicle capacity.
Level 2	18 August 2020 - 20 September 2020	More sectors permitted to operate including all retail, construction, cleaning and manufacturing as well as government services. All confined to place of residence from 11pm-4am. Domestic travel permitted.
Level 1	21 September 2020 - 28 December 2020	Most restrictions lifted. All sectors permitted to operate. All confined to place of residence from 12pm-4am. Inter-provincial travel permitted with restrictions on international travel.

account for the large degree of uncertainty and randomness that can often be associated with the spread of infectious diseases [86, 12, 61]. Both deterministic as well as stochastic SEIR models have been utilized to study the spread of a variety of infectious diseases across the world including Ebola [72, 93], influenza [54, 32, 34] as well as prior outbreaks of severe acute respiratory syndrome (SARS) [33] during the 2000's. Following the outbreak of COVID-19 in 2020 there has been a significant increase in use for such models, having been used to model the spread of COVID-19 in various countries such as Belgium [1], Chile [26], China [74], Japan [67] and Italy [97]. The goals of such analysis range from early phase pandemic estimation [1, 97], peak prediction [67], ICU bed capacity prediction [26] and evaluation of containment strategies [74]. In South Africa these models have been utilized to determine the impact of non-pharmaceutical interventions [49, 141, 118], to evaluate vaccine roll-out strategies [87], evaluate the viability of alternative medical responses [59, 90] and predicting future COVID-19 cases [49, 141, 87, 79]. To the best of our knowledge there is currently only one SEIR model developed in South Africa that is stochastic in nature (see [80]). The potential contribution of a stochastic SEIR model within the South African context is thus evident.

Infectious diseases such as COVID-19 are transmitted primarily through close contact with infectious individuals. The benefit to be gained from the inclusion of spatial elements that approximate population mobility patterns in the modelling process is thus immediately apparent. In order to facilitate the inclusion of spatial components, numerous authors have proposed and utilized spatial variations of SEIR models. These spatial models allow the spread of a disease within a particular region to be affected by the state of the epidemic in neighbouring regions [24]. Spatial SEIR models saw a noteworthy increase in popularity throughout the 2010's, being used to model the spatial spread of infectious diseases such as cholera [44, 18], dengue [36, 138], Ebola [98, 24, 25], foot and mouth disease [122, 37, 75, 31] and malaria [111, 137]. A major simplifying assumption that is often used in these models is that both the latent and infectious periods of the disease being studied are distributed exponentially (see e.g. [34, 32]). While this assumption simplifies analysis it also implies that the probability of becoming infectious or recovering from an infection is the same for each point in time due to the memory-less property of the exponential distribution [24]. The path-specific SEIR model removes this restriction and allows for these times to be distributed according to any general distribution [98, 24, 25]. This allows for greater flexibility and realistic modelling capabilities compared to previous spatial SEIR models. Spatial models have also been used to model the spread of COVID-19 in countries such as Italy [134], Argentina [125], China [56], England [113] and Spain [8]. The clear advantage of utilizing spatial SEIR models lies in allowing policymakers to identify regions more adversely affected by the spread of a disease in order to aid in formulating an appropriate allocation of resources. Reviewing the current literature it is clear that there is a significant deficit of spatial SEIR models being applied to study COVID-19 in South Africa. Most previously applied SEIR models have studied the disease at a country [49, 141, 87, 79] or provincial [118, 59, 90] level and thus disregard spatial components [49, 141, 87, 79]. To the best of our knowledge there is currently only one paper ([42]) that strives to address this need. There is thus significant novelty in the application of a spatial SEIR model within a South African setting for the purpose of modelling COVID-19.

The inclusion of spatial components within epidemiological models represents an opportunity for researchers to study the spatial spread of a disease within and between affected regions. However the inclusion of a spatial component also represents a non-trivial problem, that being the specification of some structure that accurately describes the spatial association between the study regions. Spatial associations are typically included through the use of a spatial weight matrix, the specification of which has been an open problem in the field of Econometrics and general spatial modelling for some time [6, 13, 40]. Previous attempts at formulating spatial weight matrices were based on study region properties such as contiguity, distance and geostatistical data [3, 29]. These methods can often be very simplistic and rarely offer a realistic representation of the true spatial association within the study region [6, 13, 40]. Furthermore the chosen forms in spatial epidemiology models is often motivated by a need for simplicity rather than accuracy and thus the formulation of a proper spatial component in spatial epidemiology still represents an open problem. Within the field of epidemiological modelling mobile network data has seen an increase in use as a proxy for human mobility, being used for modelling diseases such as cholera [44, 18], dengue [36, 138] and malaria [111, 137]. Most recently mobile network data has been gathered in numerous countries across the world to aid in the understanding of human mobility for the purpose of informing non-pharmaceutical interventions against COVID-19 [41, 132, 91, 95]. There are several concerns with regards to the use of mobile network data for this purpose, including data availability, computational cost and user privacy [91, 53]. Currently there is no existing literature that utilizes mobile network data for the modelling of COVID-19 within the South African context, although some preliminary attempts have been presented (see [42, 100].)

The inclusion of spatial components as well as general distributions for transition times can lead to the number of unknown parameter(s) within the model becoming rather large. This can lead to cases where the full likelihood function for the data becomes either intractable or too computationally expensive to evaluate [130, 124, 83, 25]. In 1984 Rubin proposed that statisticians utilizing Bayesian inference should not be limited to cases where the likelihood function is wholly tractable [110]. This lead to the conceptualization of an approximate Bayesian technique that is used to simulate observations from a desired posterior distribution without the need to evaluate the full likelihood function. This method, termed approximate Bayesian computation, can be utilized to fit complex epidemiology models such as the spatial SEIR models where factors such as computational cost and the curse of dimensionality are a noteworthy barrier to use [130, 124, 83, 25]. The basic idea of the algorithm is to propose a parameter value from the prior, which is then used to simulate an artificial dataset [101, 130, 15]. The proposed parameter value is then only accepted if the simulated dataset is sufficiently similar to the observed dataset [101, 130, 15]. Thus we do not need to evaluate the full likelihood function. Over the following decades numerous variations of ABC were proposed, with each aiming to compensate for an area wherein previous attempts were lacking. The first formal rejection algorithm was proposed by Tavaré in 1997 [127]. This sparked a renewed interest in ABC and prompted researchers such as Fu and Li [46], Weiss and von Haesler [136] and Pritchard et al. [101] to propose their own variations on the initial ABC algorithm over the proceeding years with the current definition of the rejection ABC algorithm as the result.

In order to improve the accuracy and efficiency of the ABC rejection algorithm Beaumont et al. [17] as well as Blum and Francois [21] proposed adjustments using linear regression and feed-forward neural networks (FFNN's) in 2002 and 2010 respectively. Chief among the improvements made upon the rejection algorithm however are two new algorithms that incorporate popular Bayesian inference techniques into an approximate calculation framework. In 2003 Marjoram et al. [77] proposed the Markov chain Monte Carlo ABC (MCMC ABC) algorithm, which utilizes elements of Metropolis-Hastings sampling to improve upon the efficiency of the rejection algorithm [77, 120, 130]. In 2007 Sisson et al. [120] proposed the sequential Monte Carlo ABC (SMC ABC) algorithm which improves upon several of the shortcomings of both previous versions of ABC [120, 130, 76, 83, 15]. Approximate Bayesian inference was first proposed for use within the field of genetics, where high-dimensionality can often result in the likelihood function being intractable or computationally infeasible [60, 21, 14, 76, 107]. However such techniques have seen increasing use within the field of epidemiology, being used for modelling diseases such as tuberculosis, HIV, Ebola, influenza and smallpox over the last 15 years [126, 117, 130, 81, 104, 82, 68, 83, 25]. To the best of our knowledge, ABC techniques have not been utilized for inference on COVID-19 at the time of writing.

The ABC family of algorithms base the decision of whether to retain or discard candidate parameter values on whether the data that is simulated using these candidates is sufficiently similar to the observed data [101, 130, 15]. In order to increase the acceptance rate and decrease the computational cost of these algorithms it has become standard to compare selected summary statistics calculated for both the simulated and observed data rather than to compare the entire datasets [60, 130, 81, 124, 15]. Sufficient summary statistics would be the ideal choice for this purpose however they are most often unavailable [107, 124, 15]. The choice of summary statistic is more often than not restricted to a set of expertly crafted candidate summary statistics for which the performance is evaluated and compared [60]. This however introduces additional bias and uncertainty and thus there has been a growing interest in the development of methods for automatically constructing summary statistics [126, 83, 15].

Recent studies have proposed the use of artificial neural networks (ANN) for summary statistic construction [43, 58, 28, 78, 2]. In 2012 Fearnhead and Prangle [43] proposed a semi-automatic method for summary statistic selection involving non-linear regression techniques. A major contribution of their work was showing that the mean of the true posterior can serve as a sufficient summary statistic [43, 58]. Building upon this idea and motivated by a desire for greater representative power, in 2017 Jiang et al. [58] proposed using a deep neural network (DNN) to automatically learn summary statistics. In their application they trained a DNN to estimate the mean of the true posterior that would serve as the summary statistic in an ABC algorithm [58]. In 2018 and 2019 two more applications of ANN's were utilized in the field of genetics by Chan et al. [28] and Wiqvist et al. [78] respectively. These applications utilized exchangeable and partially exchangeable networks (ENN and PEN respectively) to incorporate the exchangeability of population data into the estimation process [28, 78]. Finally, in 2020 Kesson et al. [2] compared the performance of a convolution neural network (CNN), an ENN and a PEN for the purpose of summary statistic selection. They found that while all three models perform generally well, the CNN outperformed the other models in some scenarios [2]. All of the above-mentioned applications were focused either on the field of genetics for which ABC was initially formulated [28, 2] or utilized simplistic models such as moving average processes to maintain simplicity [58, 2]. At the time of writing, no attempt has been made to incorporate neural networks into an approximate Bayesian inference for the purpose of compartmental epidemiological modelling.

By considering the above-mentioned history and literature we have determined several opportunities to contribute to the existing literature of COVID-19 in a South African setting that will be addressed in this mini-dissertation. All of the currently available literature set within South Africa that studies the spread of COVID-19 [49, 141, 118, 87, 59, 90, 49, 141, 87, 79] does so without the explicit inclusion of a spatial component. The model(s) utilized in this mini-dissertation will be constructed at a higher spatial resolution than country or provincial level and will explicitly make allowance for the spatial association between the different sub-regions in South Africa. Furthermore, all models developed thus far in South Africa have been deterministic in nature which discounts the inherent uncertainty present in disease modelling [49, 141, 87, 79]. The model(s) utilized in this mini-dissertation will be stochastic in nature, allowing for greater uncertainty to be included in the estimation process. The use of mobile network data as a proxy for human mobility and the general uncertainty present in the construction of spatial association structures within the South African context is also underdeveloped. We will utilize two types of population mobility data in order to derive comparisons between data from different sources and the implications of using different constructions for spatial association structures. Finally, by employing deep learning to facilitate the use of an approximate Bayesian inference technique for model fitting we will add to the currently limited literature on this application of neural networks. We will do all of this against the backdrop of the early phases of the COVID-19 pandemic in the Western Cape, South Africa. The primary objectives of this mini-dissertation are thus as follows:

- Study the spatial association structure in South Africa using two sources of population mobility data and develop spatial weight matrices that express these underlying spatial structures.
- Employ a stochastic spatial SEIR model to model the spatial spread of COVID-19 in South Africa utilizing various resources for COVID-19 data.
- Utilize and compare the performance of different algorithms for approximate Bayesian computation to fit the selected model(s).
- Study the effect of using artificial neural networks to aid in the fitting of the selected model(s) during the approximate Bayesian step by condensing pandemic data down to a lower dimensionality representation.

The rest of this mini-dissertation is structured as follows. In Chapter 2 we discuss the various data that were utilized for analysis in this mini-dissertation and the means through which certain data were obtained. In Chapter 3 we discuss the development of spatial structures that approximate the spatial autocorrelation of COVID-19 present within South Africa. Chapter 4 is dedicated to the discussion and development of the compartmental epidemiological models used for modelling the spread of COVID-19. In Chapter 5 we discuss approximate Bayesian computation as a method for fitting and evaluating stochastic compartmental models. Chapter 6 then delves into the background and properties of artificial neural networks and their potential use in conjunction with approximate Bayesian computation (ABC) techniques. Chapter 7 discusses the application of all the statistical principles discussed in this mini-dissertation and Chapter 8 presents the results. Chapter 9 then offers a discussion of these results followed by a conclusion provided in Chapter 10.

Chapter 2

Data

The data utilized for this mini-dissertation were collected using a variety of sources and means. Given the complex nature of the phenomena under study it proved impossible to identify a single source of data with sufficient information. Additionally, it was noted that different data sources each exhibited their own merits (such as high spatial resolution, good data range, availability etc.) as well as shortcomings (such as low spatial resolution, high computational cost, lack of availability etc.) and thus it was decided to use multiple data sources in order to discuss and compare the insight that can be gathered from each source. In this chapter we outline the data sources used for this mini-dissertation and offer a brief description of the data.

2.1 Western Cape COVID-19 data

Since the initial outbreak of the COVID-19 pandemic, governments all over the world have been heavily criticized for the high level of secrecy exhibited with regards to data pertaining to the pandemic[1]. Case numbers that are made available to the general public are typically of very low spatial resolution (for example, in South Africa the case numbers are most often only published at a provincial level). This makes modelling the pandemic very difficult for researchers who are not affiliated with government institutions with direct access. Furthermore, most data made available through other sources are also highly aggregated and in many cases inaccurate[2]. This is almost certainly the result of a lack of the infrastructure

[1] For example: Case numbers (https://1e.to/LSxp3C), government correspondence (https://1e.to/oYeTqO), vaccine roll-out plan (https://1e.to/U1OnIV) and government restrictions (https://1e.to/ibCzGV).

[2] Global.health: https://global.health/

and expertise required to capture and record case data in a timely and accurate manner[3].

In South Africa, the only party that has succeeded in consistently providing COVID-19 case number data to the public at a high spatial resolution and in a timely manner is the Western Cape government. Since the 28th of March 2020, the premier of the Western Cape, Alan Winde, has publicly disclosed the number of infections confirmed within each local municipality of this province through press releases[4]. A further initiative was undertaken to improve the ease with which these figures could be accessed and so the Western Cape government launched a COVID-19 dashboard[5], shown in Figure 2.1, which contained various forms of data relating to COVID-19 in the Western Cape from as early as the 10th of March 2020. It is indicated on the wesbite where the dashboard is presented that the dashboard is estimated to have a data completeness and data accuracy of 93% and 84% respectively.

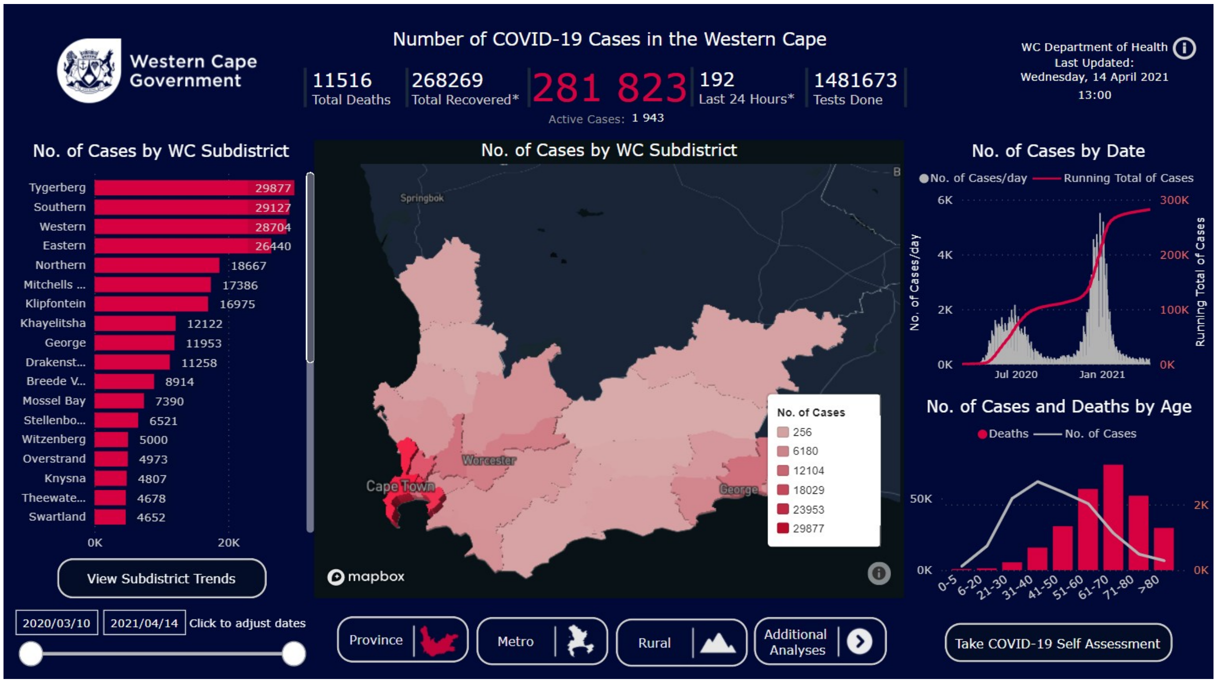


Figure 2.1: Western Cape COVID-19 dashboard (screenshot taken 15 April 2021)

As can be seen in Figure 2.1, at the time the screenshot was taken, the information contained on the dashboard included the total number of cases for each local municipality as well as the recoveries, deaths and tests conducted in the entire province. The total number of cases and deaths by age group were also available. Thanks to the inclusion of date sliders all this information was available over time as well, rendering the dashboard an extremely valuable tool for research purposes. Initially it was planned to utilize this dashboard to obtain case data for the Western Cape province for the purpose of modelling

[3] Health-e news: https://1e.to/JGVS92

[4] Press release 4 July 2020: https://www.gov.za/speeches/update-coronavirus-premier-alan-winde-4-jul-2020-0000

[5] COVID-19 dashboard: https://coronavirus.westerncape.gov.za/covid-19-dashboard

the spread of COVID-19. Unfortunately the dashboard was altered at some point in time to no longer provide the same information. The currently available version of the dashboard at the time of writing is shown in Figure 2.2.

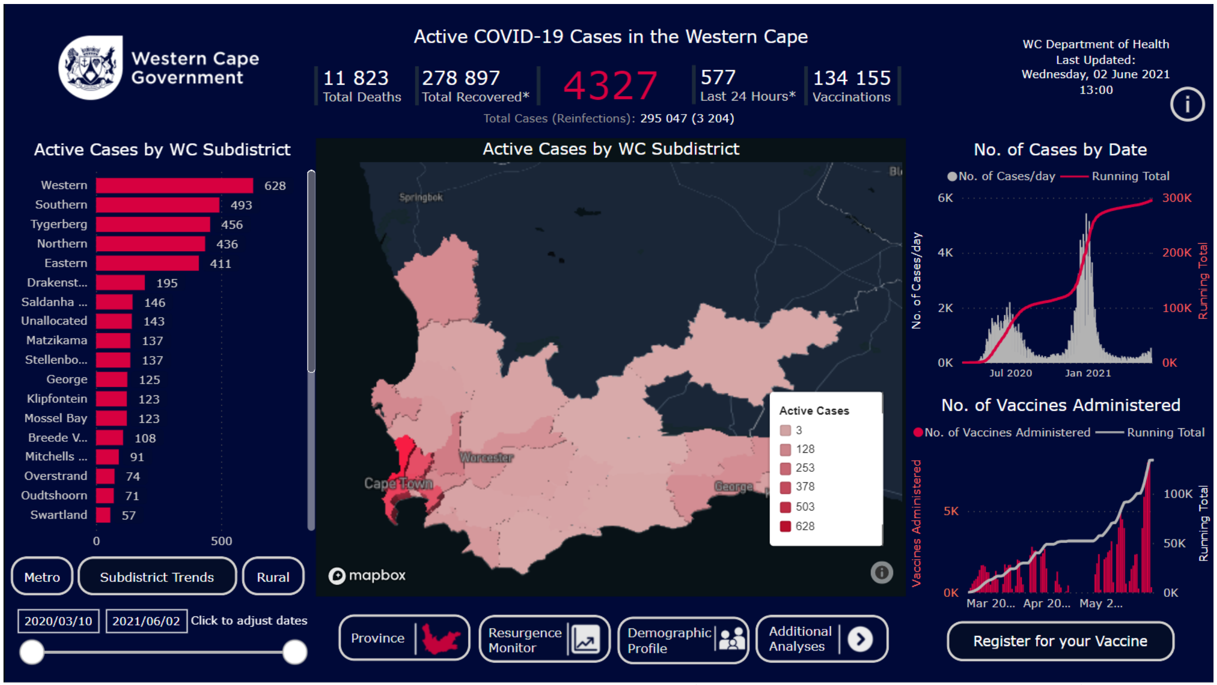


Figure 2.2: Western Cape COVID-19 dashboard (screenshot taken 17 August 2021)

Note how this new version does not contain the same highly sought-after data such as the total number of cases across numerous points in time but is instead restricted to displaying the number of *active* cases for each local municipality over time. Additionally, the dashboard's terms and conditions state the following: *"This website and its contents, including all mapping, data, and analysis are provided to the public strictly for public health and educational purposes. The Western Cape Department of Health hereby disclaims any and all representations and warranties with respect to the website, including accuracy, fitness for use, reliability, and non-infringement. Reliance on the website for medical guidance, research or use of the website in commerce is strictly prohibited."*. In addition to the dubiously implied distinction between "education" and "research", these terms and conditions forbid the use of the dashboard for research projects. It is unclear whether these terms and conditions were set at the creation of the dashboard or some later point in time. Either way, this prevents us from utilizing the dashboard directly for our intended purposes. In order to gain access to the data presented on the dashboard it will thus prove necessary to obtain the information from the original source: The daily press releases made by the Western Cape premier.

2.1.1 Web scraping

As mentioned previously, the premier of the Western Cape, Alan Winde, publishes daily press releases wherein the number of COVID-19 cases in the province are given at a local municipality level. In theory, it would be possible for a human user to peruse every press release over a desired time period and write down all numbers of interest. Aside from being very time-consuming, this process would almost certainly be unreliable due to human error. Aside from the previously discussed dashboard, there is also no repository available from which the case data itself can be downloaded in a usable format for further analysis. Most often, websites will provide an application programming interface (API) that allows users to download desired data, however when an API is not provided to download otherwise publicly available data (such as with these press releases) then it becomes necessary to use web scraping [23]. Web scraping allows us to automate the previously described process of browsing the internet and writing data down. Web scraping (sometimes called "web harvesting", "web data extraction", "web data mining" etc.) is a tool that allows researchers to automate the process of extracting and organizing data available on the internet [23, 65, 66, 140].

The history of web scraping as a means of collecting data on the internet is essentially the history of the internet itself. Shortly after the creation of the world wide web (WWW) in 1989, the first web robot, Wanderer, was created in 1993. The purpose of Wanderer was to scan and measure the size of the internet in order to create an index of web pages [116]. Interestingly, this actually predates the first search engine, W3Catalog (granted only by a few months) [116]. A web robot (or simply 'bot') is any program or software that utilizes scripts to automate the process of interacting with the WWW without human supervision [116]. A web scraper is thus just a type of web robot, one with the explicit goal of data extraction and organization. Fast forward to today and the largest search engine in the world, Google, has made a multi-billion dollar business out of crawling the web for information [23]. Web scraping has seen tremendous growth in many scientific disciplines such as computer science [66, 65], data science [66, 65, 23] and epidemiology [106] as well as industry fields such as journalism, property investment and the financial sector [65].

The typical workflow of a web scraper is as follows: website analysis, website crawling and data organization (see Figure 2.3). Web scraping requires at least a basic understanding of the way in which web pages are built [23, 65]. Important features of a web page for web scraping are hypertext markup language (HTML), cascading style sheets (CSS) and extensible markup language (XML). Most web pages are created and formatted using HTML, which is a markup language that allows web designers to specify how web pages are supposed to be structured [23, 140]. While HTML is used to form the basic structure of a web page, CSS is another programming language that allows web designers to exert more control over how a web page should look (properties such as colour, shading, fonts etc.) [23]. Finally, XML is yet another programming language of which HTML is merely an implementation. The reason for considering HTML and XML separately will be made clear shortly.

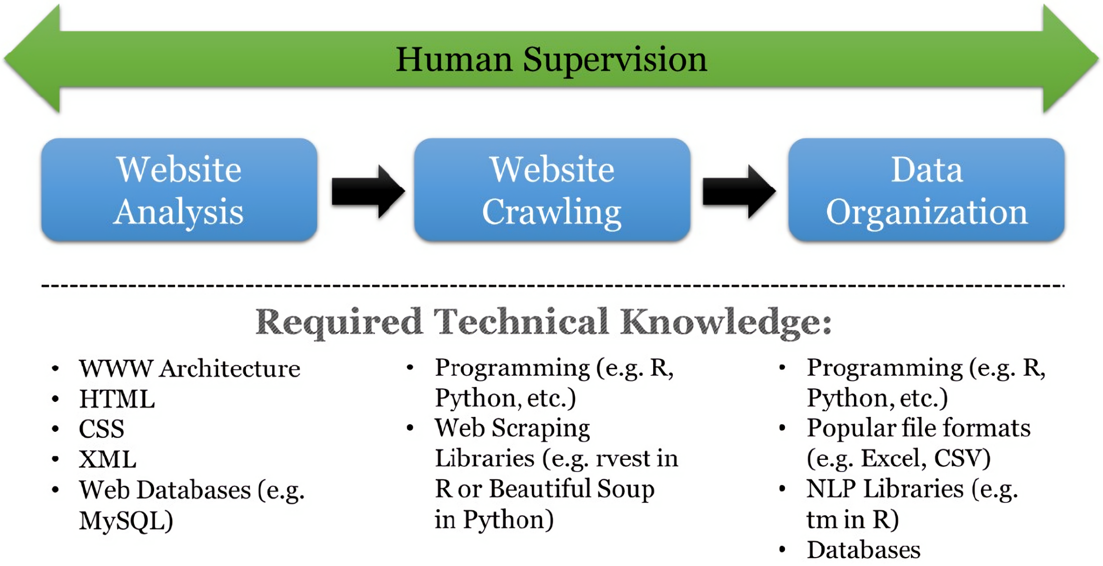
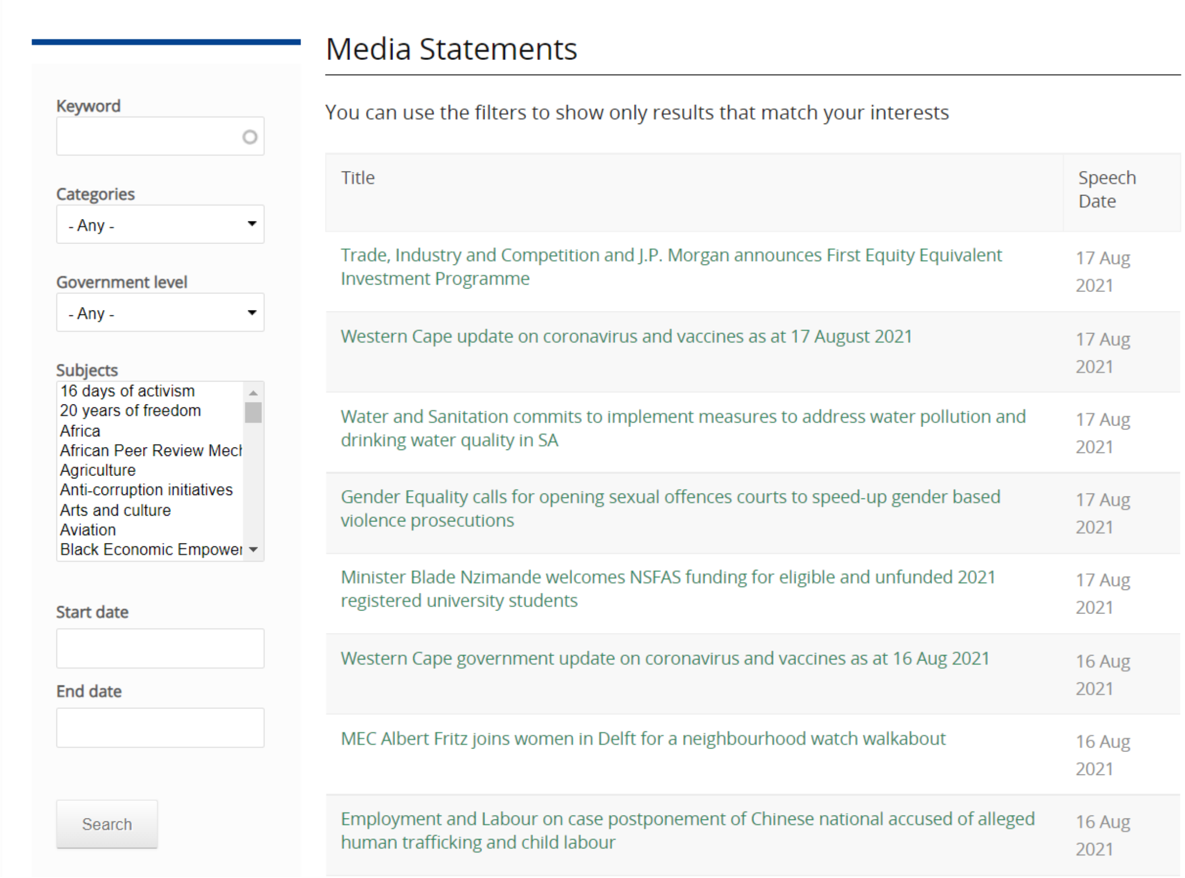

Human Supervision
Website Analysis
Website Crawling
Data Organization
Required Technical Knowledge:
• WWW Architecture
• HTML
• CSS
• XML
• Web Databases (e.g. MySQL)
• Programming (e.g. R, Python, etc.)
• Web Scraping Libraries (e.g. rvest in R or Beautiful Soup in Python)
• Programming (e.g. R, Python, etc.)
• Popular file formats (e.g. Excel, CSV)
• NLP Libraries (e.g. tm in R)
• Databases

Figure 2.3: Web scraping workflow [65]

After analysing the way in which a web page is structured we can proceed to write a script to extract the key elements from a page using a popular data science programming language such as `R` or `Python` [23, 65, 140, 66]. Web scraping in `R` is primarily achieved through the `rvest` package while in `Python` the two most popular packages are `BeautifulSoup` and `Selenium` [140, 23, 65, 66]. The popular data science package `pandas` also contains some useful functions for web scraping. In this mini-dissertation `Python` was used to web scrape the press releases through the use of the `Selenium` and `pandas` packages.

The first step in web scraping requires opening a chromium based web browser window using `Selenium` and then navigating to the desired web page using its uniform resource locator (URL). Dashboards like the one in Figure 2.1 are very convenient since they only require a single web page to be opened. However when scraping something like press releases there is the issue that the URL for the press releases can be very different, making it difficult or impossible to write a concise peace of code to iterate through the addresses without writing down a full list of URLs. In order to circumvent this issue we utilize the South African government media statements repository[6] shown in Figure 2.4. This repository allows us to enter a desired keyword for a media statement as well as a start and end date to consider.

[6] Media statement repository: https://www.gov.za/media-statements



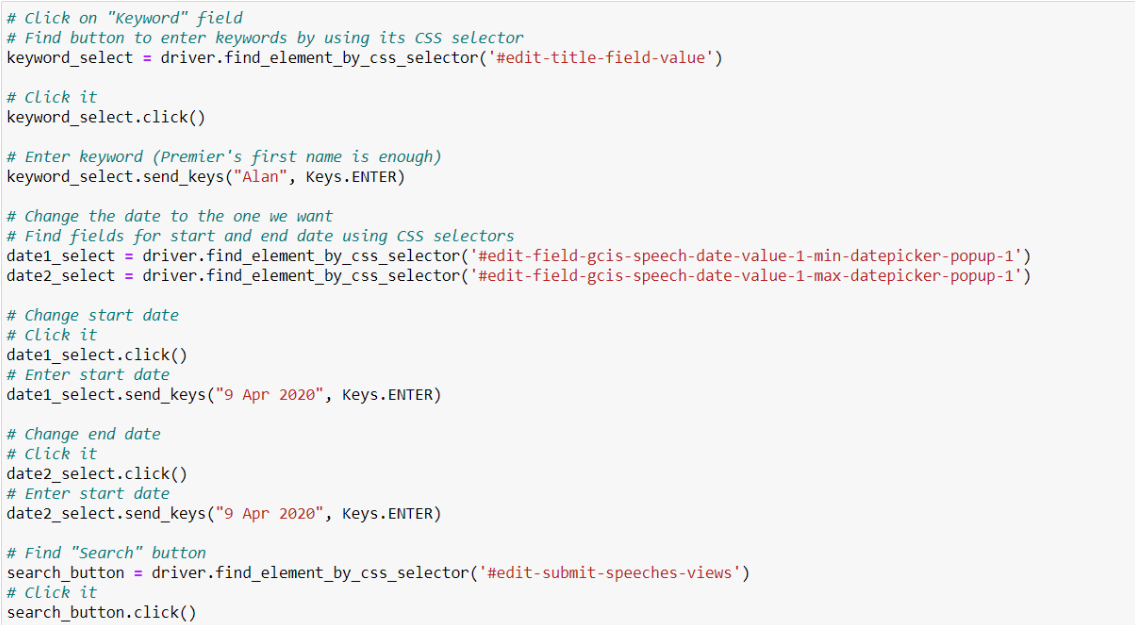
Media Statements

You can use the filters to show only results that match your interests

Title	Speech Date
Trade, Industry and Competition and J.P. Morgan announces First Equity Equivalent Investment Programme	17 Aug 2021
Western Cape update on coronavirus and vaccines as at 17 August 2021	17 Aug 2021
Water and Sanitation commits to implement measures to address water pollution and drinking water quality in SA	17 Aug 2021
Gender Equality calls for opening sexual offences courts to speed-up gender based violence prosecutions	17 Aug 2021
Minister Blade Nzimande welcomes NSFAS funding for eligible and unfunded 2021 registered university students	17 Aug 2021
Western Cape government update on coronavirus and vaccines as at 16 Aug 2021	16 Aug 2021
MEC Albert Fritz joins women in Delft for a neighbourhood watch walkabout	16 Aug 2021
Employment and Labour on case postponement of Chinese national accused of alleged human trafficking and child labour	16 Aug 2021

Figure 2.4: South African government media statements repository

Using this repository we can first create a list of URLs that we will then use to web scrape the data presented in the Western Cape premier's press releases. This is achieved by setting up our script to find and interact with the appropriate elements of the repository [23]. In order to find the press release made by premier Alan Winde on a given day the only criteria that generally needs to be supplied is the keyword 'Alan' and the date of the press release. We will thus need to set up our script to find and enter text into the "Keyword", "Start date" and "End date" text boxes. We accomplish this by finding a unique identifier that will allow the web scraper to find each element. These unique identifiers can be formed using either HTML, CSS or XML. This is where the technical difference between XML and HTML comes into play, since in some situations it might be easier to use XML than HTML, even though HTML is just an implementation of XML [23]. The best identifier to use will vary from project to project (i.e. from web page to web page) [23]. Given that CSS selectors are quite comprehensive with regards to their syntax, they allow users to easily form a unique identifier for a particular element and were used throughout in the web scraping for this mini-dissertation [23]. The `Python` code to draw up the press releases made by the premier for the 9th of April 2020 (as an example) is shown in Figure 2.5.



```
# Click on "Keyword" field
# Find button to enter keywords by using its CSS selector
keyword_select = driver.find_element_by_css_selector('#edit-title-field-value')

# Click it
keyword_select.click()

# Enter keyword (Premier's first name is enough)
keyword_select.send_keys("Alan", Keys.ENTER)

# Change the date to the one we want
# Find fields for start and end date using CSS selectors
date1_select = driver.find_element_by_css_selector('#edit-field-gcis-speech-date-value-1-min-datepicker-popup-1')
date2_select = driver.find_element_by_css_selector('#edit-field-gcis-speech-date-value-1-max-datepicker-popup-1')

# Change start date
# Click it
date1_select.click()
# Enter start date
date1_select.send_keys("9 Apr 2020", Keys.ENTER)

# Change end date
# Click it
date2_select.click()
# Enter start date
date2_select.send_keys("9 Apr 2020", Keys.ENTER)

# Find "Search" button
search_button = driver.find_element_by_css_selector('#edit-submit-speeches-views')
# Click it
search_button.click()
```

Figure 2.5: Python code to make the web scraper display a desired press release

Once the desired press release is displayed on the page it only becomes necessary to extract the embedded web address. By repeating this process for each date in a desired range we can obtain a list of URLs to use for the second step in the web scraping: extracting the data. Fortunately the `pandas` package is able to extract all tables from a web page after being supplied with only the URL. Once all the data has been extracted and concatenated together, the final step (Data Organization) is to extract the data and store it in a usable format such as an Excel or CSV file [65].

2.1.2 The legality of web scraping

A subject that is often overlooked in web scraping projects is the topic of legality and ethics [66, 65]. It has happened numerous times in recent years that web scraping became the topic of heavy debate and legal disputes [66, 65, 23]. These disputes most often occur when the purpose of web scraping is for commercial gain, however it has happened that researchers attempted to scrape data that was not allowed and were subsequently berated for it through legal action [66, 65, 23]. We now briefly discuss the legality and ethics of web scraping.

In a recent study, Krotov et al. [65] showed that the literature on the legality of web scraping is severely lacking. In a search of five research databases for articles discussing web scraping they showed that only 5% of papers discussing web scraping explicitly discuss the legality of this practice [65]. This shows that many researchers neglect to even consider the legality of web scraping. While it is the case that web scraping exists in a “grey area” in the legal system, since there is no legislature that directly addresses it, it is still highly recommended that individuals interested in web scraping consider the legal repercussions of their actions beforehand [66, 66]. Legal disputes over web scraping have primarily occurred in the United States of America and the European Union and thus the discussion that follows will be set against the backdrop of these legal systems [66, 66, 23]. While these legal systems are not necessarily identical to that of South Africa, the principles are still applicable.

Arguably the most well known example of a legal dispute erupting because of web scraping is the case of LinkedIn vs. hiQ Labs [23, 65]. In 2017, a relatively young American startup called hiQ Labs was issued a cease and desist by LinkedIn when it was discovered that the former had been web scraping large amounts of data from the latter's website [65, 23]. LinkedIn also took further steps to prevent hiQ Labs' bots from being able to access their site, including banning their IP address [23]. The matter was taken to court, where LinkedIn argued that hiQ had violated the Computer Fraud and Abuse Act (CFAA) which prohibits "the intentional unauthorized access of a computer or access that exceeds authorization" [65]. The CFAA is often brought up in cases regarding web scraping [23]. The judge ruled in favor of hiQ Labs, ruling that the way in which the data was accessed was not illegal since it was made publicly available [23, 65]. When LinkedIn appealed the ruling in 2019 the court once again ruled in favor of hiQ Labs [65]. This example serves to show that even if the owner of a web site expresses explicitly that they do not allow web scraping, the law does not prohibit it. Many powerful companies have been involved in legal battles over web scraping, such as Facebook, Google and Craigslist [23]. Other legal and ethical concerns that are often referenced when with regards to the legality of web scraping include (but are not limited to):

- Breach of contract: Some sites insert clauses in their terms of use that prevent web scraping. However even if a user is then proven guilty of web scraping they are able to argue that they were not explicitly made aware of these terms of use [65, 23, 66]. For example the previously discussed COVID-19 dashboard could still be web scraped if the party scraping it simply defended their actions by explaining that they never saw the terms and conditions (it is necessary to click on a somewhat inconspicuous icon to bring them up). This is most likely the reason why the data displayed on the dashboard was changed as well. In order to claim breach of contract website owners also have to be able to prove damages [65].

- Copyright infringement: This legal issue arises when data is scraped and republished (usually with financial gain in mind) while that data is copyrighted by the website owner. However in such cases it then becomes dubious whether the website actually owns the data on its own site. For example, user data on Facebook does not actually (legally) belong to Facebook due to the fact that it is user created [65]. Furthermore, copyright infringement does not explicitly prevent the collection of data itself [65].

- Trespass to chattels: This legal term dictates that if a web scraper causes damage to the target server that it is scraping from, then the owner of the server can claim damages [65, 23, 66]. However, given that the damage must be material and easy to prove in court, this principle is rarely used in web scraping cases [65].
- Trade secrets: If web scraping is used to learn confidential business operation strategies then legal action can be taken against the web scraping party [65, 66, 23]. Uber has previously been accused of using web scraping to investigate the business practices of competitors [65].

Other topics are often mentioned in web scraping cases are privacy (both individual and organizational), impact on decision-making and diminishing value for the organization being scraped [65]. It is clear that the law has yet to be sufficiently updated to accommodate the novel approach to data collection that is web scraping. Despite this, there are some commonly accepted practices that web scrapers are urged to abide by to limit the risk of potential legal action being taken against them. As outlined by vanden Broucke and Baesens [23], these practices are:

- First check if there is an API - If the website provides an API for downloading the data, do not resort to using a web scraper. Only then use a web scraper if the API is lacking in some capacity (such as a pay-wall or throttled download quantities).
- Do not overload the server - Be considerate of the server that you are attempting to scrape data from. Include a waiting time between successive requests for new information so as to not overload (or potentially damage) the server.
- CAPTCHA - If data can only be accessed by completing a ‘Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA)’ (usually a simple test where users are asked to identify images containing a specific item) then it is safe to assume that the data should not be scraped. There are however offers available for software that automatically overcome these hindrances as well.
- Robots Exclusion Protocol - While this has limited legal value, it is an industry standard to have a “robot.txt” file embedded in websites. This file is meant to instruct web scrapers on which directories of the website they are allowed to scrape (and which ones they are not). In order to find this file

for any website, simply append "/robots.txt" to the web address of the root directory (i.e. the main address of the website, usually ending in .com or .co.za). For the "robots.txt" file for the South African government media statements repository please refer to Appendix A.

Krotov et al. [65] have compiled a list of questions that are meant to guide researchers in determining whether their prospective web scraping projects are legal and ethical in nature (however this is under no circumstances intended to serve as legal advice). After reviewing all the available information regarding the legality of web scraping we have determined that there is no legal reason to avoid scraping and utilizing the data that is available through the Western Cape premier's press releases.

2.1.3 NICD COVID-19 data

Upon successfully web scraping the COVID-19 case data using the Western Cape premier's press releases it was discovered that there was another viable source of COVID-19 case data for the Western Cape province. The national institute for communicable disease (NICD) has also made available COVID-19 case data at a district municipality level for each of the nine provinces of South Africa[7]. This introduces an interesting opportunity for additional analysis to be performed. Specifically, comparing the insights and modelling performance when utilizing two reputable datasets for the same study region and time period. We will thus conduct all the analysis in this mini-dissertation using both sources of COVID-19 data and determine whether there is any significant difference present in the results.

2.2 Population mobility data

Due to the manner in which COVID-19 is transmitted, the use of mobility data in spatial modelling is essential to capture the intrinsic spread through the population. A common source is mobile network location data, which has been utilized previously for epidemiological modelling [44, 18, 36, 138, 111, 137]. However, this data is difficult to obtain due to increasing privacy concerns. In addition, there are most often a number of network providers in a region, each with certain market share. Without data access from all, or at least the largest proportion of providers, representativeness and mobile device penetration will be limited and should be used with caution at high spatial resolutions. It is thus worthwhile to determine to what extent different sources of mobility data convey the same message of mobility within a region.

In this section we discuss previous applications of mobile network data to epidemiological modelling, with

[7] NICD COVID-19 database: https://1e.to/IS0xv2 (Accessed August 2021)

a focus on its application to the modelling of COVID-19. We then discuss the potential shortcomings of using mobile network data for epidemiological modelling both in general as well as in the South African context. Finally, we discuss the mobile network data utilized in this mini-dissertation.

2.2.1 Epidemiological modelling

The growing popularity and widespread use of mobile devices has led to massive amounts of data being produced at any given point in time all around the world. Mobile network data can be collected either passively by mobile services providers or through the use of mobile applications. The ease with which such large quantities of data can be gathered makes it very attractive for researchers.

Mobile network data has been used numerous times in the field of spatial epidemiology to model the spread of various diseases, including cholera [44, 18], dengue [36, 138] and malaria [111, 137]. Following the outbreak of the COVID-19 pandemic, the governments of various countries across the world began collecting mobile device user data in an attempt to aid the conception and implementation of non-pharmaceutical interventions [41, 132, 91, 95]. This data has since been used by researchers to clearly establish a correlation between population mobility and COVID-19 case numbers [95, 48, 57, 142].

Limitations of mobile network data exist. First and foremost of these is the issue of user privacy. Mobile device data could potentially be misused to identify specific individuals and thus network providers are often hesitant to provide researchers with such data [91, 53]. Such data is often aggregated to a low spatial resolution to prevent this as well as reduce noise, but this comes at the cost of some data specificity. Another potential drawback of mobile network data is high computational cost. Due to very high mobile device penetration rates, mobile network data may consist of a number of entries in the order of billions. The computational cost of processing such datasets is prohibitive, potentially preventing analysis.

2.2.2 Facebook Data for Good

Facebook Data for Good is a program that strives to provide tools and insight using privacy-protected data collected through various means such as Facebook and satellite imagery. The program describes their approach as follows: “We build privacy-preserving data products to help solve some of the world’s biggest problems”. Towards that end, they have made various forms of data that could be potentially useful to researchers interested in the COVID-19 pandemic publicly available to use[8]. This includes data such as population density maps, electrical distribution grid maps and measures such as “social connectedness” which quantifies to what degree two regions are connected through social connections on Facebook.

[8] This data may be accessed at: https://dataforgood.facebook.com/dfg/covid-19 (Accessed May 2021)

It is however difficult to obtain this data at a reasonably useful spatial resolution for South Africa. One source of their data that is made available at a district municipality level is the so-called "Movement Range maps". The data indicates the change in mobility, $F_i^{(t)} \in (-1, 1)$ (as a percentage), for district municipality i on a given day t over the period 1 March 2020 - 28 February 2021 relative to a one-week baseline calculated in February 2020. The daily values for each district municipality were calculated by determining the number of so-called "Bing tiles"[9] that each inhabitant visited on a given day (place of residence being determined by the location where users most often spend their nights). After incorporating some degree of noise, the average number of tiles visited by the inhabitants was determined and expressed relative to the baseline. The full description of how these values were calculated is as follows.

Let u represent a single individual and $U_{t,i}$ represent the total number of individuals within district municipality i at time t. The total number of Bing tiles visited by inhabitants of district municipality i is then

$$\text{total\_tiles}(U_{t,i}) = \sum_{u \in U_{t,i}} \min\left(\text{tiles}(u), 200\right).$$

Note that the maximum number of Bing tiles visited that a single individual can contribute is restricted to 200 in order to prevent high active users from skewing the data. In order to preserve user privacy, an error term was included by drawing from a Laplace distribution with parameters 0 and $\frac{F}{\epsilon}$ where $F =$ sensitivity parameter and $\epsilon =$ noise parameter as follows

$$\text{total\_tiles}'(U_{t,i}) = \text{total\_tiles}(U_{t,i}) + \text{Laplace}\left(0, \frac{F}{\epsilon}\right).$$

The average number of tiles per district municipality was then calculated as

$$\text{avg\_tiles}'(U_{t,i}) = \frac{\text{total\_tiles}'(U_{t,i})}{|U_{t,i}|}.$$

The mobility value for each district municipality and for each day was then finally expressed with respect to the baseline as

$$F_i^{(t)} = \frac{\text{avg\_tiles}(U_{t,i}) - \text{baseline\_avg\_tiles}'(i, \text{day\_of\_the\_week}(t))}{\text{baseline\_avg\_tiles}'(i, \text{day\_of\_the\_week}(t))}.$$

[9] Bing tiles definition: https://docs.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system (Accessed May 2021)

For further details regarding this data see https://1e.to/b9ihCD.

This data, which we will refer to simply as the “Facebook data” for the remainder of this mini-dissertation, is plotted in Figure 2.6. It is immediately clear how the transition to level 5 lockdown in late March 2020 heavily impacted the mobility of the general public, with the average district municipality experiencing 50% less population mobility.

Table 2.1: South Africa administrative boundaries

Administrative level	Spatial unit name	Number of spatial units
0	Country	1
1	Province	9
2	District municipality	52
3	Local municipality	213
4	Ward	4392

The administrative divisions of South Africa are summarised in Table 2.1. In order of increasing spatial resolution these are country, province, district municipality, local municipality, and ward, labelled as administrative levels 0 through 4 respectively. The Facebook data is available at district municipality level (i.e. administrative level 2) which is of relatively low spatial resolution.

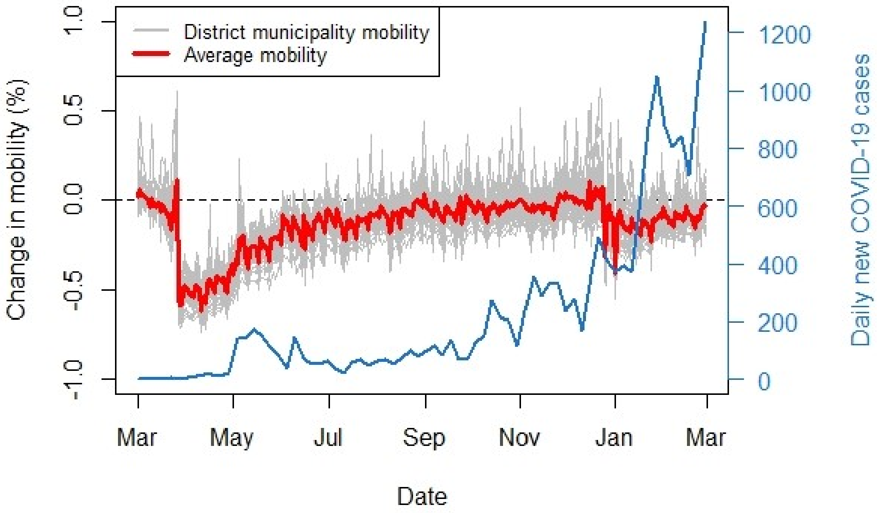


Figure 2.6: “Facebook for good” movement range maps data (1 March 2020 - 28 February 2021) relative to a baseline calculated in a week of February 2020.

2.2.3 Mobile network data

For this mini-dissertation, anonymised mobile network data was obtained from a local mobile network provider. The data was made available to a team of researchers from various institutions including the Department of Statistics at the University of Pretoria, the Foundation of Human Rights, the Council for Scientific and Industrial Research, the Medical Research Council, the Department of Statistics and Actuarial Science at the University of Witwatersrand as well as IBM Research, in order to aid in the fight against COVID-19. The work that follows is published in [100].

In South Africa, the mobile device penetration level is estimated to be as high as 95%[10]. The mobile network provider utilized in this mini-dissertation is one of the largest providers in the country, with an estimated market share of 42%.

Mobile devices operate by sending and receiving information from cell towers. When interacting with a cell tower we say that a device has "pinged" off a cell tower. A mobile device may ping off a cell tower by sending or receiving any kind of information, be it a phone call, text message or application notification. The mobile network data obtained for this research is obtained using the number of users whose mobile devices pinged off a cell tower within one ward (administrative level 4) on a given day and then later that day pinged off a cell tower in a different ward.

Formally, the data provides the number of mobile device users $m_{ij}^{(t)}$ that travelled to ward j from ward i on day t for the period 2 March - 12 May 2020 (thus spanning from before the lockdown was started up till level 3). The data is at administrative level 4, which is the highest spatial resolution reasonably possible while preserving some level of privacy of exact user location. To compare insights gained from this data and the Facebook data in Section 2.2.2, it would first be necessary to aggregate the mobile network data to the same spatial resolution which is administrative level 2. In South Africa, each ward has a unique 8-digit ID code. The first three digits of this code indicates the district municipality that the ward is a part of. For example, the ward ID 9344007 indicates that the ward is part of the district municipality with code 934. In order to aggregate the data to district municipality level, one could replace the ward IDs of the observations with their district municipality codes (i.e. only the first 3 digits), whereupon rows with identical origin and destination codes would be discarded. The mobile network data at administrative level 2 is thus given by

$$M_{I,J}^{(t)} = \sum_{i \in I, j \in J} m_{ij}^{(t)},$$

[10] See https://www.geopoll.com/blog/mobile-penetration-south-africa/ and https://www.icasa.org.za/uploads/files/State-of-the-ICT-Sector-Report-March-2020.pdf (Accessed May 2021)

where I and J are district municipalities and i and j are wards as previously indicated. Transitions contained within a single district municipality are thus discarded. Analysis revealed that this caused an average of 26% of daily observations to be discarded. More refined methods for aggregating to a lower spatial resolution may be produce better results. Future research could be conducted on how to aggregate spatial data to a higher resolution instead of a lower one, perhaps using methods akin to small area estimation or spatial micro-simulation (see e.g. [11, 96]).

The retained data is displayed in Figure 2.7. The representation differs to that of Figure 2.6 as the data provides transitions between regions in this case. We once again notice a sharp decline in population mobility in late March. The population of South Africa (mid-2021) is approximately 60.14 million[11], and yet the highest total number of inter-district municipality transitions on any given day was approximately 10 million (seen in Figure 2.7). It should be noted that the same individual can be responsible for multiple transitions and that some individuals could potentially possess multiple mobile devices. Literature does exist on the use of mobile network data to estimate population numbers, see e.g. [112]. Doing so is not within the scope of the research presented here but would be of value in testing mobile network data representativeness.

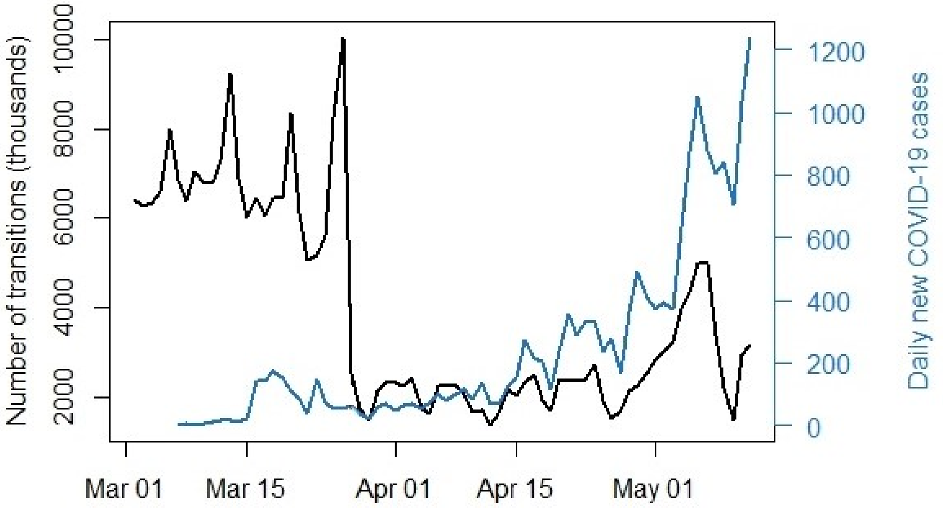


Figure 2.7: Mobile network data (2 March 2020 - 12 May 2020)

Despite the quality of available hardware[12], this process proved highly computationally expensive due to the number of comparisons that need to be run on billions of lines of data in order to create a spatial weight matrix for each day in the time period. This run-time would almost certainly improve upon proper

[11] Mid-2021 Statistics South Africa Population Report http://www.statssa.gov.za/publications/P0302/P03022021.pdf (Accessed August 2021)

[12] All analysis presented here was performed on a desktop computer running Intel Core i7 with a clock speed of 3.40GHz, a 64-bit operating system and 64 GB of installed memory.

parallelization or distributed computing due the data and relevant operations being able to be performed in parallel.

2.3 Conclusion

In this chapter we have discussed the various data that will be utilized in this mini-dissertation as well as how they were obtained. Modelling a complex phenomenon such as the spatial spread of COVID-19 requires the use of various types of data. This introduces numerous issues that must be considered such as missing values, how to compare data at different resolutions and scales, computational cost as well as a lack of willingness of certain parties to make data publicly accessible. While most of these represent issues that are essentially part of the ordinary research workflow, the last of these concerns is generally left completely out of researchers' hands.

Data acquisition for this mini-dissertation proved to be far more difficult than initially anticipated. This is rather disappointing given the emphasis that has been placed in academia and media on delivering more COVID-19 research. It is understood that data capturing, cleaning and storage can represent a challenge in a developing country with limited technology and resources such as South Africa. However, the lack of any transparency on the behalf of government bodies with regards to COVID-19 case data renders such research highly restrictive for any parties not affiliated with these organizations.

Most local governments appear to make no attempt at sharing data at high spatial resolution with the public. It could be the case that some of these parties simply do not have the means or the expertise to gather the required data in a timely and accurate manner. However others clearly have access to such data but refuse to make it publicly available for research purposes for largely unspecified reasons (see Section 2.1). A flawed compromise can often be observed where government institutions generally prefer to provide information through infographics[13] instead of machine-readable text that may be used for analysis. While such means of communicating information may be user-friendly and easy to understand for a human observer, they introduce the need for increased efforts by researchers to properly utilize the data. Such efforts have lead to the forming of collaborative groups dedicated to collecting and storing COVID-19 data[14]. While such endeavours by researchers are commendable, it illustrates that COVID-19 research within South Africa is complicated by matters that are not present in other settings and may be

[13] Infographic (noun): a visual presentation of information in the form of a chart, graph, or other image accompanied by minimal text, intended to give an easily understood overview, often of a complex subject (https://www.dictionary.com/browse/infographic)

[14] For example the work of Marivate and colleagues, information available through their GitHub page (https://github.com/dsfsi) and COVID-19 dashboard (https://dsfsi.github.io/covid19za-dash/).

prevented through increased government transparency.

Other sources of COVID-19 data for South Africa are rare but also unreliable, either being wholly incorrect[15], at a spatial resolution that renders it unusable[16] or attempting to monetise the data for profit[17]. We believe this contributes significantly to slowing the progress of research conducted in the country and prevents insight that could otherwise have aided in combatting the pandemic.

Due to our society's increasing reliance on instant communication and mobile devices, mobile network providers have access to an unimaginable wealth of data that could assist tremendously in any form of spatial research. Generally, user privacy is cited as the primary reason why such data is not made publicly available. However we would argue that even if such data were to be published at district municipality or even ward level that it would not pose a significant risk to any individual's privacy as it would not be possible to identify any single individual. It is a fact that epidemiological research endeavours such as those concerning COVID-19 benefit greatly from the inclusion of a spatial component, it being deemed essential in some cases. In this mini-dissertation we were able to obtain data from one mobile network provider at a restricted range of observed dates and without a majority representation of the population under study. While greater quantities of mobile network data would represent an increased need for better hardware and improved coding practices, it would also dramatically increase the accuracy and credibility of any analysis.

It is our hope that future research endeavours can benefit from revised regulations and policies. Preferably those that encourage organizations with invaluable data, both public and privately owned, to be more forthcoming and transparent with their data in a true show of support for the scientific community and data-driven research at large.

[15] Global.health: https://global.health/ (Accessed September 2021)

[16] Wikipedia: https://en.wikipedia.org/wiki/COVID-19_pandemic_in_South_Africa (Accessed September 2021)

[17] statista: https://www.statista.com/statistics/1108670/coronavirus-cumulative-cases-in-south-africa/ (Accessed September 2021)

Chapter 3

Mobility

The guiding principle within the field of spatial statistics is that “everything is related to everything else, but near things are more related than distant things” [129]. This expresses the fact that observations made across a geographical space often exhibit some form of dependence related to the distance between the locations where these observations were made. We refer to this as “spatial dependence”. This idea that all observations share some form of dependency diverges from the assumption of independence that is often made in other sub-fields of statistics.

When a particular phenomena exhibits evidence of spatial dependency, researchers are tasked with employing techniques that take this dependency into consideration or run the risk of producing biased results [121, 40]. In the 1850’s, crude spatial analysis was able to show that the spread of cholera through a neighbourhood in Soho, London, was caused by a water-borne virus (instead of it spreading through the air as previously believed) [27]. Within the context of COVID-19, not considering the spatial dependency between different regions could lead to inaccurate conclusions regarding the causes of the spread of the virus. A region could experience a surge in new cases, not due to the behaviour of its own inhabitants, but those from other regions such as migrant workers and tourists. When the spatial dependency between observation locations is not taken into consideration, conclusions could be that the observations are random when truly there is a distinguishable pattern [40]. In the case of an infectious disease that is spread through physical contact it is clear that regions that are closer together (or rather the inhabitants of these regions) will play a larger role in determining their respective infection rates than regions that are farther apart.

In this chapter we outline the methodology used to incorporate spatial dependencies into the modelling process. We first discuss standard approaches used in previous studies. We then propose new approaches used to represent spatial dependency within the study region using the previously mentioned population mobility data as published in [100].

3.1 Spatial weight matrices

To incorporate spatial dependencies, models with a spatial element allow spatial units to be more strongly (or weakly) correlated with one another based on some select criteria that is deemed suitable for the phenomenon being modelled. This is achieved through the use of a spatial weight matrix (sometimes called a "spatial mobility matrix" or "distance matrix") usually denoted by $\boldsymbol{W}$ [121, 50, 6, 13, 40, 3].

Definition 1 (Spatial weight matrix)**.** *Consider a system S of n spatial units, labeled $i = \{1, 2, \ldots, n\}$ within the study region. A spatial weight matrix is an $n \times n$ matrix $\boldsymbol{W} = [w_{ij}]$ satisfying*

1. $w_{ij} \geq 0$
2. $\sum_{j=1}^{n} w_{ij} = 1 \quad \forall \quad i \in S.$

This matrix formally defines an expression of spatial dependency between spatial units [121, 50, 6, 13]. Simply put, the spatial weight matrix is constructed in such a way so that entry w_{ij} quantifies the amount of spatial influence that spatial unit i exerts on spatial unit j [121, 50, 6, 13]. Spatial weight matrices are often used in the field of econometrics [6]. Examples of two popular spatial models that incorporate a spatial weight matrix are the spatial lag and spatial error models (Equations 3.1 and 3.2 respectively) [121].

$$\boldsymbol{y} = \rho \boldsymbol{W} \boldsymbol{y} + \boldsymbol{X} \boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{3.1}$$

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{X} \boldsymbol{\beta} + \boldsymbol{\epsilon} \\ \boldsymbol{\epsilon} &= \boldsymbol{\lambda} \boldsymbol{W} \boldsymbol{\epsilon} + \boldsymbol{u} \end{aligned} \tag{3.2}$$

In Equations 3.1 and 3.2 we have that $\boldsymbol{y}$ and $\boldsymbol{X}$ represent spatially dependent outcome and explanatory variables respectively [121]. Furthermore, $\boldsymbol{W}$ is a spatial weight matrix and ρ, $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$ all represent model coefficients. Finally, $\boldsymbol{\epsilon}$ and $\boldsymbol{u}$ represent error terms [121]. Here, the spatial weight matrix allows the dependent variable to be a function of "spatially lagged" values for other measurements of the same variable or error terms [121]. These spatial lagged effects will be determined by the way in which the spatial weight matrix is constructed.

Such matrices are frequently restricted to being symmetrical to simplify estimation [121]. However symmetry is not necessarily required and can result in a less realistic representation of spatial dependency [13]. Another convention frequently utilized is that $w_{ii} = 0$ for all i to exclude the possibility of so-called "self-influence" [121]. Once again however this is not required in order for $\boldsymbol{W}$ to be considered a valid spatial weight matrix, however this assumption is made in the vast majority of spatial regression work and often simplifies estimation and/or interpretation [13]. Non-zero diagonal entries can be interpreted as quantifying the resistance that each spatial unit has against influence from the other spatial units [13, 71].

Performing row-standardisation on the matrix allows the connectivity of different spatial units to be compared [50, 13, 121]. By row-standardizing we can interpret each w_{ij} as the fraction of total spatial influence on spatial unit i that can be attributed to spatial unit j [50, 71]. By column-standardizing we can also quantify the proportion of its total spatial influence exerted by spatial unit j on each of the other spatial units [71]. However there exists some debate as to whether or not this is advisable. On one hand this could render the matrix asymmetrical and can result in too much spatial dependency being attributed to very small regions [6, 45]. On the other hand it also simplifies the interpretation of the weights [50].

Before continuing it is worth briefly clarifying how spatial matrices will be utilized in this mini-dissertation as we will not be utilizing the spatial weight matrices in the regression functions given at (3.1) and (3.2), which is how they are most often applied. The main function of spatial weight matrices within our chosen SEIR modelling framework is to allow the risk of exposure to an infectious disease within a spatial region to be influenced by relevant factors within other regions [24]. This will be discussed in further detail in a later chapter, however the basic form of their use in this mini-dissertation is given by the following:

$$\pi_{ij}^{(SE)} = 1 - \exp\left[\left\{ -\eta_{ij} - \sum_{z=1}^{Z} \rho_z \left(\boldsymbol{D_z} \eta_{ik}\right) \right\}_{k \neq j}^{h_i}\right], \tag{3.3}$$

where $\boldsymbol{D_z}$ is a spatial weight matrix (also sometimes known as a "distance matrix" hence the choice of notation [24]). The interpretation of this functional form is that the probability of exposure to COVID-19 within spatial region j at time i, $\pi_{ij}^{(SE)}$, follows an exponential distribution. The parameter of this exponential distribution is determined by the exposure process within spatial region j, η_{ij}, as well as a proportion of the exposure process within other spatial regions $\eta_{ik}, k \neq j$. The proportion of the exposure process that is "shared" by other spatial regions is indicated by the spatial autocorrelation parameter ρ_z

and is determined by the spatial weight matrix $\boldsymbol{D}_z$ (note that the functional form also allows for multiple expressions of spatial dependence, i.e. spatial weight matrices, to be included). Finally, the term h_i is a temporal offset, indicating the frequency of observations (usually set to 1 day).

Therefore, since we are not utilizing the functional forms specified at (3.1) or (3.2) or any modification thereof we do not have easy access to the rich theoretical framework that's been established to improve the estimation and complexity of spatial weight matrices (see e.g. [70, 102, 84]) as they were derived with a very specific structure in mind. Rather, in this mini-dissertation the methods utilized to derive and structure the spatial weight matrices will be guided by factors known to contribute to the spread of infectious diseases as well as information pertaining to the study region.

Spatial weight matrices can be constructed using numerous definitions and conventions, some of which are fairly simple and others which are more complex. The construction of these matrices has been a matter of debate essentially since their creation and there is still no universally accepted method to construct them or to know beforehand which one will produce the best fit [6, 13, 40]. Generally, spatial weight matrices are created through the use of distance measures, the concept of contiguity or empirically derived using geostatistical data [3, 29]. We now discuss various methods of creating spatial weights matrices in increasing order of complexity.

3.1.1 Contiguity

Spatial data is generated by a very dynamic set of activities. In principle, any type of data may be regarded as spatial in nature if observed at various locations that are recorded. Let a spatial process in d dimensions be generally defined as

$$\Big\{Z(\boldsymbol{s}) : \boldsymbol{s} \in D \subset \mathbb{R}^d\Big\} \tag{3.4}$$

where Z denotes the variable observed at coordinates $\boldsymbol{s} : d \times 1$ and D indicates the domain or study region [114]. There are primarily three types of spatial data most often encountered and discussed in literature: point patterns, lattices and geostatistical data [114]. We can distinguish between these data types by the characteristics of the domain D [114]. In cases where the domain D is fixed and discrete we refer to the spatial data as lattice data [114]. This is the case when our study region is sub-divided into smaller regions based on attributes such as postal code, census tract or administrative boundaries [114]. This means that any data collected at a ward, provincial or even country level can be considered lattice data.

Consider a scenario where only lattice data (sometimes referred to as areal data [114]) for a particular phenomenon is available, that is to say, we have observations for sub-regions of our study region but not exact geographic coordinates for observations. Let each sub-region within the study region be considered a spatial unit. If there is reason to suspect that a spatial unit's neighbours (i.e. those spatial units that it shares a boundary with) are the only spatial units that exert any kind of spatial influence then we may proceed to assign weights to each of these neighbours and no weight to any other spatial units. In order to do this however we first require an appropriate definition of what it means for spatial units to be neighbours. Let $\mathrm{bnd}(i)$ indicate the boundary of spatial unit i, we can then declare spatial units i and j neighbours if,

$$\mathrm{bnd}(i) \cap \mathrm{bnd}(j) \neq \emptyset$$

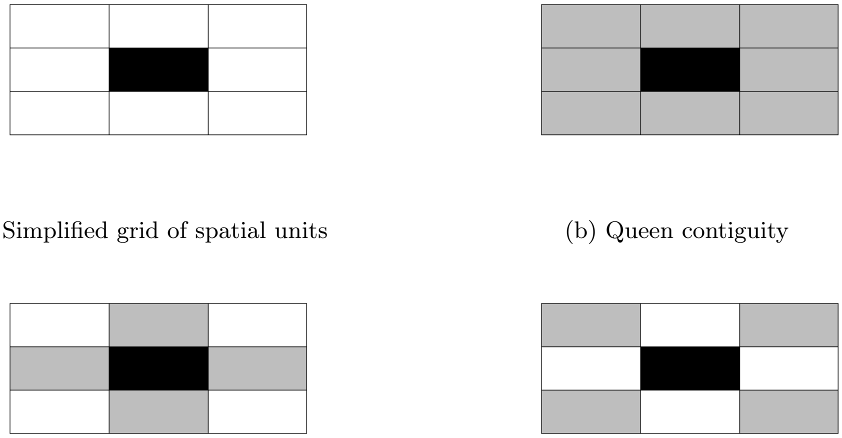
i.e. if there is an intersection between their respective boundaries [6]. If two spatial units have a common boundary they are known as *contiguous* [6]. By evaluating the contiguity for every pair of spatial units we can derive a binary contiguity matrix, where

$$w_{ij} = \begin{cases} 1 & \text{if } \mathrm{bnd}(i) \cap \mathrm{bnd}(j) \neq \emptyset \\ 0 & \text{if } \mathrm{bnd}(i) \cap \mathrm{bnd}(j) = \emptyset. \end{cases} \tag{3.5}$$

This particular form of contiguity if referred to as *queen contiguity.* Using this structure, any spatial units that have connecting boundaries will be designated as neighbours and subsequently assigned a value of 1 in the contiguity matrix. A potential improvement can be made however by requiring that spatial units share a minimum length of boundaries before being classified as neighbours. If we let l_{ij} be the length of shared boundary between spatial units i and j, then we can define *rook contiguity* as

$$w_{ij} = \begin{cases} 1 & \text{if } l_{ij} > 0 \\ 0 & \text{if } l_{ij} = 0. \end{cases} \tag{3.6}$$

The concept of contiguity weight matrices (as well as how they differ) can be easier understood through a visual aid. Consider a simplified case where we have 9 spatial units arranged in a grid as in Figure 3.1(a) where each cell represents a spatial unit. We proceed to evaluate the contiguity for the black cell (i.e. spatial unit 5 if counting along the rows) using different definitions. In the other sub-figures of Figure 3.1 we have the grid replicated 3 times, each time the cells that are classified as neighbours of the black cell according to a different contiguity definition are coloured grey. In Figure 3.1(b) we see that under queen contiguity all cells that make contact with cell 5 are considered to be its neighbours regardless of



(a) Simplified grid of spatial units

(b) Queen contiguity

(c) Rook contiguity

(d) Bishop contiguity

Figure 3.1: Neighbours under different contiguity structures

the degree to which they make contact with the black cell. In this case we would have each entry in the fifth row of $\boldsymbol{W}$ equal to 1 (aside from the fifth entry which we set to 0 by convention [50]). After row standardizing the fifth row of $\boldsymbol{W}$ is given by:

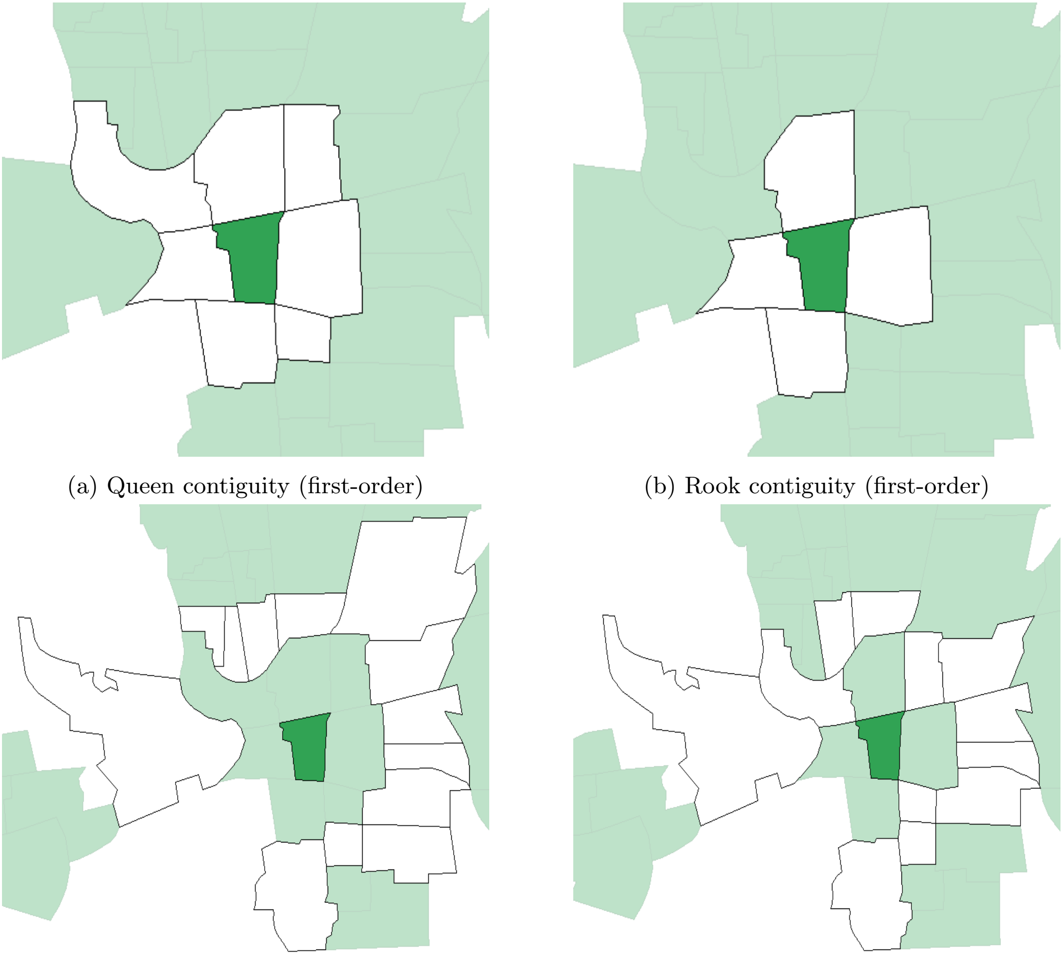
$$w_{5\cdot} = \left[\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, 0, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}\right].$$

For rook contiguity (shown in Figure 3.1(c)) we note that only cells that share a boundary with cell 5 are considered its neighbours. Cells that only connect to cell 5 at its corners are not considered to be neighbours as their boundaries intersect at only a single point. More generally, under rook contiguity only spatial units in the 4 cardinal directions are defined to be neighbours [50]. In this case the fifth row of $\boldsymbol{W}$ is given by:

$$w_{5\cdot} = \left[0, \frac{1}{4}, 0, \frac{1}{4}, 0, \frac{1}{4}, 0, \frac{1}{4}, 0\right].$$

Lastly, with bishop contiguity (shown in Figure 3.1(d)) only those cells that share a common vertex with cell 5 are considered to be neighbors [6]. We can note that the set of neighbours under bishop contiguity is merely the set difference between the set of neighbours under queen contiguity and rook contiguity.

These basic ideas can also be applied to more complex polygons with ease. In Figure 3.2(a) and (b) we depict the neighbours of an area of Columbus, Ohio under queen and rook contiguity. Observe how the spatial units that lie diagonally of the spatial unit of interest are not considered neighbours under rook contiguity. The contiguity we have thus far been concerned with is simply first-order contiguity, whereby we only consider the immediate neighbours of a given spatial unit or *first-order* neighbours [73]. We can easily extend this concept to second-order contiguity where the immediate neighbours of first-order neighbours are considered to be neighbours [73, 7]. An example of this is shown in Figure 3.2(c) and (d). Note that now, the first-order neighbours are no longer neighbours of our region of interest. Rather, the first-order neighbours *of* the first-order neighbours are considered to be neighbours of the region of interest. While not relevant specifically to the analysis in this mini-dissertation, these techniques can be applied in a similar fashion to spatial locations defined on a network [29, 30, 94]. For a discussion on such techniques please refer to Appendix B.



(a) Queen contiguity (first-order)

(b) Rook contiguity (first-order)

(c) Queen contiguity (second-order)

(d) Rook contiguity (second-order)

Figure 3.2: Queen and rook contiguity of section of Columbus, Ohio

3.1.2 Distance

Instead of (or perhaps in addition to) simply considering the bordering spatial units as having a spatial influence, we can also allow the spatial influence of unit i on unit j be some function of the distance between them d_{ij} [50, 6]. The distance between spatial units can be defined with respect to any valid measure such as Euclidean, Manhattan Block, general Minskowski or geodesic distance [6, 25]. Using this method allows for a wide variety of functional forms to be implemented, some of the more simplistic variants follow as they are given in [6].

1. Radial distance. Every spatial unit within a distance of d_c from spatial unit i is assigned a weight of 1 while all other spatial units receive a weight of 0. This is equivalent to drawing a circle of radius d_c at the centroid of a given spatial unit and considering any other spatial units with centroids contained within the circle as neighbours. This choice is however biased with regards to the location of centroids [40].

$$w_{ij} = \begin{cases} 1 & \text{if } d_{ij} < d_c \\ 0 & \text{if } d_{ij} > d_c. \end{cases} \tag{3.7}$$

2. Power distance. This functional form allows the spatial influence of a spatial unit to decline as the distance between the relevant spatial units increases. The power of the function determines how rapid this decline occurs,

$$w_{ij} = d_{ij}^{-\alpha} \ , \ \alpha > 0. \tag{3.8}$$

3. Exponential. Similarly to the power function, this form allows the spatial influence of spatial units to decline at a desired rate as distance between spatial units increases, however here the maximum weight that can be allocated to any spatial unit is restricted to 1,

$$w_{ij} = \exp(-\alpha \cdot d_{ij}) \ , \ \alpha > 0. \tag{3.9}$$

4. Double power. An example of a function that affords a greater degree of flexibility. It allows spatial influence to diminish at a varying rate as distance increases due to its bell shape,

$$w_{ij} = \begin{cases} \left[1 - \left(\frac{d_{ij}}{d_c}\right)^k\right]^k & \text{if } 0 \leq d_{ij} \leq d_c \\ 0 & \text{if } d_{ij} > d_c \end{cases} \tag{3.10}$$

for some value k.

In addition to the basic functions mentioned above, researchers in the field have suggested a plethora of functional forms utilizing the distance between spatial units [6]. These models are often set up with a very clear goal in mind and are thus very specific. For example, in [71] Leenders discusses a model variation applicable to the social sciences where the amount of political resources that spatial unit i shares with j is incorporated. Weights can also be scaled by attributes such as spatial unit area to restrict the spatial weights for smaller regions. A particular model that is often employed in studies concerning economic activity is the gravity model which takes into consideration the sizes of spatial unit samples [25, 4, 71]. The gravity model is given as,

$$w_{ij} = \sqrt{\frac{n_i \cdot n_j}{\delta_{ij}^2}} \tag{3.11}$$

where n_i and n_j indicate number of observations in spatial units i and j respectively and δ_{ij} is the geodesic distance between spatial units i and j.

Evidently there is a large degree of freedom available when specifying these types of weights and any non-increasing function of distance can be utilized at the discretion of the researcher [71]. Most studies utilize a weight matrix comprised of both distance and contiguity relationships [6].

3.1.3 Geostatistical

The spatial weight matrices defined thus far were all solely dependent on the topological layout of a study region and did not involve any actual observations for the phenomena being studied. Spatial weight matrices can however be estimated through the use of geostatistical data [3, 50]. Consider a scenario where we have n georeferenced observations on some variable of interest Z (i.e. each observation has attached to it some method of indicating the location where the observation was made). The geostatistical variogram is defined as half the average squared difference between observations made a distance d apart [50, 133] and can be estimated as

$$\hat{\gamma}(d) = \frac{1}{2N} \sum \left\{ Z(\boldsymbol{u}) - Z(\boldsymbol{v}) \right\}^2 \tag{3.12}$$

where $Z(\boldsymbol{u})$ and $Z(\boldsymbol{v})$ are the observations made at spatial locations $\boldsymbol{u}$ and $\boldsymbol{v}$ respectively, there are N pairs of observations made a distance of d apart and the sum is taken over all these pairs. High (low) values for $\hat{\gamma}(d)$ indicate that observations made at a distance of d tend to exhibit large (small) variation in their values for Z [50]. A necessary assumption for the variogram to exist is that of intrinsic stationarity [50, 133]. As a special case, when the Z values are all independent we expect $\gamma(d) = \sigma^2$ for all $d \neq 0$ [133]. A measure of autocorrelation between the observations at different locations can then be formed as,

$$\rho(d) = 1 - \frac{\hat{\gamma}(d)}{\sigma^2}. \tag{3.13}$$

This value will range between 0 and 1, as the distance d increases the variation between observations will also increase until it roughly approximates the variation in the data as a whole [50]. Therefore $\rho(d) \to 0$ as $d \to \infty$. Using this, we can construct our spatial matrix as

$$w_{ij} = \rho(d_{ij}). \tag{3.14}$$

So that for each pair of spatial units we assign a quantity of spatial influence equal to their estimated correlation based on how far apart they are in space [50]. In 1995, Ord and Getis developed the G_i^* local statistic that can be used to test for evidence of clustering and hot-spot analysis [92]. The test statistic is given as

$$G_i^*(d) = \frac{\sum_{j=1}^{N} I(d_{ij} \leq d) \cdot (x_j - \bar{x})}{S\sqrt{\frac{1}{N-1} \cdot \left[N\left(\sum_{j=1}^{N} I\left(d_{ij} \leq d\right)^2\right) - \left(\sum_{j=1}^{N} I\left(d_{ij} \leq d\right)\right)^2\right]}} \tag{3.15}$$

with

$$S = \sqrt{\frac{\sum_{j=1}^{N} x_j^2}{N} - (\bar{x})^2} \tag{3.16}$$

where x_j indicate observations of a particular phenomenon. Under the null hypothesis of no spatial autocorrelation at a distance of d the test statistic has a normal distribution with mean zero and variance 1 [3]. Significantly positive (negative) values of the G_i^* statistic indicate evidence of positive (negative) clustering at region i. If the statistic is not significant then there is no evidence of clustering at region i [51]. In 2004 Getis and Aldstadt derived a method of creating a spatial weight matrix by utilizing this statistic [50]. For each region of interest i we evaluate the G_i^* statistic for a series of distances $d_1 < d_2 < d_3 \ldots$ until a distance d_c is found for which the statistic does not increase absolutely [50]. This distance is then considered to be the distance at which no further evidence of clustering exists for that spatial region (i.e. where the diameter of the cluster ends) [50]. Let d_{NN1} denote the distance between a spatial unit and its nearest neighbour, then we have the following cases.

When $d_c > d_{NN1}$,

$$w_{ij} = \begin{cases} \frac{|G_i^*(d_c) - G_i^*(d_{ij})|}{|G_i^*(d_c) - G_i^*(0)|} & \forall \quad j \text{ where } d_{ij} \leq d_c \\ 0 & \text{otherwise.} \end{cases}$$

When $d_c = d_{NN1}$

$$w_{ij} = \begin{cases} 1 \quad \forall \quad j \text{ where } d_{ij} = d_c \\ 0 \quad \text{otherwise.} \end{cases}$$

When $d_c = 0$

$$w_{ij} = 0 \quad \forall \quad j.$$

In 2006 Aldstadt and Getis [3] expanded upon the use of the G_i^* statistic as a means of constructing spatial weight matrices and cluster identification by deriving “A Multidirectional Optimum Ecotope-Based Algorithm” or simply *AMOEBA* for short. Similarly to the previously mentioned use for G_i^* [50], this algorithm relies on calculating the test statistic for a series of distances for each spatial unit until the test statistic fails to increase absolutely [3]. This equation however differs in that rather than simply evaluating a series of distances for spatial unit i, we consider each combination of neighbouring spatial units that result in G_i^* increasing absolutely and take the most homogeneous of these [3]. For the full algorithm please refer to [3]. For both of these two methods there exists a non-zero probability of obtaining rows of all 0 in $\boldsymbol{W}$ [3, 50]. If a spatial weight matrix such as this were to be used for some other purpose (such as being included in a spatial epidemiological model) special consideration would have to be made regarding how to handle such cases.

As evident by the variogram and G_i^* examples above, using geostatistical methods effectively simplifies to using clustering techniques or high spatial autocorrelation to decide upon the appropriate weights in $\boldsymbol{W}$. This method could be seen as more appropriate for most endeavors requiring the construction of a spatial weight matrix since it does not make any *a priori* assumptions regarding the structure of spatial influence [3].

In a recent paper Ejigo and Wencheko proposed a spatial weight matrix that allows for the incorporation of covariates into the calculation of weights [40]. The reasoning behind this method is to not only consider geographical proximity but also covariate based proximity (i.e. similarity) when calculating weights in order to account for non-stationary spatial risk factors [40]. For example, in this study it was found that soil samples taken close together tended not to be highly correlated, however soil samples taken the same distance away from the local river were found to indeed exhibit high correlation. Therefore the distance between the observation sites and the river were included into the weights, which were calculated as follows

$$w_{ij} = \exp(-(\alpha \cdot u_e + (1-\alpha) \cdot d_{ij})) \tag{3.17}$$

where e_i is the covariate information for spatial unit i, $u_e = |e_i - e_j|$ is the absolute difference between the covariate measurements for the two spatial units and $\alpha \in [0, 1]$ indicates how much weight is allocated to the distance and covariate proximity respectively.

Note that if $\alpha = 0$ then the weights simply reduce to those calculated using the exponential distances as given by Equation (3.9). Note that in this case the covariate that was included was exogenous and formed part of the topological features of the study region. Given that this study is quite new, it has yet to be shown whether this method will be viable for more "human" problems such as the focus of this mini-dissertation. This method could offer a chance to incorporate covariates such as average local income, demographic composition or local unemployment rates into the estimation of spatial weights. There are two caveats however. The resulting spatial weight matrix does not serve as a measure of how spatial units influence one another but rather how similar they are [40]. It will thus be necessary to adapt this idea somewhat. Furthermore, it will however be necessary to first establish that whatever variables we use do indeed have an impact on the outcome being modelled [40]. The author mentions in particular that future study might focus on incorporating multiple covariates.

3.2 New methodology

In this section we will present and discuss methodology for constructing spatial weight matrices that will be utilized in this mini-dissertation. We discuss both standard approaches utilized often in literature as well as present new methodology as published by Potgieter et al. [100]. We present the theory and application simultaneously.

As stated previously, when modelling a spatially dependent phenomenon it is imperative that an accurate representation of spatial dependence be formulated. However, there is no established and undisputed methodology for the construction and evaluation of spatial weight matrices [6]. We must thus rely on the interpretation and comparison of alternative spatial weight matrix definitions in order to select anything resembling an "optimal" construction. In this section we outline the construction of four alternative spatial weight matrices as provided in [100]. With the exception of one, these spatial weight matrices were constructed using the previously discussed population mobility data (obtained from a local mobile network provider and Facebook).

The results will be presented at a district municipality level following prior discussion regarding the spatial resolution of the data. Figure 3.3 illustrates the 52 district municipalities of South Africa. The four largest cities in the country are Tshwane, Johannesburg, Durban and Cape Town, situated in the City of Tshwane, City of Johannesburg, eThekwini and City of Cape Town district municipalities respectively as indicated in colour in Figure 3.3. These four cities are the focal points of economic activity and travel in the country, and it is thus logical that they would play a substantially larger role in the transmission of the virus than other municipalities.

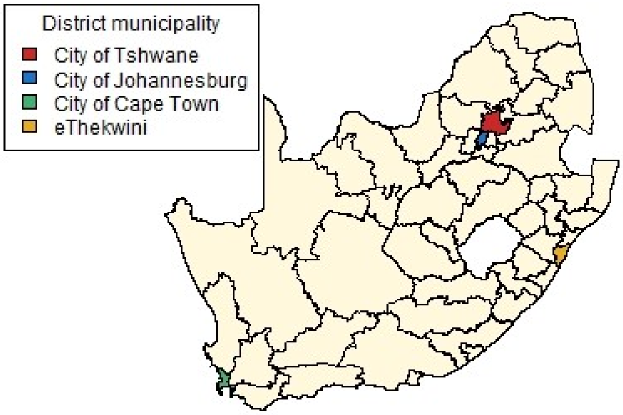


Figure 3.3: South African district municipalities (locations of four largest cities indicated in colour)

3.2.1 Method 1 - Exponential distance

The exponential distance definition of a spatial mobility matrix is used frequently in studies involving spatial autocorrelation, and is a popular choice in spatial econometrics [121, 6, 40, 3]. This matrix is included here to draw comparisons between it and more data-driven models. The entries of the spatial weight matrix are given by Equation (3.9) with the optional parameter α determining the rate at which spatial associations decline as distance increases. For simplicity we let $\alpha = 1$ in this mini-dissertation. The entries of this matrix are thus given by

$$w_{ij} = \exp(-d_{ij}) \tag{3.18}$$

where d_{ij} is the Euclidean distance between the centroids of district municipality i and j. Under this model, district municipalities are most strongly spatially correlated with the districts that are closest to them geographically. The entries of this matrix are illustrated over a map of South Africa in Figure 3.4. Note that in order to improve ease of readability, spatial weights smaller than 5% will not be shown for any of the results presented in this section. Since spatial weight matrices are calculated at a daily basis, all illustrations of this kind throughout this section illustrate the average daily spatial weight matrix (either over the entire study period or during different stages of lockdown). Note that since this method depends only on geographical distance no temporal component can be incorporated and thus the spatial weight matrix is constant.

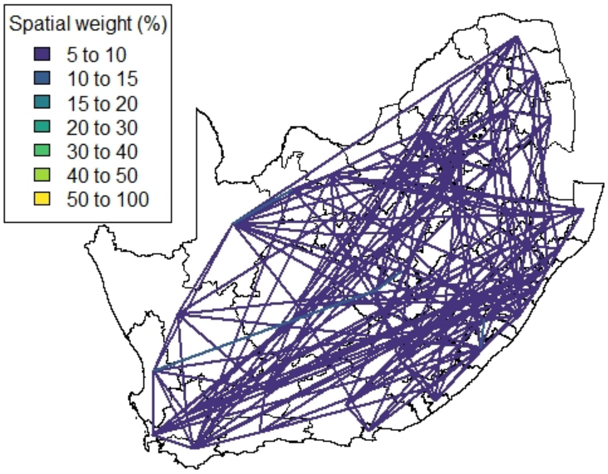


Figure 3.4: Method 1 spatial weights (weights $\leq 5\%$ not shown)

We note that there are no significantly large spatial associations within this matrix. Instead, each district municipality appears to have a set of neighbours, each of which it has an approximately equally strong spatial association. This representation is thus very simple and easy to understand but is not very realistic. It unfortunately offers no insight into the spatial autocorrelation within the study region and is instead only used for simplicity.

3.2.2 Method 2 - Mobile network data

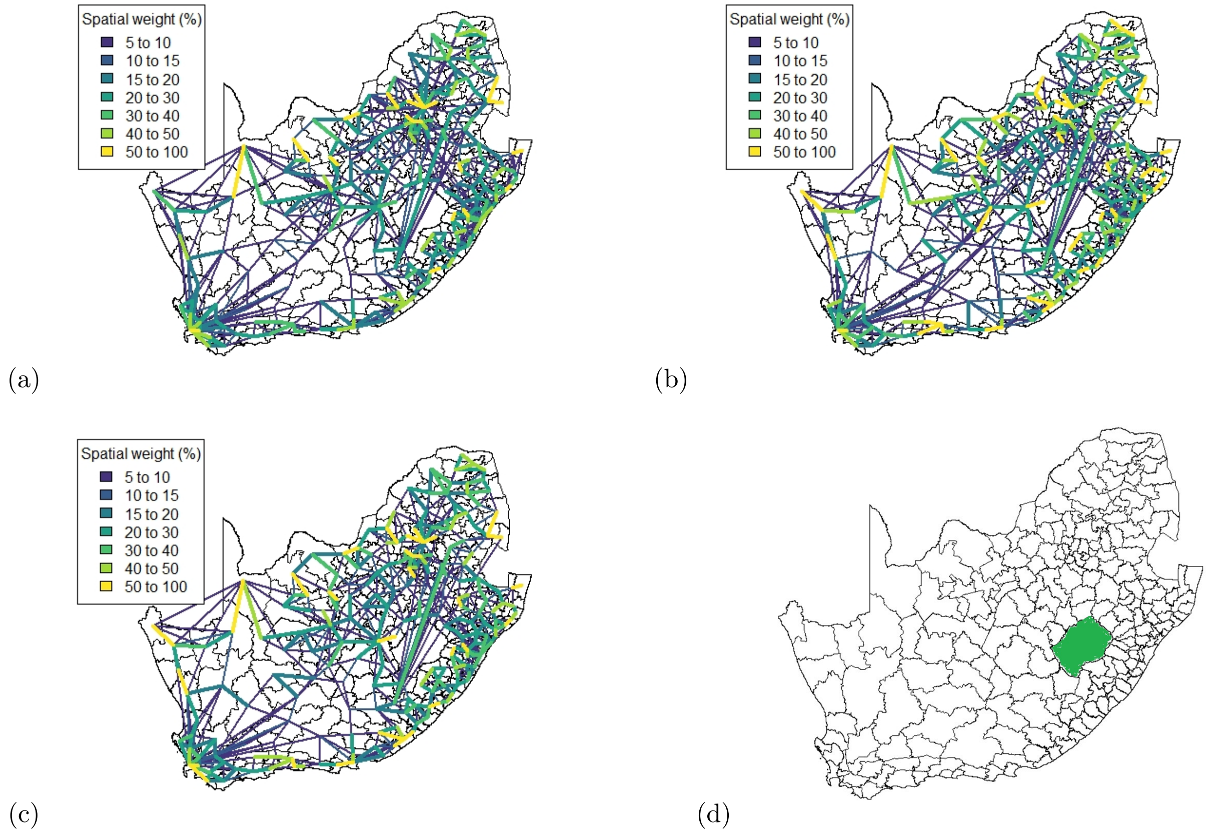
The mobile network data indicates the number of individuals that travelled from district municipality I to district municipality J on day t. These entries are used to construct a spatial weight matrix as follows,

$$w_{ij}^{(t)} = M_{IJ}^{(t)}. \tag{3.19}$$

After which the rows of the matrix are row-standardized. This model expresses spatial weights as a function of the amount of flux (both inwards and outwards) occurring at a spatial region, and is sometimes referred to as a spatial interaction matrix [13]. District municipalities where more (fewer) individuals travelled to other district municipalities will thus have a larger (smaller) effect on other district municipalities.

Since this data was initially made available at a ward level it is worth calculating the spatial weight matrix at various spatial resolutions in order to study the interpretations that can be made at various administrative levels. For this reason we first determine the spatial weight matrices at a local municipality level before proceeding towards a district municipality level.

Figure 3.5 illustrates the resulting spatial weight matrix for every level of lockdown that the mobile network data spans at a local municipality level. This spatial weight matrix identifies very strong spatial associations over relatively shorter distances (indicated by the yellow lines). We note that many strong spatial associations appear to be clustered around the four largest cities in the country (see Figure 3.3) however there are many strong spatial associations spread across other parts of the country as well. In particular, we note strong associations in the North-Western region of the country as well as some spatial associations that span across Lesotho (a neighbouring country that is landlocked by South Africa, shown in Figure 3.5(d)).

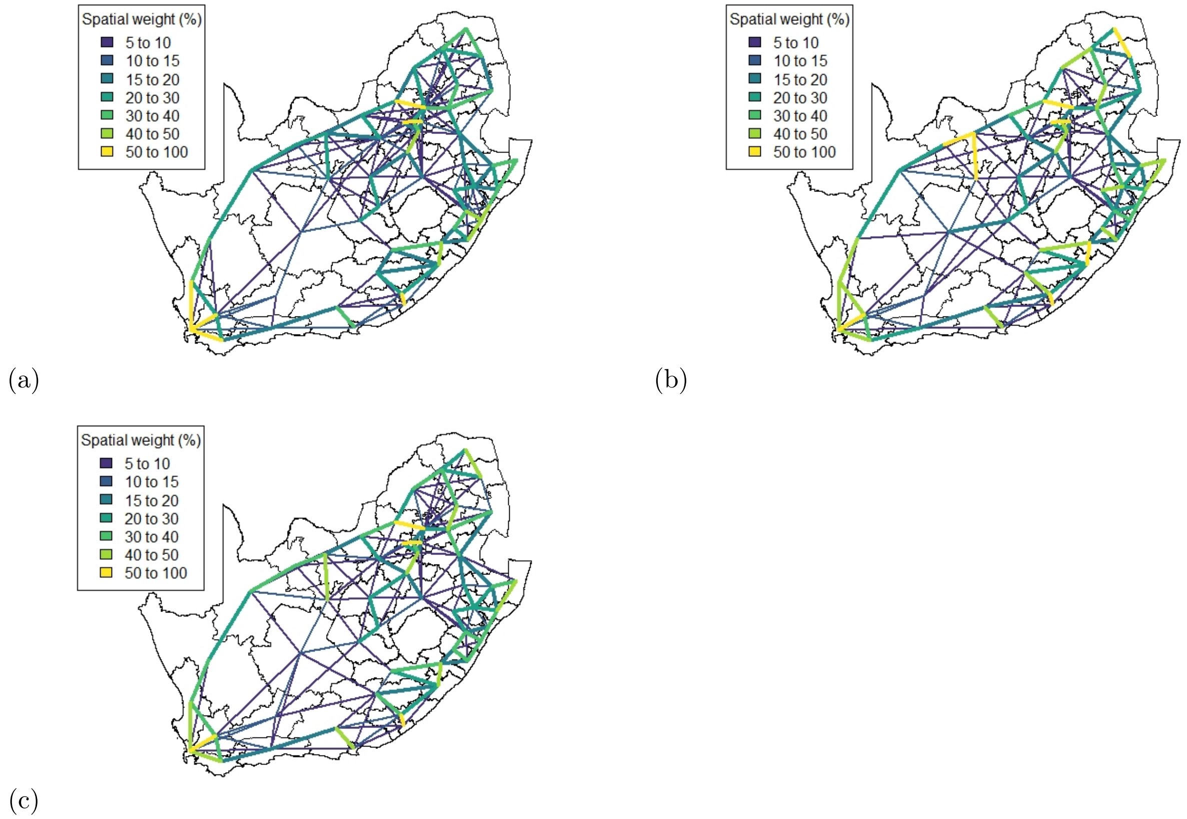


(a) (b)

(c) (d)

Figure 3.5: Method 2 spatial weight matrix entries (weights ≤ 5% not shown) (a) Business as usual, (b) Level 5, (c) Level 4, and (d) South Africa at Administrative level 3 (neighboring country Lesotho in green)

The spatial weight matrices for the mobile network data were also aggregated to administrative level 2[1], shown in Figure 3.6, in order to be comparable with the other candidate methodologies. While some strong spatial associations can still be identified around the country's borders, many previously identified associations (including several significant associations previously spanning across the neighbouring country of Lesotho) are now negligible. It is clear that while this lower spatial resolution does capture some of the spatial associations present in the data, much information is lost when aggregating between spatial resolutions.



(a)

(b)

(c)

Figure 3.6: Method 2 spatial weight matrix entries (weights $\leq 5\%$ not shown) (a) Business as usual, (b) Level 5, (c) Level 4, and (d) South Africa at Administrative level 2

3.2.3 Method 3 - Weighted Facebook data method

In order to create a spatial mobility matrix using the Facebook data, we employ the same approach as Ejigu et al. [40]. Similar to Equation (3.17), the entries of this matrix take into account proximity as well as spatial region covariate information. The entries of the spatial weight matrix are given by

$$w_{ij}^{(t)} = \exp\left(-\left(\alpha \cdot \left|F_i^{(t)} - F_j^{(t)}\right| + (1-\alpha) \cdot d_{ij}\right)\right) \tag{3.20}$$

[1] Please see Section 2.2.3 for a full discussion on how the data was aggregated.

where $F_i^{(t)}$ is the mobility of district municipality i at time t (indicated by the Facebook data discussed in Section 2.2.2) scaled by population size, d_{ij} is the Euclidean distance between the centroids of district municipality i and j, and $\alpha \in (0, 1)$ is a control parameter indicating the amount of weight that should be given to the covariate term [40].

In [100] the authors set α to 0.6. This value was selected in order to allow the covariate data to play a slightly more prominent role in the estimation process without disregarding the importance of distance. The parameter incorporates the fact that we are making an assumption that the Facebook data can be used to capture transitions between regions even though it is isolated region data. The value of 0.6 gives the weighted calculation a slight nudge towards the Facebook data. Note that if $\alpha = 0$ then this matrix simplifies to that of Method 1.

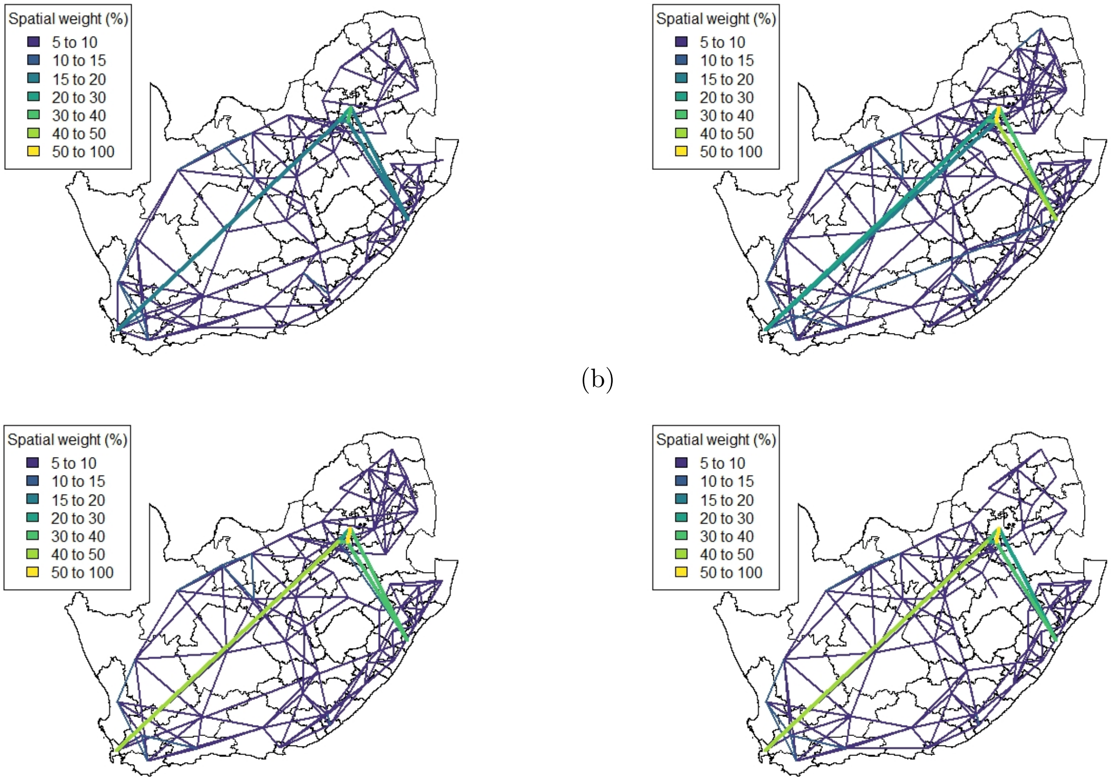
In order to preserve user privacy, the Facebook mobility data was highly censored and aggregated (see Chapter 2). This resulted in the data exhibiting relatively little variation over much of its time span (see Figure 2.6). In order to induce a greater amount of variation in the data, the mobility measures were scaled by district municipality population size. This also serves to account for the fact that increased mobility in a given district municipality should have a greater (lesser) affect on neighboring wards if the population size in that district municipality is large (small).

Figure 3.7 shows the resulting matrix for each considered level of lockdown. By incorporating both mobility and population size into this matrix, the strong spatial association between the four largest cities in South Africa is brought to light, despite the large geographical distance between them. If only Euclidean distance had been taken into account, this association would not have been identified, as was the case with Method 1.

3.2.4 Method 4 - Scaled Facebook data method

An additional spatial weight matrix was constructed based on further variation of the exponential distance model. For this matrix, the rows of the exponential distance matrix are scaled using the (unscaled) Facebook mobility data. For example, if for a given day t,the mobility within district municipality i was 20% lower than the baseline, i.e. $F_i^{(t)} = 0.8$, then the entire row i is multiplied by 0.8. Each entry in the exponential distance matrix is thus scaled by some number within the interval (0,2). The entries in the matrix are given by

$$w_{ij}^{(t)} = \left(1 + F_i^{(t)}\right) \cdot \exp(-d_{ij}). \tag{3.21}$$



(a)

(b)

(c)

(d)

Figure 3.7: Method 3 spatial weight matrix entries (weights $\leq 5\%$ not shown) (a) Business as usual, (b) Level 5, (c) Level 4, and (d) Level 3

This construction allows the exponential distance matrix to be scaled such that the spatial influence of more (less) mobile district municipalities is increased (decreased). This also renders the exponential distance matrix non-symmetric, which should offer a more realistic representation of spatial influence. Methods 3 and 4 are novel approaches to constructing spatial weight matrices using the Facebook mobility data.

This spatial weight matrix was constructed as a potentially more realistic alternative to the exponential distance matrix. Despite containing a temporal element (in the form of daily mobility measurements retrieved from the Facebook data), the results for this matrix do not show any significant change across the various levels of lockdown. Figure 3.8 shows the elements of the spatial weight matrix.

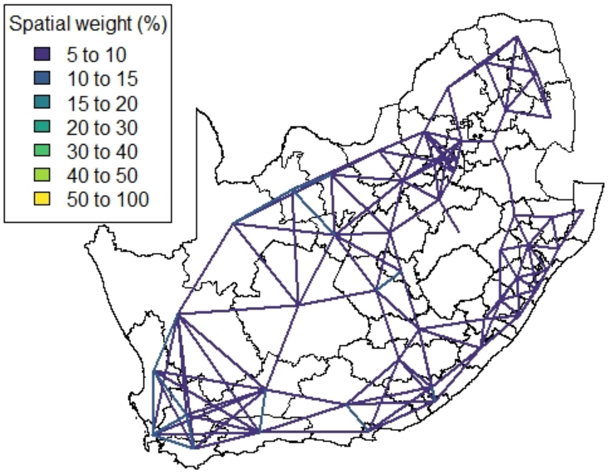


Figure 3.8: Method 4 spatial weights (weights $\leq 5\%$ not shown)

3.3 Discussion

We now discuss the various interpretations that can be derived from the spatial weight matrices discussed within this section. Method 1 employed the exponential distance model which is often used in econometric modelling where a spatial component is included. In Figure 3.4 we note that there are no significantly large spatial associations present when utilizing this method. Instead, all spatial associations are very weak and relatively equal. This model is clearly too simple to serve as a representation of spatial association when modelling a phenomenon as intricate as the spread of COVID-19.

Method 2 was perhaps the most simplistic in terms of conceptualization, since the entries of the spatial weight matrix are simply the proportion of individuals who transitioned between particular pairs of locations. Despite this, it offers one of the most interesting representations of spatial associations for locations within the study region. This is almost certainly due to the mobile network data being of a particularly high quality and spatial resolution. In Figures 3.5 and 3.6 we note that this method introduces a much larger amount of variation in the spatial weight matrix entries compared to Method 1.

Recall that this data was originally obtained at ward level (administrative level 4). When aggregated to local municipality level (i.e. administrative level 3) as shown in Figure 3.5 we note that there are many strong spatial associations (indicated in yellow) clustered around the previously indicated focal points of activity (see Figure 3.3) as well as a few selected locations spread out across the country. However when aggregated to district municipality level (i.e. administrative level 2) as shown in Figure 3.6 we note that many of the previously identified strong spatial associations are no longer present. This illustrates that while the lower spatial resolution is necessary for computational reasons as well as to facilitate the comparison of the different methods, it results in the loss of a great deal of information. Furthermore, while this data is of exceptionally high quality, the high spatial resolution proved to be somewhat preventative to analysis due to great computational costs. Converting between the various levels of spatial resolution took a very long time (upwards of a month) to complete.

While the strong spatial associations observed around the four major cities are quite intuitive, the other associations spread across the country are worth additional discussion. The relatively mild associations that criss-cross over Lesotho in Figure 3.5 as well as the strong associations across parts of the country could potentially be attributed to migrant workers travelling for work. This seems reasonable given the size of South Africa's mining industry and the fact that many individuals need to travel for work. The fact that these spatial associations appear to be stronger during level 5 lockdown could perhaps be attributed to the fact that while other individuals remain at home these individuals still travel to earn a living and thus make up a larger proportion of the travel in and out of district municipalities. Regardless of the level of lockdown such individuals need to travel for work and are thus potential means for the disease to spread that require further monitoring.

Method 3 is based on a somewhat recent paper and so is somewhat more "innovative" than the other candidate models. It allows us to strike a balance between the importance of distance and other region attributes that could affect the phenomenon being studied. When studying Figure 3.7 we find that including the Facebook mobility data (as well as population size) as auxiliary information allows us to create spatial weight matrices that identify strong spatial associations between the four focal points of activity in the country despite these locations being relatively far apart in geographical space (see Figure

3.3). Given that these locations would intuitively play a larger role in the spread of an infectious disease such as COVID-19 it stands to reason that this matrix's inclusion is warranted.

We acknowledge that Ejigu et al. [40] stated that their spatial weight matrix construction did not indicate the degree to which spatial regions influenced one another but rather how similar they are. However that was within the context of soil samples and how their proximity to a local river could be correlated to similar measurements of minerals. Within our context we are considering the effect of human mobility between different regions. We thus believe that it is not unrealistic to utilize this spatial weight matrix as a proxy for spatial influence since it is only logical that cities with larger populations and similar mobility patterns will naturally have a greater effect on one another. This is especially true given that the identified regions are hubs of human activity and thus will impact one another greatly. This should hold true even during a pandemic (if not more so).

Method 4 was proposed as a means of benefiting from the simplistic nature of Method 1 but expanding on the representativeness of the resulting matrix. It was hoped that by incorporating the Facebook mobility data we could drive the spatial associations away from the approximately uniform distribution that we observed in Figure 3.4. By incorporating the Facebook data it was also believed that the spatial weight matrix would exhibit some form of temporal variation to express changes in spatial association over time. Unfortunately this was not the case, instead of introducing stronger spatial associations such as those seen with Methods 2 and 3, this method resulted in some spatial associations being reduced and others only marginally increasing. This resulted in spatial associations either being very weak (as with the majority of the spatial associations) or only moderately strong (indicated by the blue lines we see in Figure 3.8). When comparing the results for Methods 3 and 4 (Figures 3.7 and 3.8 respectively) we note that the latter resembles the former but with the strong spatial associations between the four largest cities removed. We thus speculate that the great deal of censoring that the Facebook underwent has some part to play in the rather lackluster results for Method 4. Method 3 had the benefit of artificially induced variation due to the inclusion of population sizes, but this was not possible for Method 4 given its functional form (see Equation (3.21)). While the importance of user privacy is abundantly clear, we feel that it is evident from these results that the level of censorship that is applied to publicly available data is somewhat restrictive in practice.

The various methodologies employed to create these spatial weight matrices were conceptually simple but resulted in representations of spatial association that were vastly different. For three of the chosen methodologies (specifically Methods 1, 3 and 4) the distance between locations played a pivotal role in the estimation process. Despite this similarity in construction, the implied spatial associations of these methods are vastly different. Method 1 is based purely on the distance between locations and as a result all spatial associations are relatively equal (and thus very weak). Method 3 incorporated Facebook mobility data as well as population size and as a result is able to identify strong spatial associations between the four largest cities in the country over long distances. Lastly, method 4 is a somewhat “smoothed” version of Method 1 but ultimately fails to offer any new insights or temporal variation in estimation. Method 2 delivered perhaps the most interesting and diverse representation of spatial association but this was most probably only due to the high quality of the data used in its construction.

When considering these candidate models for the epidemiological modelling of COVID-19 the choice of which methodology to use will depend on the scale and scope of the model. If one aims to model the spread of COVID-19 over the entire country then a combination of both Methods 2 and 3 would be ideal, given that their strengths are somewhat complimentary. However if modelling the spread of the virus over a smaller region (such as a single province) then Method 2 would be the ideal choice given that it is based on high quality mobile network data and requires no assumption regarding the mobility of the population to be made. If the user wishes to ensure that the role of geographic distance is considered (which can be ideal over shorter distances) then the inclusion of either Method 1 or 4 can also be justified.

3.4 Conclusion

In this chapter we presented and discussed the results of various constructions for spatial weight matrices designed to represent the spatial association between different regions. We started with a review and discussion of methods that are most often employed in literature before introducing and discussing the methods evaluated for use in this mini-dissertation. Given the discussion above we have decided to utilize both the spatial weight matrices for Method 2 and Method 3 in the modelling that will form the focus of the discussion in the next chapter. These two methods were chosen for their complementary representations of spatial association as well as their increased significant results compared to the other considered methods.

In this mini-dissertation we found it necessary to aggregate the spatial resolution of certain data sources down in order to facilitate the comparison of the various candidate methodologies. This resulted in the loss of a great deal of data. Future research could be focused on developing some methodology that can be used to *increase* the spatial resolution of data sources, such as small area estimation or micro-simulation (see e.g. [11, 96]).

Chapter 4

Compartmental disease models

A topic that has seen a significant increase in relevance and popularity in recent times due to the outbreak of the COVID-19 pandemic is the use of epidemiological models to aid in the understanding and prediction of the way in which the disease spreads. Such models aid policymakers in deciding on control measures to help prevent the spread of the disease in order to save lives and effectively allocate medical resources. In order to understand the spread of the disease we need to understand the way in which it affects those it infects.

The natural progression that the infected experience can be broken up into several stages. Initially all individuals are susceptible to infection. It is a known fact that there is no possibility of immunity without prior exposure and recovery for diseases such as COVID-19. Upon exposure to the virus an individual undergoes a period where they can be confirmed to be infected but do not display symptoms or the ability transmit the disease to other hosts. This time period is known as the latent or incubation period[1]. The latent period for COVID-19 is estimated to be 5 days on average[2], however some estimates state that it could be as few as 2 days. The onset of symptoms commences either after or shortly before the end of the latent period. At this time the infected individual becomes capable of transmitting the virus to other susceptible individuals. The infected individual is then classified as *infectious.* After some time, the infected individual either recovers from the infection or succumbs to their symptoms and passes away. The recovery time for COVID-19 is estimated at 10 days, but has been observed to vary greatly by case severity[3].

[1] World Health Organization: https://1e.to/1xdmEP (Accessed September 2021)

[2] World Health Organization: https://1e.to/CJlDBu (Accessed September 2021)

[3] Discovery: https://1e.to/2p971X(Accessed September 2021)

When modelling a pandemic we are thus actually interested in the number of individuals who are experiencing the various stages of infection at any particular point in time. We thus want to estimate the number of individuals that are: susceptible to infection, exposed to the disease, infectious and able to spread the disease, recovered and deceased. By modelling these numbers we can estimate quantities such as the recovery and mortality rate and predict the time and span of peaks in infections.

Compartmental disease models enable us to model these desired quantities. First proposed by various researchers throughout the early 1900's [108, 109, 86, 62, 12] these models operate by dividing the population being studied into different compartments describing the different stages associated with the disease. When these models were first proposed they consisted of the following compartments:

- Susceptible (S): Those individuals that have yet to be infected by the disease being studied but are capable of being infected.
- Infectious (I): Those individuals who have been exposed to the disease and are capable of infecting other individuals through physical contact or close proximity.
- Removed (R): Individuals that have been removed from the system, either through recovery and the development of immunity or death.

These models are appropriately referred to as *Susceptible-Infectious-Removed* models (or simply *SIR* models). Over time these models have been adapted and expanded in many ways in order to model particular disease features or answer desired modelling questions. These included the introduction of a wide array of new components to the base model such as

- Dead (D): Those individuals that die as a result of the disease. This allows modelers to explicitly distinguish between cases that result in recovery and those that result in death and allow for the estimation of a mortality rate [72, 93].
- Exposed (E): Individuals that have been exposed to a disease and can be confirmed as *infected* but are not yet *infectious* [24, 25, 72, 98, 99]. This compartment is used in order to take into consideration the latent period exhibited by diseases.
- Diagnosed and hospitalized (often denoted by J): Individuals that have been confirmed as being infected and have been admitted to hospital care. These individuals are often assumed either less likely or incapable of infecting other individuals due to isolation [32, 33, 34].
- Asymptomatic (A): Individuals who are infected but do not display any symptoms of the disease.

Often such individuals are assumed to have a lower probability of infecting other individuals [32, 34].

Any combination of these various compartments as well as new ones defined specifically for a given case study is, in theory, possible. By the naming convention of the *SIR* model it should be clear that these types of models are usually named after the compartments they contain in the order of flow from one state to another. For example, *SEIJRD* indicates a model where individuals are initially susceptible to infection, hence in the "Susceptible (S)" compartment. Upon becoming infected individuals proceed through the compartments "Exposed (E)" and "Infectious (I)" before either proceeding to "Recovered (R)", "Diagnosed and hospitalized (J)" (which will also flow into the recovered compartment) or "Dead (D)" [33]. The compartments that will be used for a particular model will depend on the modelling objective and the features of the particular disease being studied.

When utilizing such models we have two variants at our disposal, one of which is largely deterministic and another which is stochastic in nature. We now discuss both these alternatives in turn, however greater emphasis will be placed on the latter.

4.1 Deterministic compartmental models

Perhaps the more frequently utilized version of compartmental epidemiological models are those models that are deterministic in nature. This is most likely due to the fact that the stochastic alternative requires making several assumptions including the distribution of a disease's parameters, which requires information that is not always available in practice [25].

Consider the 'SIR' model as initially defined by Kermack and McKendrick [62]. Assume that the disease is transmitted at a constant rate of β. This value is known as the *transmission rate* and expresses the rate at which infected individuals interact with susceptible individuals [33, 63]. Given this value, we calculate the rate at which individuals become infected as the product of the transmission rate and the proportion of the population that are infected, thus yielding $\frac{\beta I}{N}$. We then also assume that infected individuals recover at a constant rate of γ. In Figure 4.1 is depicted a state transition diagram representing the process of such a model.

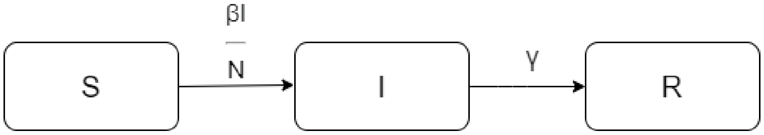


Figure 4.1: Diagram of standard SIR model

The core methodology used to estimate the model parameters requires solving a series of (non-linear) ordinary differential equations. Using the compartments and state transition in Figure 4.1 as an example, the system can be described via the following system of ordinary differential equations [62]:

$$\begin{aligned}\frac{dS(t)}{dt} &= -\frac{\beta \cdot S(t) \cdot I(t)}{N} \\ \frac{dI(t)}{dt} &= \frac{\beta \cdot S(t) \cdot I(t)}{N} - \gamma I(t) \\ \frac{dR(t)}{dt} &= \gamma \cdot I(t),\end{aligned}$$

where N is the total size of the population under study and $S(t), I(t), R(t)$ indicate the number of individuals in each compartment at time t. This model is built upon the following assumptions:

- N remains constant, i.e. the model does not account for births or deaths.
- All individuals in the population are susceptible to infection, i.e. the model does not account for forms of natural-born immunity or differing susceptibility.
- There is no latent period. This is evident by the lack of an “Exposed” compartment. The model therefore does not distinguish between “infected” and “infectious” and individuals are assumed capable of infecting others immediately upon exposure to the disease.
- Once an individual recovers or dies from the disease under investigation they cannot become infected again for the remaining study time.

These assumptions are rather restrictive and in many cases can be found to be unrealistic for application to real-world data. There exists several calibrations that can be made to this model to make it more realistic. Births and deaths (from reasons other than the disease) can be incorporated [32]. The susceptibility of individuals can also be allowed to vary [33] and through the use of an ‘SIRS’ model we can also account for the possibility of individuals becoming infected more than once (hence being able to transition back to the “Susceptible” compartment upon recovering) [55]. Furthermore infection and recovery rates can also be allowed to vary with time, which is more often the case than not when using stochastic compartmental models [24, 55, 98].

These models can naturally be extended as discussed before to include a larger number of compartments. If we wish to also model the number of individuals exposed but not yet infectious as well as distinguish between deaths and recoveries we can utilize a model such as the one depicted by Figure 4.2. The inclusion of the “Exposed” compartment is a popular inclusion to allow for the modeling of the latent period [24, 98]. This diagram is similar to the one we had for the system depicted in Figure 4.1, except now we have included an additional rate, σ, that expresses the rate at which exposed individuals become infectious. We also now split the rate of transition out of the infectious compartment using what is known as the *case fatality ratio* [63], assuming that $100\alpha\%$ of infectious individuals recover while the remaining $100(1-\alpha)\%$ pass away due to the disease being studied. This allows for the modelling of the mortality rate of a pandemic.

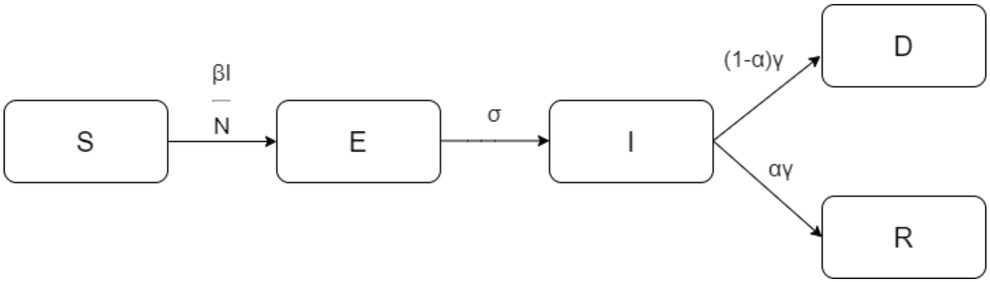


Figure 4.2: Diagram of SEIRD model

When using this compartmental model we have a different set of differential equations than before, given by Korolev in [63] as

$$\begin{aligned}
\frac{dS(t)}{dt} &= -\frac{\beta \cdot S(t) \cdot I(t)}{N} \\
\frac{dE(t)}{dt} &= \frac{\beta \cdot S(t) \cdot I(t)}{N} - \sigma \cdot E(t) \\
\frac{dI(t)}{dt} &= \sigma \cdot E(t) - \gamma \cdot I(t) \\
\frac{dR(t)}{dt} &= (1-\alpha) \cdot \gamma \cdot I(t) \\
\frac{dD(t)}{dt} &= \alpha \cdot \gamma \cdot I(t).
\end{aligned}$$

These models are deterministic due to the fact that they only rely on solving a set of differential equations at each time step. Variability can be artificially introduced into these models by estimating the model parameter numerous times for different (random) initial values for the parameters that need to be estimated [32, 63]. Alternatively one could utilize parametric bootstrap to derive confidence intervals for the model parameters [34, 105].

4.2 The basic reproductive number

Before discussing the stochastic analogue of compartmental models we introduce a term that is almost synonymous with these types of models, namely the *basic reproductive number*, most commonly denoted as R_0 [24]. The basic reproductive number is defined as the expected number of secondary infections produced by a single infected individual in a population consisting entirely of susceptible individuals [24]. For example, if $R_0 = 2$ then we expect every infectious individual to infect 2 more susceptible individuals during their infectious period. The basic reproductive number quantifies the transmission potential of a disease. This term is often confused with the *effective* reproductive number, which is similar to the former but for a population with some form of public intervention i.e. where not all individuals are equally susceptible to infection [24, 32, 72]. The effective reproductive number is less than or equal to the basic reproductive number since any non-pharmaceutical intervention is assumed to decrease the susceptibility of a portion of the population.

For a basic 'SIR' model as depicted in Figure 4.1 we have it that the basic reproductive number is given by the product of the average transmission rate and the average infectious period, i.e. 'number of people infected per time spent infectious' [24, 32, 34, 63]. Earlier we denoted the transmission rate by β. Furthermore, the average infectious period is the inverse of the rate at which individuals leave the "Infectious" compartment, i.e. $\frac{1}{\gamma}$ according to our former notation We therefore have that for this model,

$$R_0 = \frac{\beta}{\gamma}. \tag{4.1}$$

Note that for the 'SEIRD' model as defined earlier the expression for the basic reproductive number is the same due to the fact that the transmission rate is unchanged and the total rate of flow out of the "Infectious" compartment is still is just γ. If however we consider a slightly more complicated model, such as the one used by Chowell et al. [33] to model the outbreak of SARS in various countries between 2002-2003 [33] we arrive at a slightly more involved expression for R_0. This particular model uses two susceptible compartments to distinguish between individuals who are more (less) susceptible to infection as well as some of the previously defined compartments such as "Exposed (E)", "Hospitalized (J)" and "Dead (D)"[4]. This model is illustrated in Figure 4.3.

For a full discussion of the rates of transition between the different compartments as shown in Figure 4.3 refer to [33]. Note that unlike before the rate of flow both in and out of the infected compartment cannot be easily expressed by single parameters. Rather, by the use of what is referred to as the second generator

[4] Note: The compartment "Comparison (C)" was included only to compare the model estimates with officially published records.

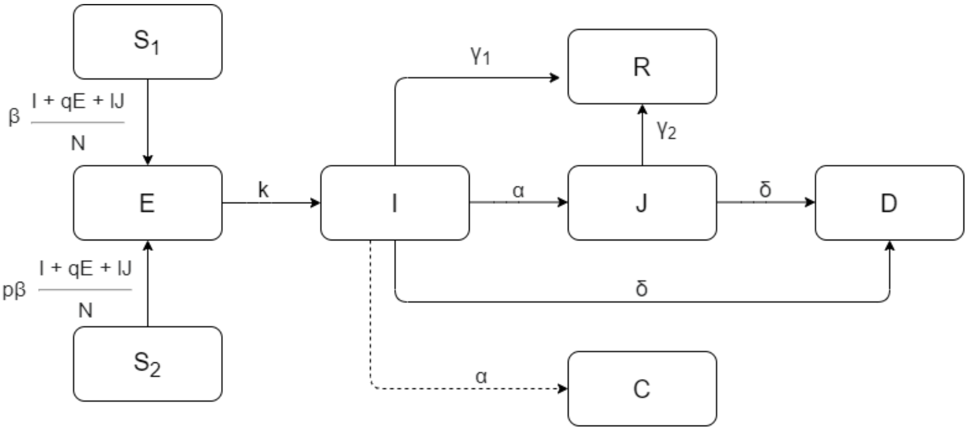


Figure 4.3: Diagram of a more complex deterministic model

approach (see [33]) it can be shown that the expression for R_0 is given as

$$R_0 = \{\beta\,[\rho + p(1-p)]\} \cdot \left\{\frac{q}{k} + \frac{1}{\alpha + \gamma_1 + \delta} + \frac{\alpha \cdot l}{(\alpha + \gamma_1 + \delta)(\gamma_2 + \delta)}\right\}. \tag{4.2}$$

This example serves to illustrate that while the definition for the basic reproductive number is rather straight forward, the expression for it will depend on the compartments included in the model as well as the interaction between them.

The basic reproductive number quantifies the power of an infectious disease with regards to how quickly and potentially uncontrollably it can spread [32]. A useful property of this quantity is its threshold values. If $R_0 \leq 1$ then we can expect a disease to die out, since each infectious individual is either infecting 1 or no individuals. However if $R_0 > 1$ then the disease will continue to spread and the pandemic will grow since every infected individual is infecting *at least* one more person [47, 72]. Naturally this value is also restricted to be strictly positive since the rate of transmission instinctively must be strictly positive as is the rate of recovery.

When attempting to investigate the variability of R_0 when using a deterministic model we can once again induce variability in our results by estimating the R_0 numerous times for different initial values [32, 63]. Alternatively one could utilize parametric bootstrap to derive confidence intervals [34, 105]. The basic reproductive number is an essential result of compartmental models and serves as a valuable summary of the effects of an infectious disease. We will revisit this quantity with respect to stochastic compartmental models in a later section.

4.3 Stochastic compartmental models

Thus far we have restricted our discussion of compartmental epidemiological models to those models that were fully deterministic in nature. While these models are quite popular and can be relatively simple to understand, we can achieve a greater degree of variability in our results by employing models that are inherently stochastic. The random nature of these models allow for greater analysis of pandemic features as well as a more realistic representation of the real-world mechanics of the disease.

Unlike their deterministic counterparts, which are all very similar in terms of construction, stochastic models allow for a greater amount of flexibility and freedom in terms of how they are set up. Consequently the literature has numerous examples of models that are all quite distinct from one another. This is because these models are often derived intuitively using statistical concepts and intuition rather than a set formula that can be modified slightly.

In this mini-dissertation we will focus on one such model. We specifically discuss a model developed by Porter et al. [98, 99] with later modifications being made by Brown et al. [24, 25]. Once again consider the SEIR model with transitions illustrated by Figure 4.4. Recall that the number of individuals within each compartment at time t is given by $S(t)$, $E(t)$, $I(t)$, and $R(t)$ respectively, which represent the number of individuals who are susceptible, exposed, infectious and removed from the system [72].

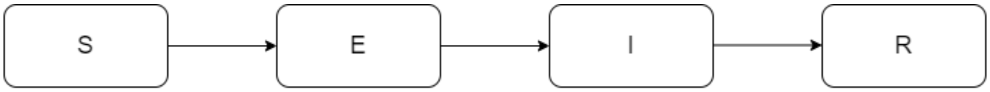


Figure 4.4: Diagrams of transitions in SEIR model

Now let, $E^*(t)$, $I^*(t)$ and $R^*(t)$ represent the number of individuals that transition into the "Exposed (E)", "Infectious (I)" and "Removed (R)" compartments at time t respectively [72]. It is then true that for a time interval $(t, t+h]$ we have the following:

$$\begin{aligned}
S(t+h) &= S(t) - E^*(t) \\
E(t+h) &= E(t) + E^*(t) - I^*(t) \\
I(t+h) &= I(t) + I^*(t) - P^*(t) \\
R(t+h) &= R(t) + R^*(t) \\
N(t) &= S(t) + E(t) + I(t) + R(t),
\end{aligned}$$

where $N(t)$ is the total size of the population which we once again assume remains constant [72]. These equations express that the number of individuals within each compartment at the end of a time interval is equal to the number of individuals at the end of the previous time interval, plus the number of individuals who transitioned into the compartment, minus the number of individuals that transitioned out of the compartment. The interval between successive observations is denoted here by h and is known as the temporal offset [72]. In most studies this is simply taken to be 1 day [24, 25, 72].

We can model the event of an individual transitioning from one compartment to another using a Bernoulli distribution. Therefore to model the event of a large number of independent individuals transitioning between states we utilize a binomial distribution as follows:

$$\begin{aligned} E^*(t) &\sim Bin\left(S(t), \pi^{(SE)}\right) \\ I^*(t) &\sim Bin\left(E(t), \pi^{(EI)}\right) \\ R^*(t) &\sim Bin\left(I(t), \pi^{(IR)}\right) \end{aligned}$$

with the respective probabilities given by

$$\begin{aligned} \pi^{(SE)} &= 1 - \exp\left[\frac{-\beta(t)}{N} \cdot I(t) \cdot h\right] \\ \pi^{(EI)} &= 1 - \exp(-\gamma_{EI} \cdot h) \\ \pi^{(IR)} &= 1 - \exp(-\gamma_{IR} \cdot h). \end{aligned}$$

It is clear to see how these equations relate to those of the deterministic version previously discussed. Just as before $\beta(t)$ is the (now time-dependent) transmission rate [72]. Here we have it that $1/\gamma_{EI}$ and $1/\gamma_{IR}$ are the mean latent and infectious times, therefore the rate of transition from “Exposed (E)” to “Infectious (I)” and from “Infectious (I)” to “Removed (R)” is given by γ_{EI} and γ_{IR} respectively [72].

By inspecting the functional forms of the probabilities above we can deduce that the model assumes that the transition time between compartments follows an exponential distribution with rate parameter identical to those in the deterministic model (times the temporal offset which is usually set to 1). The exponential distribution is often an unrealistic representation of the actual dynamics of a pandemic, primarily due to the memory-less property implying that the time spent in a compartment does not effect the probability of transitioning out of that compartment. This is in contrast to reality, where individuals experiencing symptoms become more likely to either recover or die as time passes. Despite this, the exponential distribution is often chosen for ease of computation [72, 98].

In fact, when using this model and assuming that the transmission rate remains constant over time, i.e. $\beta(t) = \beta \, \forall \, t$, the basic reproductive number is given by the exact same expression as before, i.e. that given by (4.1). This is due to the fact that, despite now using a different model structure, the transmission and recovery rate remain unchanged. However this would not be the case if we allow our transmission rate to vary as a function of time to incorporate the effects of, say, public health interventions. For example, we could have the transmission rate be given by the following

$$\beta(t) = \begin{cases} \beta, & \text{for } t < t_* \\ \beta \cdot e^{-q(t-t_*)}, & \text{for } t \geq t_*. \end{cases}$$

Here t_* is chosen as the date since the commencement of the pandemic that some preventative intervention was brought into effect that (it is assumed) caused the transmission rate to decline with an exponential rate of decay [72]. Using this functional form we see that the transmission rate will decay at an exponential rate of $q > 0$ as time passes past the commencement of the intervention. Recall that in the presence of public health interventions we instead have an *effective* reproductive number, since we assume that the entire population is not completely susceptible. Our effective reproductive number is now given by

$$R_0(t) = \begin{cases} \frac{\beta}{\gamma_{IR}} & \text{for } t < t_* \\ \frac{\beta \cdot e^{-q(t-t_*)}}{\gamma_{IR}} & \text{for } t > t_*. \end{cases}$$

Note that the effective reproduction number (indicated simply as $R_0(t)$ due its time-varying nature) is thus less than or equal to the basic reproduction number (indicated by R_0). We can thus clearly see the effect that the public intervention has on the reproduction number when incorporated into the transmission rate in the model. For any reasonably defined intervention function we should observe that $R_0(t) \leq R_0$ for all t. This example serves to show that while these systems can appear to vary significantly from their deterministic counterparts, they utilize similar logic with regards to how they conceptualize and model the spread of a disease.

As previously mentioned, the assumption of exponentially distributed latent and infectious time periods is not always appropriate, primarily due to the memory-less property of the exponential distribution [98]. This is due to the fact that it implies that the probability of an individual transitioning between compartments remains constant over time regardless of how long an individual has been in a particular state, naturally this is highly unrealistic. Ideally we would like to relax this assumption in order to allow for the use of general distributions based on prior information regarding the latent and infectious times [98]. Due to the fact that, by their very nature, the spread of infectious diseases occurs spatially, models that take account of spatial dependencies can be extremely informative. It would thus also be ideal to develop a modelling framework that allows for the inclusion of spatial components. To achieve both these desired properties, we discuss the *spatial* SEIR model [25], initially referred to as the *path-specific* SEIR model by Porter & Oleson [98] before being adapted into its current form by Brown et al. [24, 25].

We establish the following notation. Let $t_i : i = 1, 2, \ldots, T$ be the index representing the amount of time that has passed since the initial outbreak of the pandemic, with T being the maximum length of the investigation. Next, assume we have n distinct spatial locations, each indicated by $s_j : j = 1, 2, \ldots, n$. Then, let l be the index representing the amount of time an individual has spent in a particular state [25, 98]. Lastly, assume that the maximum amount of time an individual can occupy the latent and infectious state is M_1 and M_2 respectively [25, 98]. We then define the 3-dimensional arrays $\boldsymbol{E}$ and $\boldsymbol{I}$, with dimension $T \times n \times M_1$ and $T \times n \times M_2$ respectively. Each cell (i, j, l) represents a point in time since the onset of the pandemic, a spatial region and an amount of time spent in either the latent or infectious state. The actual number in the cell then indicates the number of individuals for whom this is true.

Finally, we make the following assumptions [98]:

1. Individuals who become exposed to the pathogen are guaranteed to become infectious and cannot transition back to being susceptible.
2. Individuals who are in the infectious compartment are constantly infectious (their ability to infect susceptible individuals does not vary over time).
3. The event of an individual transitioning from the exposed to the infectious compartment and an individual transitioning from the infectious to the removed compartment are independent.
4. All individuals have the same latent and infectious time distributions.

The core mechanics of the *path-specific* SEIR model are then given by (please refer to the original manuscript of [24] for the full derivation):

$$\boldsymbol{I}^*_{ij} \sim \sum_{l=1}^{m_1} Bin\left(\boldsymbol{E}_{ijl}, P\left(Z_1 \leq l + h | Z_1 > l\right)\right) \tag{4.3}$$

$$\boldsymbol{R}^*_{ij} \sim \sum_{l=1}^{m_2} Bin\left(\boldsymbol{I}_{ijl}, P\left(Z_2 \leq l + h | Z_2 > l\right)\right). \tag{4.4}$$

Recall that $E^*(t)$, $I^*(t)$ and $R^*(t)$ were previously used to indicate the number of individuals that transition into the exposed, infectious and removed compartments. We maintain this notation, defining the matrices $\boldsymbol{S}^*$, $\boldsymbol{E}^*$, $\boldsymbol{I}^*$ and $\boldsymbol{R}^*$, with identical interpretation and some added properties. $\boldsymbol{I}^*_{ijl}$ indicates the number of individuals that transition into the infectious compartment t_i time units after the initial outbreak of the pandemic, in spatial region s_j after spending l time units in the exposed compartment. $\boldsymbol{I}^*_{ij}$ is then the total number of such individuals, summed across the third dimension (i.e. disregarding how much time they spent in the previous compartment). The definition of $\boldsymbol{R}^*_{ij}$ follows in the exact same way.

These distributions explain that the number of individuals that transition from exposed to infectious (or infectious to removed) after spending an amount of time l in the former compartment is distributed binomial with first parameter given by the number of such individuals and probability given by the general distribution we have specified for the latent (or infectious) time period (once again please refer to the original manuscript of [24] for the full derivation).

As explained in [98], the individuals that do not transition between states are handled via diagonalization. Therefore, we define the following

$$X_{ijl} \sim Bin\left(\boldsymbol{E}_{ijl}, P\left(Z_1 \leq l+h | Z_1 > l\right)\right)$$
$$Y_{ijl} \sim Bin\left(\boldsymbol{I}_{ijl}, P\left(Z_2 \leq l+h | Z_2 > l\right)\right).$$

We then have the following

$$\boldsymbol{E}_{i+1,j,l+1} = \boldsymbol{E}_{ijl} - X_{ijl}$$
$$\boldsymbol{I}_{i+1,j,l+1} = \boldsymbol{I}_{ijl} - Y_{ijl}.$$

Essentially, those individuals that do not transition at a particular point in time are merely passed diagonally down the first and third dimension of our 3-dimensional arrays (recall that the second dimension represents the region and thus remains unchanged). The core improvement this model has over previous models lies in the fact that it allows general distributions to be specified for the latent and infectious time distributions, this is given by the random variables Z_1 and Z_2 respectively [98]. It now remains to define how individuals transition from the susceptible to the exposed state. We will utilize the improvements made to this model by Brown et al. [24, 25].

Firstly, we assume that the intensity process (which determines the exposure probability at any given time) for spatial region s_j at time t_i is given by θ_{ij} [24]. It is not possible to identify every $\theta_{ij} \,\forall\, s_j, t_i$ and so instead we incorporate these values into a linear predictor. The $\{\theta_{ij}\}$ values are arranged into a $T \times n$ matrix. We then assume that the intensity process for all spatial regions is determined by a set of P independent covariates (and is determine in the same way). We indicate the parameter values for these covariates as $\boldsymbol{\beta}$ [25]. These covariates could be region attributes such as population health susceptibility, government restrictions, weather data or simply a function that varies over time in some pre-selected manner.

Let $\boldsymbol{X_j} : T \times P$ be the design matrix for spatial region s_j where each row represents a different point in time and each column a different covariate, then the intensity process for spatial region s_j is an $T \times 1$ column given by

$$\boldsymbol{\theta_j} = \boldsymbol{X_j}\boldsymbol{\beta}. \tag{4.5}$$

By appending the design matrix for every spatial region row-wise, we can derive the matrix $\boldsymbol{X} : TN \times P$. The expression $\boldsymbol{\theta} = \boldsymbol{X}\boldsymbol{\beta}$ simply evaluates to a $Tn \times 1$ vector, where every set of T consecutive values indicates the intensity process of a different spatial region [24].

Furthermore, assume the following with regards to contact between individuals [24].

- The probability of a susceptible individual becoming infected upon contact with an infectious individual is given by p.
- The number of epidemiological significant contacts between individuals within spatial unit s_j at time t_i is distributed $Poi(\lambda_{ij})$.
- The intensity process depends on the previous two (unidentifiable) quantities through a log-link function, specifically $\theta_{ij} = \log(\lambda_{ij} p)$.
- When an individual travels from one spatial region to another, their contact behaviour is accurately modeled by the contact distribution for that spatial region.
- There exists some known function f that describes the spatial correlation between spatial locations and accepts as an argument some distance metric d_{j_1,j_2} for spatial locations s_{j_1} and s_{j_2}.

The last assumption requires some measure of distance or "closeness" between spatial locations. To fulfill this role, we readily rely on the previously defined concept of a spatial weight matrix to serve as this measure. In order to increase flexibility we can also choose to use more than a single spatial weight matrix if we wish to account for varying forms of spatial dependence [24]. Define the set $\{\boldsymbol{D}_z : z = 1, 2, \ldots, Z\}$ as a collection of Z different spatial weight matrices (alternatively referred to as 'spatial distance matrices' hence the chosen notation) [24]. The probability of a new infection at time t_i in spatial region s_j is then given by,

$$\pi_{ij}^{(SE)} = 1 - \exp\left[\left\{ - \eta_{ij} - \sum_{z=1}^{Z} \rho_z \left(\boldsymbol{D_z} \eta_{ik}\right) \right\}_{k \neq j}^{h_i}\right], \tag{4.6}$$

where $\eta_{ij} = \delta_{ij} e^{\theta_{ij}}$ is the exposure process of spatial region s_j at time t_i and δ_{ij} is the proportion of inhabitants of spatial region s_j that are infected at time t_i [24]. The exposure process, which is a function of the intensity process, serves as the parameter for the exposure probability.

As stated previously, the interpretation of this functional form is that the probability of exposure to COVID-19 within spatial region s_j at time t_i, $\pi_{ij}^{(SE)}$, follows an exponential distribution. The parameter of this exponential distribution is determined by the exposure process within spatial region s_j, η_{ij}, as well as a proportion of the exposure process within other spatial locations $\eta_{ik}, k \neq j$. The proportion of the exposure process that is "shared" by other spatial locations is indicated by the spatial autocorrelation parameter ρ_z and is determined by the spatial weight matrix $\boldsymbol{D}_z$ (note that the functional form also allows for multiple expressions of spatial dependence, i.e. spatial weight matrices, to be included). Finally, the term h_i is a temporal offset, indicating the frequency of observations.

It is also possible to extend this model to include the possibility for individuals to become susceptible once again after they've recovered from infection [25], however due to the relatively short study period we will not investigate this further in this mini-dissertation.

4.4 The empirically adjusted reproductive number

We now resume our discussion on the reproductive number. Recall that the basic reproductive number, most commonly indicated as R_0, is a single measure that quantifies the transmission potential of a disease. It is informally defined as the expected number of secondary infections per infectious individual [24]. As was seen earlier in Figure 4.3, the calculation of R_0 can become quite involved as it is often dependent on the parametric formulation of the particular model used. When first developing the spatial SEIR model, Brown et al. [24] devised a measure that is similar to R_0, having similar threshold and interpretation but with some notable differences. This quantity is known as the empirically adjusted reproductive number and will be denoted as $R^{(EA)}(t)$ [24]. The key differences between these two quantities of interest are outlined in Table 4.1.

The first difference between these measures is that while R_0 is time-invariant and represents the expected number of secondary infections over the entire study period, $R^{(EA)}(t)$ varies as a function of time by incorporating the time-variant intensity process discussed earlier. Secondly, while the basic reproductive number assumes an entirely susceptible population, the empirically adjusted reproductive number considers that the risk of exposure to the disease varies over time and thus susceptibility is not constant. As was shown earlier (Section 4.2), the expression for the basic reproductive number is heavily dependent on the parametric form of the model. In contrast, the empirically adjusted reproductive number is determined by directly inspecting the sizes of the various compartments that make up the model being used. Granted the simple parametric form of the basic reproductive number may be preferred to the infinite summation involved when utilizing the empirically adjusted version.

Table 4.1: Differences between R_0 and $R^{(EA)}(t)$

	Basic reproductive number	**Empirically adjusted reproductive number**
1	A single value for the entire study period. Measure is time-invariant.	Varies over time as the intensity process changes over the study period.
2	Assumes an entirely susceptible population.	Considers the time-varying intensity process that drives the spread of the disease.
3	Dependent on parametric form of model.	Calculated using compartment population sizes.
4	Closed parametric form for most models.	Involves infinite summation with sum realistically tending to zero.

The empirically adjusted reproductive number commences by calculating the expected number of secondary infected individuals per infectious individual [24]. Let the indicator variable $I_k(t_i, s_j, s_l)$ indicate the event that a susceptible individual k originally from spatial region s_j becomes infected at time t_i due to epidemiological contact established in spatial region s_l. The expected number of time such infections will occur is given by

$$E\left[\sum_{k=0}^{N_{i,j}} \left(I_k(t_i, s_j, s_l)\right)\right] = S_{ij} \cdot P\left(I_k(t_i, s_j, s_l | k \in S)\right) \tag{4.7}$$

where the summation is taken over the population size of spatial region s_j at time t_i, indicated by $N_{i,j}$. The average number of such infections per infectious individual is then simply (4.7) divided by the number of infectious individuals in the same spatial unit and at the same time, therefore

$$\frac{S_{ij} \cdot P\left(I_k(t_i, s_j, s_l | k \in S)\right)}{I_{i,j}}. \tag{4.8}$$

Recall that we can incorporate more than one form of spatial dependency through the use of multiple spatial weight matrices [24]. If we opt to only use one spatial weight matrix then it holds that

$$P\left(I_k(t_i, s_j, s_l | k \in S)\right) = 1 - \exp\left(-f(d_{jl}) \cdot \eta_{il}\right) \tag{4.9}$$

where d is a chosen measure of distance between spatial locations s_j and s_l and f is the function used to determine the spatial weight matrix value for these two spatial locations. If we use more than one spatial weight matrix this equation is replaced by

$$P\left(I_k(t_i, s_j, s_l | k \in S)\right) = 1 - \exp\left(-\sum_{z=1}^{Z} \rho_z \{\boldsymbol{D}_z\}_{jl} \cdot \eta_{il}\right) \tag{4.10}$$

where once again $\eta_{il} = \delta_{il} e^{\theta_{il}}$. Note that the absence of the first term in the exponent seen in Equation (4.6) is due to the fact that we assume $j \neq l$ (i.e. we are calculating the probability for two distinct spatial units). For the case where $j = l$ we merely have

$$P\left(I_k(t_i, s_j, s_l | k \in S)\right) = 1 - \exp\left(-\eta_{il}\right) \tag{4.11}$$

since the distance metric is not relevant in this case. A full derivation of these expressions can be found in the original doctoral thesis by Brown [24].

The values calculated using Equation (4.8) for all spatial locations s_j and s_k at a time t_i can be arranged into a matrix $\boldsymbol{G}(t_i) : n \times n$ which serves as an analogue of a 'next generation matrix' [24]. The concept of a next generation matrix was first employed by Allen and van den Dreissche to calculate the basic reproductive number [24]. The important feature of these matrices is that each row sum of a matrix calculated for a particular point in time t_i indicates the average number of infections caused by individuals in a particular spatial region [24].

We can therefore calculate the average number of secondary infections per infectious individual per spatial region at any single point in time. We can then generalize this to the entire study period by taking the sum of all these matrices, weighted by the probability that an individual who becomes infectious at time t_i will still be infectious at every point in time in the remainder of the study period [24]. We therefore have the following equation

$$R^{(EA)}(t_i) = \sum_{t=t_i}^{\infty} \boldsymbol{G}(t) \cdot \left[\prod_{k=t_i}^{t} \left(1 - \pi_k^{(IR)}\right)\right]. \tag{4.12}$$

Naturally the weighting sum will quickly approach zero, since the probability of an individual remaining infectious over a long period of time will quickly approach zero for most active pathogens (including COVID-19) [24].

The empirically adjusted reproductive number is far more flexible than the basic reproductive number. This is due to the fact that it can be readily applied to any stochastic discrete-time compartmental model and is not dependent on the parametric form of the model. It can also be shown that the basic reproductive number is a special case of the empirically adjusted reproductive number, provided some assumptions are satisfied (see [24] for proof).

4.5 Conclusion

In this chapter we discussed the history and theory of compartmental epidemiological models. After an overview of their initial conceptualization and development over the last century, we discussed their various possible constructions and the assumptions made in order to model the spread of infectious diseases. It is evident that there is a large degree of freedom that can be achieved when using these models, with it being possible to modify existing models to suit any particular disease under study.

We devoted some special consideration to the reproductive number, a quantity that expresses the expected number of secondary infections of a disease. This quantity is highly important to epidemiological research and policymakers as it expresses the potential of a disease to spread. The empirically adjusted variant appears to have several attractive features that render it potentially more useful and will be utilized later in this mini-dissertation.

It is clear from the discussion in this chapter that compartmental models can easily become quite complex in terms of fitting and evaluating candidate models (despite being relatively straight forward to interpret). This could potentially represent a problem with regards to the fitting and evaluation of such models and perhaps could affect stochastic models more adversely due to the uncertainty attributed to model parameters. In the following chapter we discuss a method for effectively fitting such potentially complex stochastic models.

Chapter 5

Approximate Bayesian Computation

In this chapter we review the theoretical background relating specifically to the techniques that will be employed to fit and analyze the spatial epidemiological model(s) used in this mini-dissertation. We start with a review of traditional Bayesian inference before proceeding to discuss a family of approximate Bayesian computation methods that are particularly useful in the field of epidemiology. For the analysis performed in this mini-dissertation, the approximate Bayesian computation (ABC) rejection algorithm (Section 5.1) and the sequential Monte Carlo ABC algorithm (Section 5.3) will be the most relevant topics of discussion in this chapter. We also discuss the history and limitations of approximate Bayesian techniques to fully conceptualize these techniques.

The world of statistical inference is primarily divided into two schools of thought, those being Bayesian and frequentist [135]. The former considers probability as a numeric expression of the beliefs or uncertainty of events. In contrast, the latter considers probability as simply a limiting frequency of an event [135]. While the frequentist approach can be considered more rigid and objective, the Bayesian approach allows for more freedom with regards to the information that can be incorporated in the estimation process [135, 81]. The core principle of Bayesian statistics is that parameters of interest are random variables rather than fixed but unknown quantities [81]. This is particularly useful within the context of stochastic compartmental models, since all model parameters are assumed to be random variables in order to express the uncertainty present in the model [81].

In addition to allowing for uncertainty in model parameters, the Bayesian framework also allows for the incorporation of prior information or beliefs regarding the model parameters [81]. Given that it is often the case that there exists a wealth of literature that can offer researchers some idea of parameter distribution, being able to take such information into account is very desirable [25]. In the case of COVID-19 for example, there has thus far been a wealth of research done on the virus in a relatively short amount of time (e.g. [1, 97]) that can offer prior information for key parameters.

We briefly recap the basis of Bayesian inference for the sake of completeness before proceeding to discuss the methods that are of particular interest in this mini-dissertation. For further information, please refer to [88, 135]. We commence by deriving the core theorem required for Bayesian inference, Bayes' theorem [135]. Consider two events A and B and assume that $P(B) > 0$, by applying the law of conditional probability twice in immediate succession we then have the following

$$\begin{aligned} P(A|B) &= \frac{P(A \cap B)}{P(B)} \\ &= \frac{P(B|A) \cdot P(A)}{P(B)}. \end{aligned} \tag{5.1}$$

Equation 5.1 is known as Bayes' theorem [135]. Bayes' theorem allows us to use observed information to update our prior beliefs to improve our parameter estimates. For example, suppose we have the prior belief that some parameter θ has a distribution given by $\pi(\theta)$. Suppose we then observe data, X, which is dependent on θ through the likelihood function $f(X|\theta)$ and has an unconditional probability $f(X)$. We can use this observed data, in conjunction with Bayes' theorem, to update our beliefs about the distribution of θ. Using Equation 5.1 we thus have

$$f(\theta|X) = \frac{f(X|\theta) \cdot \pi(\theta)}{f(X)}.$$

A slight simplification occurs when noting that the denominator in this expression is simply a normalizing constant [131, 77]. We can thus simplify this expression as follows

$$f(\theta|X) \propto f(X|\theta) \cdot \pi(\theta). \tag{5.2}$$

The term $f(\theta|X)$ is referred to as the *posterior distribution* of θ given X as it describes our new beliefs about θ after having observed X. This result can be used in a multitude of ways to perform inference on the posterior distribution.

Approximate Bayesian computation (ABC) is a method that can be used to simulate values from a target posterior distribution without the need to evaluate the likelihood function [101, 130, 15]. This is of particular interest in cases involving highly complex systems or data with high dimensionality where the likelihood function, $f(X|\theta)$, can often be found to either be wholly intractable or computationally expensive to evaluate [130, 124, 83, 25]. This issue is especially common in the field of genetics [60, 21, 14, 76, 107].

5.1 ABC rejection sampler

In the early 1980's Rubin [110] expressed that users of Bayesian inference ought not be limited to cases where the likelihood is strictly tractable. He consequently conceptualized an algorithm through which samples could be generated from a desired posterior distribution without needing the likelihood function to be evaluated. Over the course of the next two decades numerous attempts were made at formulating an algorithm that could achieve this goal. In 1997 Tavaré et al. [127] first proposed a rejection algorithm whereby a summary statistic was calculated for the available data. Samples for the parameter of interest were then drawn from the prior distribution, after which they were accepted with probability proportional to that of the calculated summary statistic value given the proposed parameter value. Following this, numerous advancements were proposed by various researchers. Fu and Li [46] extended the idea by simulating an artificial dataset using the proposed parameter value and comparing the summary statistic value derived for the original and artificial datasets. Weiss and von Haesler [136] then extended this idea to allow for numerous summary statistics and allowed proposed values of the parameter of interest to be accepted with some level of tolerance. Specifically, proposed values for the parameter of interest were accepted if $||\boldsymbol{s}' - \boldsymbol{s}|| \leq \epsilon$ for a chosen distance metric $||\cdot||$ and tolerance level $\epsilon \geq 0$ where $\boldsymbol{s}'$ and $\boldsymbol{s}$ are the chosen summary statistic(s) calculated for the simulated and observed dataset respectively [17]. Finally, in 1999 Pritchard et al. [101] formalized what is now commonly referred to as the ABC rejection algorithm [101]. The full ABC rejection algorithm, as presented by Toni et al. [130], is given in Algorithm 1.

Algorithm 1 The ABC rejection algorithm

```
$N$ = number of desired samples from the posterior distribution.
$\epsilon$ = tolerance level.
$\rho$ = distance metric.
Initialize $i = 1$.
while $i < N$ do
    Simulate $\theta^*$ from $\pi(\theta)$.
    Simulate a dataset $\hat{X}$ from $f(X|\theta^*)$.
    if $\rho\left(X, \hat{X}\right) \leq \epsilon$ then
        Accept $\theta^*$.
        $i = i + 1$.
    else
        Reject $\theta^*$.
```

The intuition behind the rejection algorithm is rather straight forward. We start by sampling a proposed value for θ from its prior distribution. We then use this value in our statistical model to simulate an artificial dataset $\hat{X}$. We then compare this simulated dataset to the observed data, X. If the simulated data is sufficiently similar to the observed data then we accept the proposed value for θ. We determine the similarity between the two datasets using a distance metric ρ and a positive tolerance level ϵ. We repeat this process as many times as required to obtain the desired sample size. Note how Algorithm 1 does not require the likelihood function to be explicitly evaluated, however it does require that the user be able to simulate from it.

In the original conceptualization by Rubin the tolerance level ϵ would be set to zero [110]. By using a tolerance of zero it restricts the accepted parameter values to those that produced artificial datasets that matched the actual data exactly [110, 124, 25]. However, a tolerance of zero also causes a larger proportion of proposed parameter values to be rejected and thus increases computational cost [126, 117, 25]. When using a non-zero tolerance however we observe that we do not end up simulating from the true posterior distribution [16, 107, 124]. Recall that we wish to derive the posterior distribution of the parameter of interest θ given by $f\left(\theta|X\right)$ which (if using Equation (5.2)) requires the evaluation of the likelihood $f\left(X|\theta\right)$. However due to the intractability or computational infeasibility of the likelihood, we instead evaluate $f_\theta\left(\rho\left(X,\hat{X}\right)\leq\epsilon\right)$ i.e., the distribution of parameter values that produce simulated datasets $(\hat{X})$ that sufficiently approximates the observed data (X) for some selected distance metric $\rho\left(\cdot,\cdot\right)$ and tolerance level ϵ [35, 124]. We are thus not evaluating Equation (5.2) but rather

$$f\left(\theta\middle|\rho\left(X,\hat{X}\right)\leq\epsilon\right)\propto f_\theta\left(\rho\left(X,\hat{X}\right)\leq\epsilon\right)\cdot\pi(\theta). \tag{5.3}$$

It has been shown that as $\epsilon\to 0$, Equation (5.3) does approximate the true posterior distribution [130, 107, 124, 15]. The inference performed using this method is thus not exact but rather an approximation and hence we have the naming convention referring to the method as an approximate computation method [25]. There are many model parameters, including the tolerance level and distance metric, which play a role in the quality of this approximation which will be discussed later in this chapter. Most often users will not directly compare the simulated and actual datasets due to the acceptance rate being very low when comparing high dimensional data [130]. Instead, the comparison is made between a selected summary statistic(s), S, calculated for both the observed and simulated data [17, 77, 130, 35]. We therefore replace the calculation within the "if" statement in line 8 of Algorithm 1 with $\rho\left(S\left(X\right),S\left(\hat{X}\right)\right)\leq\epsilon$. The selection of appropriate summary statistics is a long-standing open problem in the field of approximate Bayesian computation and will also be discussed later in this chapter.

Numerous adjustments have been proposed to improve the efficiency and accuracy of the rejection algorithm. One such adjustment is regression-adjustments [17, 117, 15]. Beaumont et al. [17] proposed to improve the computational and statistical efficiency of the algorithm by performing local linear regression to weaken the effect of discrepancies between the actual and simulated summary statistics. They also proposed smoothing the proposed parameter values for the linear regression step as well as when estimating the posterior distribution [17]. Blum and Francois proposed a similar solution whereby a one-layer feed-forward neural network (FFNN) was utilized to perform non-linear regression on proposed parameter values [21]. While adjustments of this nature have proven to increase the overall computational efficiency and accuracy of the basic ABC rejection algorithm [17, 21, 15] it has also been shown that such adjustments can potentially produce more misleading and biased results when the observations are not well explained by the model [15].

The ABC rejection algorithm can be highly inefficient when the prior distribution is diffuse with respect to the posterior distribution [120, 81, 130, 25, 15]. Chief among the many algorithms proposed to overcome this drawback are two that take the basic methodology of the ABC rejection algorithm and supplement it with elements of Metropolis-Hastings MCMC [77] and Sequential Monte Carlo routines [120, 16, 130], respectively. We now discuss the development and characteristics of these two algorithms.

5.2 Markov Chain Monte Carlo ABC

Approximate Bayesian computation is focused on posterior inference that circumvents the need to evaluate the likelihood function. In a similar fashion, Markov Chain Monte Carlo (MCMC) methods can be utilized to study a desired posterior distribution without the need of a full expression for the posterior [131, 77]. MCMC allows us to study a distribution by directly sampling observations from it using the prior distribution and likelihood function and using a Markov chain, thus circumventing the requirement for a closed form expression [77].

One of the most commonly used forms of MCMC is known as the Metropolis-Hastings algorithm [131, 77]. The full algorithm, known as Metropolis-Hastings, is given in Algorithm 2.

The proposal distribution q serves to add noise to new proposal values (this prevents us from repeatedly sampling the exact same observations) [131]. In the original formulation of the algorithm by Metropolis this distribution was restricted to being symmetrical [5]. Hastings later generalized the result to non-symmetrical proposal distributions [5]. If q is symmetrical then the calculation of α at line 6 in Algorithm 2 simplifies to comparing the posterior densities for the two proposed values. It has been shown that

Algorithm 2 The Metropolis-Hastings algorithm

```
Initialize $\theta_0$ as some feasible starting value.
$q$ = proposal distribution.
$N$ = number of desired samples from posterior.
for $i = 1$ to $N$ do
    Sample $\theta_i$ from $q(\theta_i|\theta_{i-1})$.
    Compute $\alpha = \min\left(1, \frac{q(\theta_{i-1}|\theta_i)\cdot f(\theta_i|X)}{q(\theta_i|\theta_{i-1})\cdot f(\theta_{i-1}|X)}\right)$.
    Accept $\theta_i = \theta_i$ with probability $\alpha$, otherwise set $\theta_i = \theta_{i-1}$.
```

under suitable conditions the stationary and limiting distribution of the chain is the desired posterior distribution [77].

The intuition behind MCMC is rather straight forward. We start at an initial value for the parameter of interest. We then propose a new parameter value and either accept this new proposal if it results in a higher posterior probability or randomly decide whether to accept it if it results in a lower posterior probability. The reason for the latter decision is to allow the process to occasionally transition to lower density regions of the parameter space. Other versions of MCMC exist that are more suited to deal with correlated parameters, such as Gibbs sampling or difference evolution MCMC [131].

In 2003, Marjoram et al. [77] proposed an algorithm through which samples could be drawn from a desired posterior distribution without evaluating the likelihood function by incorporating elements of the basic ABC rejection algorithm as well as the Metropolis-Hastings MCMC algorithm. This was deemed a necessary improvement over the ABC rejection algorithm since sampling from the prior when using the latter could lead to parameter values from low probability regions being accepted [76]. Subsequently known as the Monte Carlo Markov Chain ABC (MCMC ABC) algorithm, the full procedure is given in Algorithm 3 [77].

Algorithm 3 The Markov Chain Monte Carlo ABC algorithm

```
$N$ = number of desired samples from the posterior distribution.
$\epsilon$ = tolerance level.
$\rho$ = distance metric.
$q$ = proposal distribution.
Initialize $i = 1$.
Initialize $\theta_0$.
for $i = 1$ for $N$ do
    Propose a value for $\theta^*$ using a proposal distribution $q(\theta|\theta_{i-1})$.
    Simulate a dataset $\hat{X}$ from $f(X|\theta^*)$.
    if $\rho\left(X, \hat{X}\right) \leq \epsilon$ then
        Set $\theta_i = \theta^*$ with probability $\alpha = \min\left(1, \frac{\pi(\theta^*)q(\theta_i|\theta^*)}{\pi(\theta_{i-1})q(\theta^*|\theta_{i-1})}\right)$
        and $\theta_i = \theta_{i-1}$ with probability $1 - \alpha$.
    else
        Set $\theta_i = \theta_{i-1}$
```

Note how Algorithm 3 differs from Algorithm 1 in that we now simulate proposed values for θ^* using a proposal distribution q. Furthermore, if the artificial dataset simulated using θ^* is adequately similar to the observed dataset (as once again determined through the use of some distance metric ρ) then it is not necessarily accepted. Rather, the proposed parameter value is accepted with some probability α, calculated in line 11 of Algorithm 3 in the same way as for the Metropolis-Hastings algorithm.

It has been shown that the chain defined in Algorithm 3 is guaranteed to have a limiting and stationary distribution equal to the true posterior distribution [77, 130]. By embedding the Metropolis-Hastings MCMC procedure within the ABC framework we derive several benefits. This algorithm displays a higher degree of efficiency when compared to the ABC rejection algorithm due to acceptance rates being far higher [77, 120, 130]. In cases where the proposal distribution is symmetric (i.e., $q(\theta_i|\theta^*) = q(\theta^*|\theta_i)$), α is only dependent on the prior distribution [130]. Finally, MCMC ABC can also be used with flat improper priors [15]. There are however many potential drawbacks to using MCMC ABC. These drawbacks include poor mixing in the tails of the posterior distribution (see e.g. [22, 103] for proposed solutions), correlated samples and the possibility that low tolerance levels can result in the chain getting ‘stuck’ in regions of low probability for long periods of time [120, 130].

5.3 Sequential Monte Carlo ABC

Some of the drawbacks of MCMC ABC include correlated samples as well as the Markov chain occupying low probability spaces for long periods of time [120, 130]. These drawbacks can be rectified using Sequential Monte Carlo (SMC) methods [120, 130, 81]. SMC methods operate by sampling an initial set of parameter values, $\left\{\theta_{t-1}^{(i)}\right\}$, from the prior distribution [39]. The next set of parameter values is then redrawn from this initial sample [39]. Up to some number of populations T, each set of N parameter values is drawn from the set of parameter values obtained in the previous step (using a proposal distribution q) [39]. The sampling probabilities at each step are determined by a set of weights $w_i^{(t)}$ which are either assumed equal (for initial sample) or determined as the normalized prior probability of each parameter value. The normalization of the weights is achieved through the use of a perturbation kernel, K_t. An example of a SMC algorithm (adapted from [39]) is shown in Algorithm 4.

First proposed by Sisson et al. [120], the Sequential Monte Carlo ABC (SMC ABC) algorithm improves several of the drawbacks experienced with the MCMC ABC algorithm [120, 130, 76, 83, 15]. Algorithms falling into this class of ABC can be grouped together into two categories that are typically used [76, 15]. The first category can be considered a population Monte Carlo variation of ABC [120, 130, 16]

Algorithm 4 A sequential Monte Carlo algorithm

```
1:  $N$ = number of desired samples from the posterior distribution.
2:  $T$ = number of parameter populations to simulate.
3:  $q$ = proposal distribution.
4:  $K_t$ = perturbation kernel.
5:  Initialize the population indicator $t = 1$.
6:  Initialize the particle indicator $i = 1$.
7:
8:  if t = 1 then
9:      Sample $\theta_i^{(1)}$ independently from $\pi(\theta)$.
10: else
11:     Sample $\theta_i^*$ from the previous population of accepted parameter values $\left\{\theta_{t-1}^{(i)}\right\}$ with weights $w_{t-1}$.
12:     Generate $\theta_i^{(t)}$ from $q\left(\theta_i^{(t)}|\theta_i^*\right)$.
13:
14: Calculate the weight for particle $\theta_t^{(i)}$,
15: if t = 1 then
16:     $w_i^{(t)} = \frac{1}{N}$
17: else
18:     $w_i^{(t)} = \frac{\pi\left(\theta_i^{(t)}\right)}{\sum_{j=1}^{N} w_j^{(t-1)} K_t\left(\tau_t^{-1}\left(\theta_i^{(t)} - \theta_j^{(t-1)}\right)\right)}$
19:
20: if $i < N$ then
21:     Set $i = i + 1$.
22:     Return to step 8.
23:
24: if $t < T$ then
25:     Set $t = t + 1$.
26:     Return to step 6.
```

while the second category is more akin to particle-fitting algorithms [38]. In this mini-dissertation we restrict our focus to the former class as these are more generally applied in literature and to the field of epidemiological modelling in particular [83, 25]. This class of algorithms has been developed independently and simultaneously by numerous researchers [81] and thus there are multiple versions of the procedure with only slight variations. In Algorithm 5 we present the relatively simple implementation of SMC ABC presented by Beaumont et al. [16] but acknowledge that there exist slight differences in implementation in the available literature (see e.g. [120, 130]).

In the SMC ABC algorithm, an initial simulated set of parameter values $\left\{\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(N)}\right\}$ (referred to as "particles" [120, 130, 16]) is sampled from the prior distribution. Future proposals for parameter values are then re-sampled from these parameter values and assigned weights at each iteration of the procedure [120, 130, 16, 76]. The weights are incorporated as a means of importance sampling and to ensure convergence of the accepted parameter values [16]. Note that Beaumont et al. chose to utilize the normal distribution kernel, $\mathcal{N}$, as the proposal distribution [16]. As before, a perturbation kernel, K_t, is

Algorithm 5 The sequential Monte Carlo ABC algorithm

```
1: $N$ = number of desired samples from the posterior distribution.
2: $T$ = number of parameter populations to simulate.
3: $\{\epsilon_1, \epsilon_2, \ldots, \epsilon_T\}$ a decreasing sequence of tolerance levels.
4: $\rho$ = distance metric.
5: $K_t$ = perturbation kernel.
6: Initialize the population indicator $t = 1$.
7: Initialize the particle indicator $i = 1$.
8:
9: if t = 1 then
10:     Sample $\theta_i^{(1)}$ independently from $\pi(\theta)$.
11: else
12:     Sample $\theta_i^*$ from the previous population of accepted parameter values $\left\{\theta_{t-1}^{(i)}\right\}$ with weights $w_{t-1}$.
13:     Sample $\theta_i^{(t)}|\theta_i^* \sim \mathcal{N}(\theta_i^*, \tau_t^2)$ where $\tau_t^2$ is twice the empirical variance of $\left\{\theta_{t-1}^{(i)}\right\}$.
14:
15: Simulate a dataset $\hat{X}$ from $f(X|\theta_i^{(t)})$.
16:
17: if $\rho\left(X, \hat{X}\right) > \epsilon$ then
18:     Return to step 9.
19: else
20:     Calculate the weight for particle $\theta_t^{(i)}$,
21:     if t = 1 then
22:         $w_i^{(t)} = \frac{1}{N}$
23:     else
24:         $w_i^{(t)} = \frac{\pi\left(\theta_i^{(t)}\right)}{\sum_{j=1}^{N} w_j^{(t-1)} K_t\left(\tau_t^{-1}\left(\theta_i^{(t)} - \theta_j^{(t-1)}\right)\right)}$
25:
26: if $i < N$ then
27:     Set $i = i + 1$.
28:     Return to step 9.
29:
30: if $t < T$ then
31:     Set $t = t + 1$.
32:     Return to step 7.
```

used to normalize the weights at each iteration. The kernel can either be a pre-selected function such as the uniform or Gaussian density function [130, 16] or can be modified at each iteration (see [16] for a discussion on why this would be preferred). The multivariate Gaussian density function has been shown to be an ideal choice [16, 14]. As opposed to the SMC algorithm given in Algorithm 4, in Algorithm 5 proposed parameter values are only accepted if their resulting simulated dataset, $\hat{X}$, is sufficiently similar to the observed dataset, X. A decreasing sequence of tolerance levels is used to ensure that the accepted parameter values tend towards the desired posterior distribution [130]. The values for the tolerance ϵ can either be determined via some deterministic decreasing sequence or using quantiles from prior iterations [17, 16, 25].

The SMC ABC algorithm offers significant improvements over both the ABC rejection and MCMC ABC algorithms. It avoids the problem of becoming “stuck” in areas of low probability present in the MCMC ABC algorithm [120, 130]. Samples drawn using SMC ABC are uncorrelated as opposed to highly correlated as they are when drawn using the MCMC ABC algorithm [120, 130]. Particle values that do not serve as good samples from the posterior distribution are discarded at later iterations [120]. Due to the procedure being based on population samples and the ability to set initial tolerances to be higher it is possible to explore complex distributions (for example multimodal distributions) more efficiently [120, 119, 83]. By allowing a different sampling distribution and tolerance level at each iteration it enables us to study the affect of the choices made for each iteration individually [120]. Note that in the special case where $T = 1$, Algorithm 5 simplifies to Algorithm 1, i.e. the ABC rejection algorithm [130]. This is true since it means we only simulate a set of parameter values, simulate an artificial dataset and determine whether or not to retain the sample based on similarity with the observed data. We then merely repeat this process until we have a sample of N retained parameter values, just as is done in Algorithm 1.

5.4 Model comparison

An essential part of Bayesian inference is comparing various candidate models to determine whether there is evidence to suggest that a particular model describes the data better than the other(s) [130, 35]. Fortunately, the Bayesian inference framework allows different models to be compared using the Bayes factor [35, 124] and this value can readily be calculated for models fitted using approximate Bayesian methods [101, 130, 124, 25]. Let m_1 and m_2 be two candidate models, the Bayes factor is then defined as

$$B_{12} = \frac{P(m_1|X)/P(m_2|X)}{\pi(m_1)/\pi(m_2)} \tag{5.4}$$

where $\pi(m_i)$ and $P(m_i|X)$ are the prior and posterior distribution for model $m_i, i = 1, 2$ [130, 14, 124]. If we assume that the prior distribution of each model is uniform then this simplifies to

$$B_{12} = \frac{P(m_1|X)}{P(m_2|X)}. \tag{5.5}$$

This value can then be calculated for each combination of candidate models $m_1, m_2, \ldots, m_K$ [14, 15]. The intuition behind these expressions is that we are comparing the probability of the two candidate models being a good fit for the observed data. In other words, given the observed data, how much more likely are we to select model 1 as a better fit than model 2? In order to determine this, we determine the ratio of the marginal distributions for each model, given the observed data. If the ratio of the marginal distribution of model 1 to that of model 2 is very large, then we can conclude that model 1 provides a better fit for the observed data. There are numerous ways to interpret the values obtained for the Bayes factor. An example of this (as given in [130]) is given in Table 5.1.

Table 5.1: Interpretation of Bayes factor for model comparison

B_{12}	Evidence against m_2
1 - 3	Very weak
3 - 20	Positive
20 - 150	Strong
$>$ 150	Very strong

In order to calculate the Bayes factor, we need to derive the posterior probability of each fitted model. To derive this, we can utilize another approximate technique known as the model choice algorithm which is given in Algorithm 6 [107, 124, 15].

Algorithm 6 The ABC model choice algorithm

```
1: $N$ = number of desired samples from the posterior distribution.
2: $\epsilon$ = tolerance level.
3: $\rho$ = distance metric.
4: Initialize $i = 1$.
5: while $i < N$ do
6:     Simulate $m$ from model prior $\pi(m)$.
7:     Simulate $\theta_m$ from the prior $\pi_m(\theta)$.
8:     Simulate a dataset $\hat{X}$ from $f_m(X|\theta_m)$.
9:     if $\rho\left(X, \hat{X}\right) \leq \epsilon$ then
10:        Set $m^{(i)} = m$.
11:        Set $\theta^{(i)} = \theta_m$.
12:        Set $i = i + 1$.
13:    else
14:        Return to step 6.
```

Models that produce accepted parameter values with greater frequency can be concluded to offer a better description of the data. We can thus approximate the posterior probability of each model by its associated acceptance rate [76, 124]. Note that there are many similarities between Algorithm 6 and Algorithm 1. The model choice algorithm is simply the ABC rejection algorithm but applied to the task of model comparison instead of simulation [107, 124, 15]. It is naturally also possible to derive MCMC and SMC equivalents of Algorithm 6 [130, 14]. The output of Algorithm 6 is two vectors, one indicating the candidate model that was accepted at each iteration of the process and another housing a set of parameter values simulated from that model.

An advantage to utilizing Bayesian inference for model comparison as opposed to hypothesis testing is that candidate models need not be nested [130]. Bayesian inference also automatically penalizes models with many parameters [130, 14]. Furthermore, measures such as the Bayes factor do not test a specific hypothesis but rather quantify the amount of evidence in favour of a candidate model [130]. A criticism commonly faced by model selection in Bayesian inference (and especially by ABC) is failing to evaluate the feasibility of the entire hypothesis space [124]. Since it is not always possible to evaluate every possible model in the hypothesis space (due to factors such as computational cost) this remains an open problem in the field of Bayesian inference and one that researchers need to be aware of [124]. It is important to note that the Bayes factor should not necessarily be used to seek out a single model with the "best" descriptive power [130, 35]. Rather it is worth keeping in mind that different models can potentially describe different parts of the data equally well [130]. Rather than discard the estimates derived from models with less evidence in their favour, one might consider techniques such as parameter averaging to consider the inference from all candidate models [117, 130, 35].

5.5 Simulation parameters

In order to perform any of the ABC algorithms described in this section it is necessary to specify the following parameters:

- Distance metric, ρ
- Tolerance level, ϵ
- Summary statistic(s), S

In this section we discuss the importance and concerns associated with each of these parameters. The distance metric is used to measure the discrepancy between the observed and simulated datasets [17, 130, 15]. Despite many studies emphasizing the importance and relative difficulty of selecting an appropriate distance metric (see e.g. [126, 117, 76]), the literature is quite vague on methodologies to achieve this. In practice the Euclidean distance is the most chosen distance metric [15]. As with the choice of summary statistics, the choice of a distance metric is most often dependent on the phenomenon being investigated [126, 15]. For example, in the field of epidemiology, it has been stated that the Euclidean distance is a natural choice with several attractive features [25]. However, this is not the case in settings where data exhibits exchangeability (a property whereby the order of observations does not affect the joint distribution of random variables) since the Euclidean distance disregards this property [15].

The tolerance parameter is most often a matter of computational cost and efficiency [76]. When utilizing a non-zero tolerance all inference that is made using ABC is considered an approximation instead of an exact calculation [119, 124, 25]. It is then natural that there would be interest in using lower tolerance values. In fact, the MCMC ABC and SMC ABC algorithms were originally developed to allow for lower tolerance values and subsequently higher accuracy than could realistically be achieved with the ABC rejection algorithm [83, 15]. A particular strength of the SMC ABC algorithm is its ability to use a decreasing sequence of tolerance values to improve accuracy [120, 119, 83]. The choice of tolerance also represents a bias-variance trade-off [17, 124]. If the tolerance is set too high, then the estimated posterior distribution tends towards the prior distribution and away from the actual posterior [17, 126, 117]. If the tolerance is too low however, then the process becomes highly computationally expensive due to less proposed parameter values being accepted [126, 117]. Surprisingly, relatively few innovations have been proposed with regards to the tolerance level. Beaumont et al. [17] suggested setting the tolerance to achieve a desired acceptance rate. Ratmann et al. [103] proposed a novel approach where the tolerance level is also treated as a model parameter (and thus a random variable). Lastly, Silk et al. [119] proposed a method through which the decreasing sequence of tolerance values in the SMC ABC algorithm can be optimized.

In many cases the methods outlined in this chapter will be applied to high-dimensional datasets [60, 130, 124, 83, 25]. This renders it difficult and inefficient to utilize the full observed and simulated dataset(s) when determining whether to accept proposed parameter values due to acceptance rates being very low [130, 15]. To overcome this concern, it has become standard practice to instead compare the values for summary statistics calculated for both the observed and simulated datasets [60, 130, 81, 124, 15]. While this addresses the concern of inefficiency it also introduces additional uncertainty regarding the choice of summary statistic(s). Indeed, the selection of summary statistics is perhaps the most prevalent ongoing matter of research within the field of approximate Bayesian inference [126, 83, 15]. Because only members of the exponential family of distributions enjoy fully sufficient summary statistics [107, 124, 15], the main concern regarding the choice of summary statistic is that an unknown amount of information is lost when using insufficient summary statistics [35, 107, 76, 124]. It has been shown that when using insufficient summary statistics there is a gap present in the existing literature regarding model comparison and selection [14, 107, 124]. It is this gap in the literature that renders the debate regarding ABC-based model choices inconclusive [76].

Numerous studies have been conducted to formulate some method through which sufficient summary statistic(s) can be identified. Some studies suggest that increasing the number of summary statistics utilized in the procedure may improve the amount of information that is retained in the approximation

[35] however it has been shown that simply increasing the number of summary statistics yields diminishing returns in terms of accuracy and increases computational cost [17, 60, 35, 124]. Joyce and Marjoram [60] proposed a methodology whereby the inclusion of a summary statistic is dictated by the amount of improvement in inference brought on by its inclusion (as measured by the improvement in a defined log likelihood function). The focus of this method is then to select an optimal subset of summary statistics [60]. This gave rise to the concept of approximate sufficiency (AS) in ABC [83, 15]. However, some have expressed concerns regarding this method, specifically regarding the resulting correlation between summary statistics that is never addressed in the study [76]. Furthermore, this method requires the user to manually specify a list of candidate summary statistics and the order in which summary statistics are evaluated changes the results [60]. Fearnhead and Prangle showed that the optimal summary statistics are the true posterior means of the model parameters [43]. They propose calculating these quantities through a separate implementation of ABC to determine the true posterior means which are then used as the summary statistic in later steps of the routine [43].

Other suggested techniques include dimension reduction techniques whereby the summary statistics are projected onto a lower dimensional plane through methods such as neural networks, partial least squares regression or principal component analysis [21, 14, 35, 15]. Recently, several studies have proposed various types of neural networks that strive to automatically learn the required summary statistics (see e.g. [58, 28, 78, 2]). These techniques will be discussed further in the next chapter. Various parties have expressed that the literature requires an automatic method for selecting summary statistics since having to identify or test candidate summary statistics manually induces greater error in the resulting inference [81, 124]. Most often however the choice of summary statistic is guided by the field and phenomenon under investigation [126, 117, 81, 124].

5.6 ABC in the field of epidemiology

ABC was initially formulated for use in the field of genetics [60, 21, 14, 76, 107]. In recent years however it has seen application in various fields, including an ever-growing utilization in the field of epidemiology. To conclude our discussion of ABC we briefly discuss some of the previous applications of ABC to epidemiological modelling.

Tanaka et al. [126] implemented the MCMC ABC algorithm to fit and study a variant of a linear birth-death process for modelling the spread of tuberculosis in San Francisco. Shriner et al. [117] utilized the model comparison capabilities of the ABC framework to evaluate candidate models for the evolution of intrahost-HIV and utilized parameter averaging in their final analysis [117]. Various studies have utilized ABC in conjunction with both deterministic and stochastic SEIR models (see Chapter 4) to study the transmission of infectious diseases including Ebola, influenza and smallpox [130, 81, 104, 82, 68, 83].

One instance that is of particular interest in this mini-dissertation is the application of the SMC ABC algorithm to fit the spatial SEIR model discussed in Chapter 4. The full SMC ABC algorithm as proposed by Beaumont et al. [16] is given in Algorithm 5. In fitting the spatial SEIR model Brown et al. [25] implemented four changes to this algorithm, specifically

- A batch size $N^* \geq N$ was utilized in order to run the simulations and distance comparisons in parallel for computational efficiency.
- The first iteration was allotted a larger batch size. This allowed the authors to decrease the size of the initial tolerance.
- Rather than specifying a full vector of tolerance levels, they had tolerance levels evolve as follows: $\epsilon_{t+1} = c \cdot \epsilon_t$ where $c \in (0, 1)$. This circumvents the need for the user to manually specify the tolerance levels.
- The perturbation kernel was generalized to a multivariate Gaussian distribution.

As stated by Brown et al. [25], the ABC framework is ideally suited for compartmental models that incorporate spatial elements. This is due to the fact that, despite such models containing many unobserved quantities, only a few quantities of interest are required to simulate the data [25]. For instance, for the spatial SEIR model discussed in Section 4.3 we can partition the unknown parameters into simply

- $\boldsymbol{\theta} = \left[\boldsymbol{\beta}, \boldsymbol{\gamma_{(EI)}}, \boldsymbol{\gamma_{(IR)}}, \boldsymbol{\rho}\right]$ and
- $\boldsymbol{\zeta} = [\boldsymbol{S}, \boldsymbol{E}, \boldsymbol{I}, \boldsymbol{R}, \boldsymbol{E}^*, \boldsymbol{I}^*, \boldsymbol{R}^*]$

where $\boldsymbol{\beta}$ is the vector of covariates for the linear parameterization of the intensity process and the parameters $\boldsymbol{\gamma_{(EI)}}$ and $\boldsymbol{\gamma_{(IR)}}$ indicate the rates of transition from the "Exposed" to "Infectious" and "Infectious" to "Removed" compartments respectively. The set of spatial autocorrelation parameters is then indicated as $\boldsymbol{\rho}$ and $\boldsymbol{S}, \boldsymbol{E}, \boldsymbol{I}, \boldsymbol{R}$ indicate the number of individuals within the susceptible, exposed, infectious and removed compartments respectively at a given point in time. Finally, $\boldsymbol{E}^*, \boldsymbol{I}^*, \boldsymbol{R}^*$ represent the number of individuals transitioning into the "Exposed", "Infectious" and "Removed" compartments at a given point in time respectively.

Any information relating to the pandemic can then be simulated from the conditional distribution $P(\boldsymbol{\zeta}|\boldsymbol{\theta})$ [25]. Following the discussion of this chapter, the use of ABC methods for fitting the parameters of a stochastic compartmental epidemiological model is clearly justified. ABC is easy to implement and understand and offers a practical solution to the problem of evaluating likelihoods (i.e. models) whose computational cost would otherwise prohibit analysis. The choice of summary statistic represents a challenge for those implementing ABC algorithms.

5.7 Conclusion

In this chapter we discussed the history and evolution of approximate Bayesian computation as a method for inference involving highly complex systems with potentially intractable likelihood expressions. We discussed the initial conceptualization of these techniques and proceeded to discusses later additions to the methodology that deliver improved results. We also discussed the use of these methods to compare two candidate models.

We then discussed the importance of several of the parameters that must be decided upon when utilising this approximate framework. These include chosen tolerance levels, measures of data similarity as well as the selection of appropriate methods for summarizing data in order to improve computational efficiency. In the next chapter we once again consider summary statistic selection and evaluate the use of artificial neural networks to derive an automatic construction of summary statistics for use within the approximate Bayesian computational framework.

Chapter 6

Deep learning

In the previous chapter we discussed ABC and its numerous implementations and advancements over the last few decades. We also outlined that the accuracy and efficiency of this family of algorithms is heavily dependent on the summary statistic(s) chosen to reduce the dimensionality of the data for more efficient simulation evaluation. Recently, attempts have been made to develop a method through which these summary statistics can be constructed automatically instead of requiring candidate summary statistics to be expertly crafted. This has mostly been achieved through the application of different types of artificial neural networks (ANN) [15], which we discuss in this chapter.

6.1 Artificial neural networks

Artificial neural networks (ANN's) models have seen an increase in relevance in both academic and commercial settings in recent years due to a growing interest in the field of data science. However this family of models, or rather the ideas and principles that they are derived from, is nothing new. The history of neural networks can be traced back all the way to the 1940's [139]. The field saw what could be considered its most significant growth period in the 1960's due to the work of researchers such as Rosenblatt and Widrow et al. [19, 20]. In this mini-dissertation we will only discuss developments necessary for our intended application, for a more thorough review of the history of neural networks please see [139, 19, 20, 52]. Neural networks were initially conceived as an attempt to mimic the way in which the human brain learns to classify objects and recognize patterns (hence the naming convention of "neural" referring to the human brain) [20]. However, given the actual complexity of the human brain it is fair to say that some of the claims regarding the biological plausibility of neural networks are greatly exaggerated [20].

Consider a simple classification problem where we wish to classify observations as belonging to either class C_1 or C_2 based on the value of a discriminant value $y(\boldsymbol{x})$ which is a function of some data $\boldsymbol{x}$. The simplest choice of discriminant function is a weighted linear combination of the entries of $\boldsymbol{x}$, such as

$$y(\boldsymbol{w}, \boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0 \tag{6.1}$$

with $\boldsymbol{w}$ and w_0 being a vector of weights and a bias term respectively [19]. A simple decision rule can be utilized to classify observations based on calculated function values, for example: An observation $\boldsymbol{x}$ can be classified into class C_1 if $y(\boldsymbol{w}, \boldsymbol{x}) \geq 0$ or class C_2 if $y(\boldsymbol{w}, \boldsymbol{x}) < 0$ [19]. A graphical representation of this function is as given by Figure 6.1.

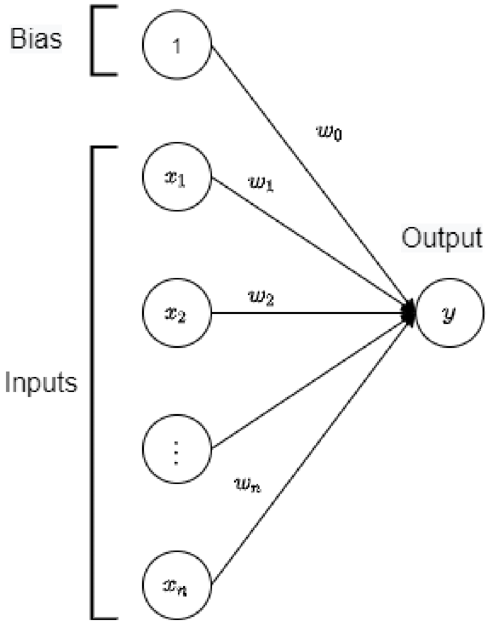


Figure 6.1: Diagram of weighted linear discriminant neural network for two classes

Even a relatively simple model (i.e. linear regression) such as this can be considered a type of neural network. A neural network with a single layer of adaptive weights such as this is known as a *single-layer perceptron* [19, 20]. It is relatively straight forward to generalize this model to more than two classes. Consider a problem where we have K classes, where $K > 2$. We can develop a weighted linear discriminant function $y_k(\boldsymbol{x})$ for each class C_k of the form

$$y_k(\boldsymbol{w}_k, \boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}. \tag{6.2}$$

An observation $\boldsymbol{x}$ can then be classified into class C_k if $y_k(\boldsymbol{x}) > y_j(\boldsymbol{x})$ for all $j \neq k$ [19]. As with the two-class case, we can represent this function graphically, as shown in Figure 6.2.

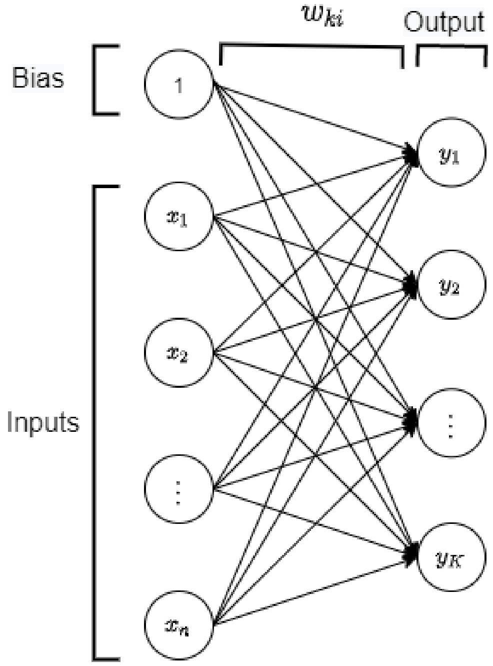


Figure 6.2: Diagram of weighted linear discriminant neural network for many classes

These functions can be generalized further by taking non-linear transformations of these linear combinations [19]. We can achieve this by using a non-linear function h, referred to as an *activation function* [19, 20, 52] as follows

$$y(\boldsymbol{w}, \boldsymbol{x}) = h\left(\boldsymbol{w}^T \boldsymbol{x} + w_0\right). \tag{6.3}$$

By selecting different activation functions we can enable the neural network to approximate a variety of continuous functions and models [19, 20]. For example, if we let h be the logistic sigmoid function, given by

$$h(a) = \frac{1}{1 + \exp(-a)} \tag{6.4}$$

then the neural network simplifies to a logistic regression model [19]. The flexibility of a single-layer perceptron can be extended even further by taking non-linear transformations of linear combinations of *functions* of $\boldsymbol{x}$ known as *basis functions* [19, 20]. Define a series of M basis functions given by $\phi_1, \ldots, \phi_M$, then a single-layer perceptron can be written as

$$y(\boldsymbol{w}, \boldsymbol{x}) = h\left(\sum_{j=1}^{M} w_j \phi_j(\boldsymbol{x}) + w_0\right). \tag{6.5}$$

We thus use the series of basis functions to transform our observations, we then assign a weight to each basis function value and pass a linear combination of these weighted values to our activation function which yields the desired output.

The goal of a neural network can be described as deriving a representation of some continuous function of the data that is too complex to achieve through simpler modelling techniques such as linear regression [19, 20]. As seen earlier, logistic regression and linear regression can both be shown to be special cases of a neural network. In order to approximate more advanced functions however it proves necessary to include more adaptive weights and doing so in a successive fashion, thus producing a *multi-layer perceptron.* Each of the examples provided thus far has been of a single-layer perceptron model, where the input and output values are separated only by a single layer (set) of adaptive weights. A neural network with two layers of adaptive weights wherein there are no feedback loops is known as a feed-forward neural network (FFNN). A graphical representation of a FFNN is shown in Figure 6.3. It has been shown that a neural network with at least two layers of adaptive weights is known as a *universal approximator*, due to them being able to approximate any continuous function [20].

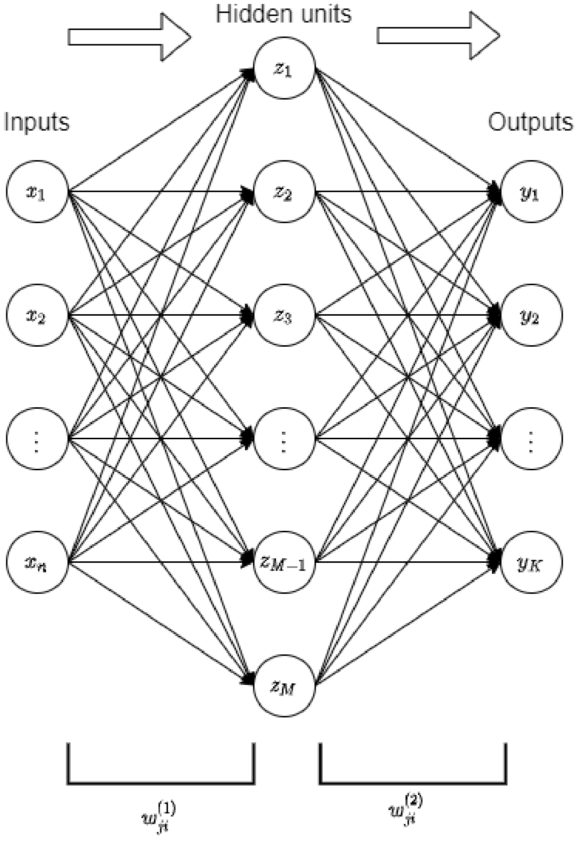


Figure 6.3: Diagram of a feed-forward neural network

In a multi-layer perceptron the output created by a layer of weights can serve as the input to the next layer. First, we determine M weighted linear combinations of the input variables $x_1, x_2, \ldots, x_n$ as follows

$$a_j = \sum_{i=1}^{n} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \tag{6.6}$$

where $j = 1, 2, \ldots, M$ and the superscript (1) indicates that the adaptive weights are found within the first layer of the network. The calculated a_j are known as *activations* [19, 20]. We then transform each activation using an activation function to produce the output

$$z_j = h(a_j). \tag{6.7}$$

These output values are not the final output of the model but rather that of the nodes between the input and output layers that are called *hidden units* [19, 20, 52]. The output of these hidden units are then used as the input values for the final output layer, as follows

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)} \tag{6.8}$$

where $k = 1, 2, \ldots, K$ indexes the outputs of the model. These activations are then once again transformed using a selected activation function to produce the final output of the model [20, 52]. The choice of activation function is dependent on the nature of the data being modelled or assumed underlying distributions [20]. For quick reference on the appropriate activation function please refer to Table 6.1.

Table 6.1: Appropriate activation functions

Task	Appropriate activation function	Definition
Regression	Identity	$h(x) = x$
Binary classification	Sigmoid or *tanh*	$h(x) = \frac{1}{1+e^{-x}}$ or $h(x) = \frac{e^{2x}-1}{e^{2x}+1}$
Multi-class problems	Softmax	$h(\boldsymbol{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{z_j}}$
Multi-layer perceptrons	Rectified Linear Unit (ReLU)	$h(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$

While there is no way to know for sure which activation function is the best fit for a given modelling task, there are some general rules of thumb. These include that the ReLU function is used in general for most neural networks, with sigmoid and *tanh* being used for recurrent neural networks[1]. Sigmoid

[1] Source: https://1e.to/xGZP5D (Accessed October 2021)

activation functions have been known to perform well in neural networks that are designed as classifiers[2]. Lastly, the softmax activation function is only used within the final output layer of the neural network while the identity function is rarely ever used and only included here as a specific case of an activation function.

We can thus write the k-th output of a multi-layer perceptron as follows

$$y_k(\boldsymbol{w}, \boldsymbol{x}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} \cdot h\left(\sum_{i=1}^{n} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \tag{6.9}$$

where $\boldsymbol{w}$ now represents the vector of all weight and bias terms in the model [20]. Note that we essentially repeated the same calculation twice, by taking a non-linear transformation of a weighted linear combination of input values, however note that the activation function need not to be the same for each layer and hence the activation function for the second layer has been indicated by σ [20].

The first attempt at incorporating artificial neural networks into the ABC framework was proposed by Blum and Francois [21]. Rather than constructing optimal summary statistics, the goal of their implementation was to utilize non-linear regression models (including an FFNN) in order to maximize model efficiency [21]. They achieved this by using a FFNN to project the summary statistics to a lower dimensionality representation in order to improve efficiency [21].

Before proceeding to discuss more recent advancements in the application of neural networks to ABC we first clarify some of the terminology concerning neural networks that often becomes misconstrued. "Artificial neural network" is merely the full name for neural networks and does not necessarily indicate any sort of deviation from the neural networks we have discussed thus far. The term can thus be used generally to refer to any neural network. The term "deep neural network (DNN)" however refers to a neural network containing more than two layers of adaptive weights [20, 52]. Also note that in general neural networks are all "feed forward" (i.e. containing no feedback loops) except in cases where a recursive property is specifically included [20]. Lastly, with regards to the naming convention whereby neural networks are described as "n-layer neural network", we echo the sentiment expressed by Bishop [20] and choose a naming convention whereby the number of layers of adaptive weights indicates the proper name of a neural network. For example, the neural network shown in Figure 6.3 has two layers of adaptive weights and is thus a "two-layer neural network".

More recent attempts at automatic summary statistic construction have relied on other versions of neural

[2] Source: https://1e.to/CkuNlC (Accessed October 2021)

networks. First of these advancements was Jiang et al. [58] who trained a DNN to approximate the mean of the true posterior distribution of the parameter(s) of interest. This was based on the findings of a previous paper by Fearnhead and Prangle [43] where they showed that a sufficient summary statistic for ABC can be formulated by using the mean of the true posterior distribution, $\mathbb{E}[\theta|X]$. Their methodology entailed simulating a large number of training sets $\left\{\theta^{(i)}, X^{(i)}, 1 \leq i \leq N\right\}$ by drawing samples from the joint distribution $\pi(\theta, X)$ which were then used to train a DNN with $X^{(i)}$ as input and $\theta^{(i)}$ as output [58]. This circumvented the concern of sufficient data by allowing any number of training samples to be simulated [58]. The DNN was trained by minimizing the squared error loss function given by

$$J(\boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} \left\| f_{\boldsymbol{w}} \left(X^{(i)} \right) - \theta^{(i)} \right\|_2^2 \tag{6.10}$$

where $\boldsymbol{w}$ is the vector of all DNN parameters (i.e. weights and bias terms) and the DNN is indicated by $f_{\boldsymbol{w}}$ [58]. They also showed that any function that minimizes (6.10) can be considered an approximation of the posterior mean $\mathbb{E}[\theta|X]$ [58]. The choice of using a DNN was motivated by a desire for high representative power for high-dimensionality non-linear functions and showed an improvement over previous implementations in terms of accuracy [58].

As previously stated, ABC algorithms were initially developed for use in the field of genetics [60, 21, 14, 76, 107]. A feature of population data is exchangeability and so far there have been two attempts made at developing an automatic summary statistic construction that takes this property into account [28, 78]. These two methods, proposed by Chan et al. [28] and Wiqvist et al. [78], utilize a variation of neural networks known as exchangeable neural networks (ENN). We mention these contributions here for the sake of completeness but will not discuss them in further detail due to a lack of relevance for our practical objective. See [28, 78] for further information.

Since their creation, neural networks have become staples in the field of image analysis [19, 20]. A convolutional neural network (CNN) is a specific type of neural network that is adept at processing data that can be arranged into a grid-like structure such as images or time series [52]. CNN derives its name from the *convolutional* operation that is performed (at least once) within its hidden layers [20, 52]. While not utilized in this mini-dissertation, CNN's represent a viable tool for the purpose of summarizing time-series data and a full discussion on these models is included in Appendix C. Kesson et al. [2] compared the performance of a CNN, DNN and PEN and showed that CNNs performed better in cases of high-dimensional and increasingly complex problems for likelihood-free inference [2].

6.2 Backpropogation and gradient descent

We now discuss how neural networks are trained on existing data through two distinct stages, those being backpropagation (or sometimes simply *backprop*) as well as gradient descent [20, 52]. A fully connected neural network with multiple layers of adaptive weights can have a staggering amount of weight and bias parameters that need to be estimated. As with other machine learning regression techniques, the primary idea behind fitting a neural network lies in the minimization of some loss function such as the sum of squared residuals [20]. By evaluating the loss function at each iteration we can adjust parameter values such that the loss function value becomes smaller [20]. The ideal choice of loss function will be determined by the modelling objective[3], with some general guidelines given in Table 6.2.

Table 6.2: Appropriate loss functions

Modelling problem	Loss function(s)	Function
Regression	Mean Squared Error (MSE)	$\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$
Binary or multi-class classification	Cross-entropy	$\frac{1}{n}\sum_{i=1}^{n}\left[Y_i \cdot \log(\hat{Y}_i) + (1 - Y_i) \cdot \log(1 - \hat{Y}_i)\right]$

The process of improving upon a desired loss function requires us to first determine the derivative of the loss function with regards to the weight and bias parameters [20]. Backpropogation refers to the process of developing computationally efficient expressions that can be used to evaluate these derivatives [52, 20].

In order to explain the process of backpropagation we utilize an example of a simple FFNN shown in Figure 6.4. In this neural network a one-dimensional input value is passed onto two different weight-bias pairs $\left(w_{11}^{(1)}, w_{10}^{(1)} \text{ and } w_{21}^{(1)}, w_{20}^{(1)}\right)$ to produce the activations a_1 and a_2. These activations are then used as input to the activation function $h(\cdot)$ to produce the transformed activations z_1 and z_2. The final output of the neural network is then a weighted sum of these values and a final bias term (note, for simplicity we do not include another transformation function before the final output layer). This neural network was two layers of adaptive weights and a total of 7 parameters that need to be estimated (four within the first layer and three within the second layer).

In order to derive a set of equations with which we can evaluate the derivatives of some loss function $E(\boldsymbol{w})$ to the neural network parameters we use the chain rule of differentiation which we will briefly recap [20, 52]. Let $x \in \mathbb{R}$, and let $f, g : \mathbb{R} \to \mathbb{R}$. Suppose that $y = g(x)$ and $z = f(g(x)) = f(y)$. The chain rule

[3] Machine learning mastery: https://1e.to/gy9oif (Accessed October 2021

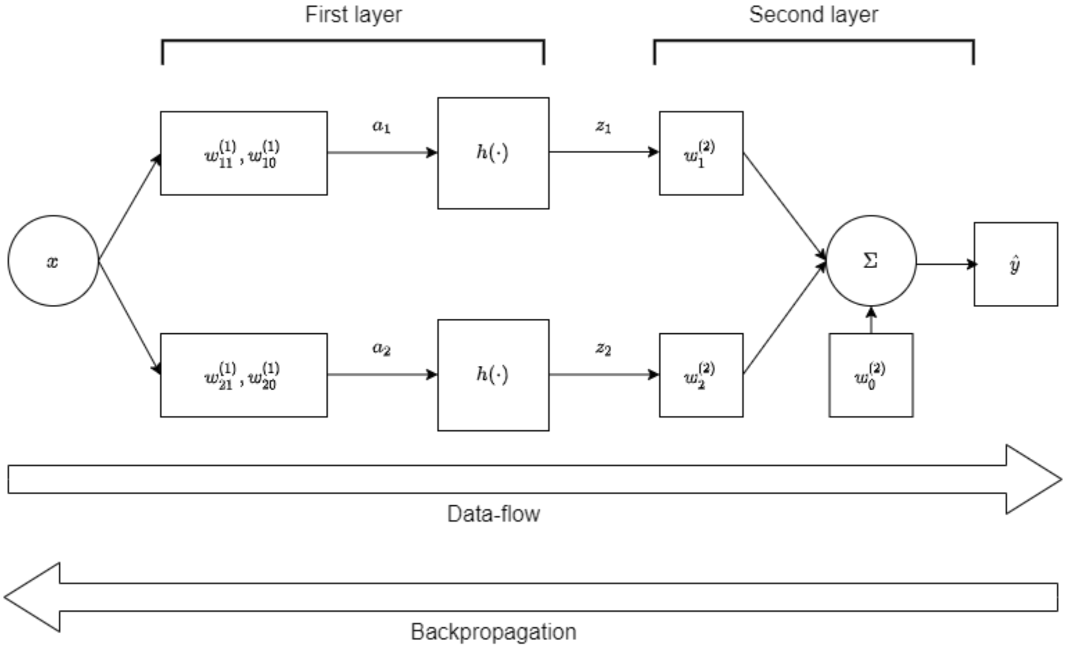


Figure 6.4: Backpropogation diagram

then states that

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}. \tag{6.11}$$

Suppose that $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{y} \in \mathbb{R}^n$, $g : \mathbb{R}^m \to \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$ and $\boldsymbol{y} = g(\boldsymbol{x})$ and $z = f(g(\boldsymbol{x})) = f(\boldsymbol{y})$, then we can generalize this result to vectors as follows

$$\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j} \cdot \frac{\partial y_j}{\partial x_i}. \tag{6.12}$$

For the example discussed in this section we will utilize the sum of squared residuals as our loss function. Note however that all derivations can be extended arbitrarily to any scalar-valued loss function [52, 20]. Recall that the neural network depicted in Figure 6.4 is feed-forward, i.e. the data is fed forward from one end of the neural network to the other (progressing in one direction from input to output as indicated by the arrow) [52, 20]. When performing backpropagation however, we proceed backwards through the neural network, starting from the output layer and ending at each desired parameter, developing an expression for the corresponding derivative using the chain rule [52, 20]. Starting from what we previously derived at Equation 6.9, we can simplify the calculation performed in this neural network as follows

$$y_k(\boldsymbol{w}, \boldsymbol{x}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} \cdot h\left(\sum_{i=1}^{n} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right)$$

$$\begin{aligned}
y(\boldsymbol{w}, \boldsymbol{x}) &= \sigma\left(\sum_{j=1}^{M} w_j^{(2)} \cdot h\left(\sum_{i=1}^{n} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_0^{(2)}\right) \text{ (output is 1-dimensional so } k = 1\text{)} \\
&= \sum_{j=1}^{M} w_j^{(2)} \cdot h\left(\sum_{i=1}^{n} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_0^{(2)} \text{ (no second activation function)} \\
&= \sum_{j=1}^{M} w_j^{(2)} \cdot h\left(w_{j1}^{(1)} x + w_{j0}^{(1)}\right) + w_0^{(2)} \text{ (data is 1-dimensional so } i = 1\text{)} \\
&= w_1^{(2)} \cdot h\left(w_{11}^{(1)} x + w_{10}^{(1)}\right) + w_2^{(2)} \cdot h\left(w_{21}^{(1)} x + w_{20}^{(1)}\right) + w_0^{(2)} \text{ (expanding summation).}
\end{aligned}$$

We therefore have that

$$\hat{y} = w_1^{(2)} \cdot h\left(w_{11}^{(1)} x + w_{10}^{(1)}\right) + w_2^{(2)} \cdot h\left(w_{21}^{(1)} x + w_{20}^{(1)}\right) + w_0^{(2)}. \tag{6.13}$$

We can thus express our loss function in terms of the model parameters as follows

$$\begin{aligned}
SSR &= \sum_{i=1}^{N} \left[y_i - \hat{y}_i\right]^2 \\
&= \sum_{i=1}^{N} \left[y_i - \left[w_1^{(2)} \cdot h\left(w_{11}^{(1)} x_i + w_{10}^{(1)}\right) + w_2^{(2)} \cdot h\left(w_{21}^{(1)} x_i + w_{20}^{(1)}\right) + w_0^{(2)}\right]\right]^2
\end{aligned}$$

where N is the total number of observations in the training dataset. We first determine the derivative of the predicted values with regards to the model parameters,

$$\begin{aligned}
\frac{\partial \hat{y}}{\partial w_0^{(2)}} &= 1 \\
\frac{\partial \hat{y}}{\partial w_1^{(2)}} &= h\left(w_{11}^{(1)} x_i + w_{10}^{(1)}\right) \\
\frac{\partial \hat{y}}{\partial w_2^{(2)}} &= h\left(w_{21}^{(1)} x_i + w_{20}^{(1)}\right) \\
\frac{\partial \hat{y}}{\partial w_{10}^{(1)}} &= h'\left(w_{11}^{(1)} x_i + w_{10}^{(1)}\right) \\
\frac{\partial \hat{y}}{\partial w_{20}^{(1)}} &= h'\left(w_{21}^{(1)} x_i + w_{20}^{(1)}\right) \\
\frac{\partial \hat{y}}{\partial w_{11}^{(1)}} &= h'\left(w_{11}^{(1)} x_i + w_{10}^{(1)}\right) \cdot x_i \quad \text{(chain rule)} \\
\frac{\partial \hat{y}}{\partial w_{21}^{(1)}} &= h'\left(w_{21}^{(1)} x_i + w_{20}^{(1)}\right) \cdot x_i \quad \text{(chain rule)}
\end{aligned}$$

where h' indicates the first derivative of the activation function h with respect to the activation. Using all the derivations we've made thus far, we can finally derive expressions for the derivative of the loss function with regard to the model parameters as follows:

$$
\begin{aligned}
\frac{\partial SSR}{\partial w_0^{(2)}} &= \frac{\partial SSR}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial w_0^{(2)}} \\
&= 2\sum_{i=1}^{N} [y_i - \hat{y}_i] \times (-1) \\
&= -2\sum_{i=1}^{N} [y_i - \hat{y}_i] \\
\frac{\partial SSR}{\partial w_1^{(2)}} &= \frac{\partial SSR}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial w_1^{(2)}} \\
&= 2\sum_{i=1}^{N} [y_i - \hat{y}_i] \times \left(-h\left(w_{11}^{(1)} x_i + w_{10}^{(1)}\right)\right) \\
&= -2\sum_{i=1}^{N} [y_i - \hat{y}_i] \cdot h\left(w_{11}^{(1)} x_i + w_{10}^{(1)}\right) \\
\frac{\partial SSR}{\partial w_2^{(2)}} &= \frac{\partial SSR}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial w_2^{(2)}} \\
&= -2\sum_{i=1}^{N} [y_i - \hat{y}_i] \cdot h\left(w_{21}^{(1)} x_i + w_{20}^{(1)}\right) \\
\frac{\partial SSR}{\partial w_{10}^{(1)}} &= \frac{\partial SSR}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial w_{10}^{(1)}} \\
&= -2\sum_{i=1}^{N} [y_i - \hat{y}_i] \cdot h'\left(w_{11}^{(1)} x_i + w_{10}^{(1)}\right) \\
\frac{\partial SSR}{\partial w_{20}^{(1)}} &= \frac{\partial SSR}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial w_{20}^{(1)}} \\
&= -2\sum_{i=1}^{N} [y_i - \hat{y}_i] \cdot h'\left(w_{21}^{(1)} x_i + w_{20}^{(1)}\right) \\
\frac{\partial SSR}{\partial w_{11}^{(1)}} &= \frac{\partial SSR}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial w_{11}^{(1)}} \\
&= -2\sum_{i=1}^{N} [y_i - \hat{y}_i] \cdot h'\left(w_{11}^{(1)} x_i + w_{10}^{(1)}\right) \cdot x_i \\
\frac{\partial SSR}{\partial w_{21}^{(1)}} &= \frac{\partial SSR}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial w_{21}^{(1)}} \\
&= -2\sum_{i=1}^{N} [y_i - \hat{y}_i] \cdot h'\left(w_{21}^{(1)} x_i + w_{20}^{(1)}\right) \cdot x_i
\end{aligned}
$$

This concludes the application of backpropagation to this particular problem. Note how even for a relatively simple neural network many derivations and calculations are involved in backpropagation. Note also how the expressions for the derivatives become increasingly involved the closer a parameter is to the input layer in the neural network, since the chain rule needs to be applied a larger number of times in order to evaluate the derivatives with respect to these parameters. We now proceed to explain how these equations may be utilized for gradient descent.

We have thus far determined the derivative of the loss function with respect to the model parameters. We wish to utilize these derivatives to guide the process of selecting parameter values that minimize the loss function, for which we turn to gradient descent [20, 52]. Having selected a vector of initial model parameters $\boldsymbol{w}^0$, basic gradient descent proceeds using the following equation

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} - \eta \nabla E\left(\boldsymbol{w}^{\tau}\right) \tag{6.14}$$

where $\nabla E\left(\boldsymbol{w}^{\tau}\right)$ is the derivative of the loss function for parameter values $\boldsymbol{w}^{\tau}$ [20]. We first initialize all model parameters (either randomly from a chosen distribution or by choosing specific values). We then pass the training data through the model and calculate the derivative for the loss function with respect to each parameter value [20]. The derivative values inform us what adjustments need to be made to the parameter values to improve the model fit. We adjust the parameter values using the derivatives as well as a *learning rate* indicated by η [20] using Equation (6.14). We repeatedly utilize Equation (6.14) until either some stopping criteria is satisfied or until the parameter values have reached convergence [20]. This concludes the discussion on backpropagation and gradient descent for this mini-dissertation. For further reading on these topics as well as the expansive field of neural networks in general, please see [19, 20, 52].

6.3 Autoencoders

Thus far we have discussed the origin and nature of ANN's as well as how they are trained using observed data. In this section we offer more information on how ANN's will be used specifically within the context of approximate Bayesian computation as discussed in the previous chapter. Recall from our previous discussion that the process of determining whether to accept or reject proposed parameter values can be rendered more computationally efficient through the use of summary statistics [60, 130, 81, 124, 15]. We achieve this by comparing the summary statistics derived for the observed and simulated datasets rather than the datasets themselves. The choice of summary statistic then represents a non-trivial concern. In most applications, the choice of summary statistic is determined by the specific nature of the data and study objectives [126, 117, 81, 124]. In this mini-dissertation we evaluate the viability of training ANN's to learn a lower-dimensionality representation of both the observed and simulated datasets for efficient comparison. We achieve this through the use of a specific type of ANN known as an autoencoder.

Autoencoders are neural networks with a very specific architecture, whereby the desired output is the input. Autoencoders consist of two main parts: an encoder and a decoder [64]. The encoder condenses the input data to some desired dimensionality, this condensed form of the input data is referred to as “code” [64]. The decoder then processes this data to once again derive the original input data with as little information lost as possible [64]. A schematic representation of an autoencoder ANN is shown in Figure 6.5. Since the code is a condensed form of the data, it can serve as a summary statistic and can be utilized to compare the similarity of two sets of data.

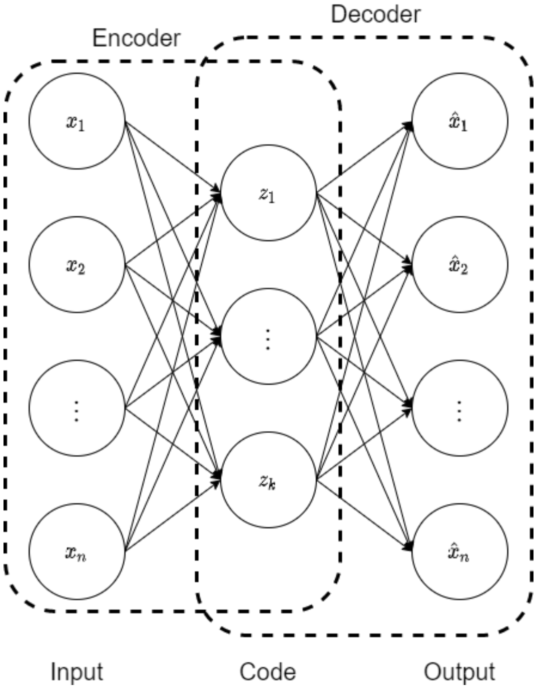


Figure 6.5: Diagram of an autoencoder neural network

Note that the autoencoder ANN depicted in Figure 6.5 is a special case of an ANN, restricted to the desired output being identical to the input. Naturally it is possible to formulate encoders based off of other types of neural networks as well by imposing this same restriction. Similarly to the relationship between DNN and ANN models, an autoencoder DNN is simply an autoencoder ANN with more hidden layers included between the input, code and output [64]. Other than the restriction imposed on the output layer, there is nothing else about autoencoders that require any special consideration and all previous derivations still hold.

In this mini-dissertation we will consider the application of both an autoencoder ANN and autoencoder DNN for the purpose of summarizing pandemic data. After fitting the models to a large sample of simulated pandemics we will determine how capable they are at identifying simulated pandemics that most closely represent our observed data.

6.4 Conclusion

In this chapter we discussed the history and theory behind artificial neural networks. While typically employed as a “black box”, the mathematical derivations behind artificial neural networks is quite intuitive. The reason for their reputation as black boxes is partially due to their large number of parameters being infeasible for users to keep track of and individually investigate even for simpler models.

We discussed the choices that must be made when using artificial neural networks, such as the activation and loss function and the number of hidden layers. We then discussed how one would fit these models using backpropagation and gradient descent. Finally, we discussed autoencoders as a special type of neural network and how they will be used in this mini-dissertation.

In the next chapter we discuss the application of the various theoretical principles discussed in this mini-dissertation up to this point.

Chapter 7

Application

In this chapter we discuss the application of the theoretical concepts outlined in the rest of this mini-dissertation. All analysis presented in this mini-dissertation was performed using a desktop computer running Intel Core i7 with a clock speed of 3.40 GHz, a 64-bit Windows operating system and 64 GB of installed RAM. All analysis was completed using either `Python` or `R`.

7.1 Web scraping

The web scraping of the Western Cape premier's news updates regarding COVID-19 case numbers was achieved using `Jupyter Notebook` version 6.3.0. The most instrumental packages for achieving this were `Selenium` and `pandas`. Functions contained within the `Selenium` package were utilized in order to open chromium based web browser windows and to interact with the web pages in a desired way. The `pandas` package contains functions that allow for the easy extraction of tables from web pages.

Unlike other research endeavours that can be parallelized or otherwise sped up through improved hardware, the time taken to successfully web scrape a desired dataset is less flexible. In this mini-dissertation, the time taken to web scrape the entire dataset (not including data cleaning) was approximately 1 hour and 12 minutes. This long run-time is due to a crawl delay needing to be included between successive requests for information. This is done in order to avoid overloading the server that the data is hosted on and is general good practice. The robots.txt file for the South African Government media repository (found in Appendix A) specifies that a crawl delay of 10 seconds must be used. This means that any web scraper attempting to gather data from the website must wait at least 10 seconds between successive requests for information, which naturally increases the run-time. The run-time is additionally influenced by available bandwidth and the speed with which web pages can be loaded.

The end result of the web scraping was a dataset containing the daily increase in COVID-19 cases in the Western Cape province at district municipality level as published in the press releases of the Western Cape premier Alan Winde. Along with the data obtained from the NICD's website (see Section 2.1.3), this means that there are two sets of COVID-19 data for the same study region and period of time but from different reputable sources that will be considered in this mini-dissertation. Both datasets will be utilized in separate models in order to determine whether there are any significant differences that can be observed with regards to model inference.

In this mini-dissertation the study period is 1 April 2020 - 30 September 2020 (181 days). This time period was chosen due to it spanning the first wave of COVID-19 infections in the study region. The study region is sub-divided into six district municipalities. The data dimensionality (for both sources) is thus 181×6 entries.

7.2 Population mobility data

Recall that two sources of population mobility data (mobile network data and Facebook data) were utilized to incorporate population mobility dynamics in this mini-dissertation. All analysis performed on this data was done in `RStudio` using only functions typically available in base `R`.

The first mobility dataset was obtained through a local mobile network provider and proved very cumbersome to work with. With a total size of 2.98 GB, this data slowed certain parts of the application down to a crawl due to its sheer size. The data is originally available at ward level, however in order to derive comparisons between it and the Facebook data it was necessary to convert the data to a lower spatial resolution, in this case district municipality level. This conversion required upward of 3 weeks to complete successfully. It is noted however that this process could be sped up exponentially through proper parallelization, distributed computing or the use of a faster programming language such as $\texttt{C}++$. This data was utilized to set up one of the candidate spatial weight matrices considered in this mini-dissertation (see Method 2, Section 3.2.2).

The second dataset was population mobility data obtained through Facebook's "Facebook data for good" program. Due to the relatively low spatial resolution and lack of direction of the mobility estimates the data is relatively small, at only 890 KB. This rendered the data easy and quick to use. It is worth noting however that the original dataset that was downloaded from the “Data for good” website was much larger due to including information for a large number of countries across the world, with a total size of 336 MB. This dataset was somewhat cumbersome and required some time for RStudio to properly load and interact with.

In its original format the Facebook data indicates the population mobility within each district municipality relative to a baseline calculated in February 2020. This data was used to set up two of the candidate spatial weight matrices considered in this mini-dissertation (see Method 3 and Method 4, Sections 3.2.3 and 3.2.4 respectively).

7.3 Compartmental model setup

The compartmental model construction and fitting was achieved using the ABSEIR package, version 1.3, in RStudio version 4.1.0. This package was created and is maintained by Dr Grant Brown and can be downloaded through their GutHub page[1]. This package contains all the functions necessary to set up, fit and evaluate the spatial SEIR model previously discussed in Chapter 4.

The models were fitted to non-cumulative COVID-19 case data. Both the web scraped press release data as well as the data obtained from the NICD’s website (both most recently discussed in Section 7.1 and first introduced in Section 2.1) were used in separate models in order to study the difference in model inference that can be made for these two datasets. It was observed that it was generally easier to obtain a good fit for a model when fitted to non-cumulative as opposed to cumulative data. While this is surprising, since the results ought to be equivalent, we speculate that this is due to a possible source of bias present in the model fitting procedure. We will discuss this source of bias in the following chapters.

The first step in constructing a compartmental model using the ABSEIR package is to construct the data model, which describes the nature of the data. In the following snippet of code we define the data model using COVID-19 case data stored as “I_star” which is either the web scraped press release data or the NICD data. We specify “type = identity” to indicate that the data is the actual case data and not a function thereof. We then indicate that this data corresponds to the $\boldsymbol{I}^*$ model compartment we previously discussed in Section 4.3 (i.e. those individuals transitioning into the infectious compartment). Lastly, we indicate that the data is not cumulative. The prior precision was set relatively low in order to allow for greater variation in the simulated parameter values.

```
# Set up data model
```

[1] Available at: https://github.com/grantbrown/ABSEIR

```
data_model <- DataModel(Y = I_star,
                          type = "identity",
                          compartment = "I_star",
                          cumulative = "FALSE")
```

The next step is to set up the exposure model, or rather the design matrix for the linear parameterization of the intensity process as described in Section 4.3. Recall that the intensity process describes the progression of the disease over time, specifically the probability of a susceptible individual becoming exposed to the disease. A simple 3-degree polynomial (with all parameters shared between all included district municipalities) was used to model the time-varying intensity process with a separate intercept term included for each district municipality.

In the following snippet of code we set up the design matrix `X` as just described. We then create the exposure model using the function of the same name with `X` as the first argument. The second and third arguments, `nTpt` and `nLoc`, indicate the length of the study period and the number of locations respectively. The final two arguments are `betaPriorPrecision` and `betaPriorMean`. All covariates for the exposure model are given beta prior distributions, for which we can set the prior mean and precision. Initially all priors were set as indicated in the code below.

```
# Set up Exposure Model (exposure varies over time as a 3-degree spline function)
# intercept for each region (i.e. each column of I_star)
intercept <- matrix(1, nrow = nrow(I_star), ncol = 1)
# a 3-degree spline function
timeBasis <- poly(1:nrow(I_star), degree = 3)[rep(1:nrow(I_star), 1),]

# combine to create design matrix X
X <- cbind(intercept, timeBasis)
# set priors
exposure_model <- ExposureModel(X, nTpt = nrow(I_star),
                                nLoc = ncol(I_star),
                                betaPriorPrecision = 0.2,
                                betaPriorMean = c(rep(-1, ncol(intercept)),
                                                  rep(0, ncol(timeBasis))))
```

In order to build a greater understanding of the modelling process and how the chosen model(s) perform under various conditions it was decided to evaluate the fit of several initial candidate models. These models would each start out focusing solely on individual district municipalities within the study region and were gradually expanded to include more district municipalities. Once these preliminary modelling results were finalized, the posterior means from these models were used as the prior mean values for the covariates in the exposure model (i.e. `betaPriorMean`).

As mentioned previously, we do not consider the possibility of reinfection in this mini-dissertation due to the relatively short study period. We thus specify that the model contains no possibility of transition from the "Removed" compartment to the "Susceptible" compartment, or rather, that the model is a *SEIR* model.

```
# Specify reinfection model
reinfection_model <- ReinfectionModel("SEIR") # no reinfection
```

Next we set up the spatial association model over the study region, referred to in the package documentation as the "distance model". We first import the spatial weight matrices constructed using the mobile network and Facebook data. We then calculate the average spatial weight matrix over the study period for each dataset and store these as our spatial weight matrices (`Mobile_net_spatial_weights` and `Facebook_spatial_weights` respectively) in a list. The model allows for the spatial autocorrelation parameters to have gamma distributed priors, for which we specify the parameter values `priorAlpha` and `priorBeta`. The mean of the prior distribution(s) were kept low due to prior beliefs that the spatial autocorrelation over the study period would not be very large due to COVID-19 lockdown regulations (see Table 1.1).

```
# Set up Distance Model
distance_model <- DistanceModel(list(Mobile_net_spatial_weights,
                                     Facebook_spatial_weights),
                                     priorAlpha = 1, priorBeta = 10)
```

Next we specify the starting values for the pandemic under investigation. We thus provide the values $\boldsymbol{S}_0, \boldsymbol{E}_0, \boldsymbol{I}_0$ and $\boldsymbol{R}_0$. Since the available data only indicates the number of new confirmed cases on each day, we need to make an assumption regarding the number of individuals who are exposed at the start of the study period. We also need to make an assumption regarding individuals who have already recovered by the time the study commences. In this mini-dissertation we opt to assume that no individuals have recovered by the commencement of the study period. We also assume that the number of exposed individuals is the sum of those individuals that end up testing positive for COVID-19 within the next five days. We also import data indicating the district municipality population and store as `N`. The initial value for the susceptible compartment is then simply the remaining population after the other compartment values have been determined. The code to achieve is given in the following snippet.

```
# Import population size data (data is for whole country)
pop <- read_xlsx("Population_data.xlsx")
pop <- pop[1:6,] # first six district municipalities are in our study region
# District municipality populations make up N
N <- pop$Population

# Initial exposed, sum of first five days (assume that they start exposed)
E0 <- apply(I_star[1:5,], 2, sum, na.rm = TRUE)
# Set up Initial Value Container
initial_value_container <- InitialValueContainer(S0 = N - I0 - E0,
                                                 E0 = E0,
                                                 I0 = I0,
                                                 R0 = rep(0, ncol(I_star)))
```

The final component of the compartmental model that needs to be calibrated is the latent and infectious time probabilities. In this mini-dissertation we studied the use of exponential as well as gamma distributions to model these probabilities. While the former is used most often in literature due to its simplicity, the latter takes advantage of the flexibility of the compartmental model derived in Section 4.3 and allows any general distributions to be specified for the random variables Z_1 and Z_2 (indicating the latent and infectious time respectively, see Equations (4.3) and (4.4)). In either case, we assume that the latent and infectious periods have mean duration of 4 and 10 days respectively [42]. The gamma parameterization was chosen to allow for a greater degree of flexibility than would be possible when using a standard exponential distribution as is used frequently in applications of SEIR models [72, 98]. When using the exponentially distributed transition probabilities it is also necessary to specify an effective sample size to indicate the believed reliability of the priors. The effective sample sizes were simply set to 100 for the application in this mini-dissertation.

```
# Gamma distributed transition probabilities
transition <- PathSpecificTransitionPriors(Z1 = function(x){dgamma(x, shape = 2,
                                                                      rate = 0.5
                                                                      )},
                                           Z2 = function(x){dgamma(x, shape = 2,
                                                                      rate = 0.2
                                                                      )})
# OR #
# Exponentially distributed transition probabilities
transitionPriors <- ExponentialTransitionPriors(p_ei = 1-exp(-1/4),
                                                p_ir = 1-exp(-1/10),
                                                p_ei_ess = 100,
                                                p_ir_ess = 100)
```

For all results presented in the following chapter the latent and infectious times may be assumed to follow a gamma distribution unless stated otherwise. The differences in inference for these two distributions will be specifically investigated in Chapter 8.

7.4 Compartmental model fitting

Having defined our compartmental model, the next step is to select an ABC algorithm to fit the compartmental model to the observed data. We do this by utilizing the `SamplingControl` function to create a sampling control object as shown in the code below. The function accepts a seed value as its first argument to ensure reproducibility. The second parameter allows the user to specify the number of CPU cores that the process is allowed to utilize for the sampling. The machine utilized for the analysis in this mini-dissertation has a total of 12 CPU cores. For general purpose model fitting 6 cores were used to allocate sufficient resources towards completing the analysis in a timely manner. In cases where more speed was desired this was increased to 8 cores. In general there were no issues observed with the performance of the sampling and the function seems to be very well optimized.

The third argument to the `SamplingControl` function is the ABC algorithm that should be used to fit the compartmental model. In this mini-dissertation we discussed the three primary classes of ABC algorithms, those being the basic ABC rejection, Markov chain Monte Carlo (MCMC ABC) and sequential Monte Carlo (SMC ABC) algorithms (see Chapter 5). These algorithms are given by Algorithms 1, 3 and 5 respectively. In order to select these algorithms the user must specify either “BasicABC”, “DelMoral2006” or “Beaumont2009” (named after those that proposed the algorithms and in what year). Note however that the `ABSEIR` package is currently set up to only fit the model using either the ABC rejection algorithm or the SMC ABC algorithm, with other implementations of ABC being planned for future updates.

```
# Set up Sampling Control object
sampling_control <- SamplingControl(seed = 230597,
                                     n_cores = 6,
                                     algorithm = "Beaumont2009",
                                     list(batch_size = 2000,
                                          epochs = 1e6,
                                          max_batches = 100,
                                          shrinkage = 0.99))
```

Due to the ABC rejection algorithm [110] being highly dependent on the selection of appropriate priors, the SMC ABC algorithm as implemented by Beaumont et al. [16] was used to evaluate the preliminary models. After the series of preliminary models was finalised, two models were fitted that included all district municipalities. The first model was fitted using the ABC rejection algorithm and the other using the SMC ABC algorithm. This allowed us to compare the performance of the two algorithms. With regards to performance it may be noted that the run-time of the ABC rejection algorithm was very fast and hardly noticeable, while the run-time of the SMC ABC algorithm could be potentially long (ranging from 20 minutes to an hour using 6 CPU cores).

The final argument to the `SamplingControl` function is a list of additional arguments that may be required by some algorithms. For example in the code snippet included we set the batch size equal to 2000. This indicates the number of pandemics that must be simulated in parallel before being compared to the observed data to determine their similarity. The number of epochs indicates the maximum number of iterations to run and is only relevant to algorithms that iteratively perform simulations such as the SMC ABC algorithm. The maximum number of batches indicates the number of pandemics that may be simulated in parallel before a proposed set of parameter values must be accepted. In this code snippet it indicates that if no new parameter values are accepted after 100 pandemics are simulated in parallel then the process will terminate under the assumption of convergence. Finally, the shrinkage argument indicates the proportion by which the tolerance values for the SMC ABC algorithm must be decreased at each iteration. In the code snippet included above we see that the tolerance values at each iteration will be 99% of the previous tolerance value. For every model fitted using the SMC ABC algorithm the argument values were kept as is shown in the code snippet above.

In order to commence with the fitting of the compartmental model we use the following snippet of code. The function to commence with the fitting of the model is named `SpatialSEIRModel`. It receives all the previously constructed objects as arguments followed by the number of samples that should be drawn and a final argument indicating whether additional output should be printed to the screen. We experienced that it was often beneficial to set the final argument to TRUE, especially when formulating initial candidate models using the SMC ABC algorithm. This is because the tolerance values used for each iteration will be printed to the screen. In general, if the tolerance values are extremely large for early iterations it can be safely concluded that the model will not produce a good fit and the process may be terminated (unless one wishes to observe the performance of the model knowing that it will not produce a good fit).

```
# Set up Sampling Control object
model1 <- SpatialSEIRModel(data_model,
                           exposure_model,
                           reinfection_model,
                           distance_model,
                           transition_priors,
                           initial_value_container,
                           sampling_control,
                           samples = 1000,
                           verbose = TRUE)
```

In order to compare the goodness of fit for two fitted models, say model1 and model2, we can calculate the Bayes factor using the following snippet of code. The first argument modelList is a list containing the models to be evaluated and the second argument, n_samples, indicates the number of samples to use.

```
# Determine Bayes factor for two models
BayesFactor <- compareModels(modelList = list(model1, model2), n_samples = 100)
```

Other properties of the ABC fitting in the ABSEIR package that are worth noting is that the similarity between simulated pandemics and the observed data is determined using the Euclidean distance function. There is currently no way for the user to select a different function. Also note that no summary statistics are enabled for the ABC algorithms in this package. This means that the entire observed and simulated dataset is compared when deciding which parameter values to retain. There is currently no way to incorporate summary statistics directly.

We note that the ABSEIR package is being continuously updated with new functionality. Any enquiries related to the package should be directed to the author's GitHub page. We found replies to raised issues were prompt and very thorough.

7.5 Artificial neural network training

The final analysis in this mini-dissertation was focused on determining the usability of autoencoder neural networks for serving as summary statistics within an approximate Bayesian framework. This would equate to using neural networks as summary statistics within the decision step of the ABC algorithm, however as noted previously there is currently no summary statistic functionality included in the `ABSEIR` package. The use of artificial neural networks in `RStudio` is somewhat reliant on the successful integration of `R` and `Python`, with packages such as `keras` and `tensorflow` being available in `R` but native to `Python`. Following some difficulty experienced with this integration, it was decided to fit the artificial neural networks using `Jupyter notebook` but to perform the required simulation in `R` since it requires the `ABSEIR` package.

In order to train the autoencoders we first simulated a large sample of artificial pandemics ($n = 1000$) using the posterior estimates from the SMC ABC algorithm model to set simulation parameter values. We then trained both an autoencoder ANN as well as an autoencoder DNN to compress this data to a lower dimensionality.

Having trained the two autoencoders, we then simulated another large set of artificial pandemics. We determined the implied tolerance level for these simulated pandemics (i.e. the minimum tolerance, ϵ, required to accept all of them within an ABC framework). Using the autoencoders we then condensed both these simulated pandemics and the observed pandemic data down to a lower dimensionality. We then determined which of the simulated pandemics had summary statistics that most resembled that of the observed data. We considered the 100 pandemics with the lowest difference in summary values and determined the implied tolerance for these pandemics. If the implied tolerance for these pandemics is lower than previously determined for the whole set, we can conclude that the autoencoders were successful in identifying the simulated pandemics that most closely resemble the observed data.

The autoencoder ANN contained only two sets of adaptive weights while the autoencoder DNN contained a total of six sets of adaptive weights. The artificial autoencoder contained only 30 hidden nodes while the deep alternative contained five sets of hidden nodes of size 128, 64, 32, 64 and 128 respectively. This implies that the two models condense the data down to a dimensionality of 30 and 32 respectively (indicated by the number of hidden nodes in the centre of each model). The original size of the input data is 1086 entries (181 days × 6 district municipalities) and thus these imply a data reduction of approximately 36. This value was chosen to showcase the potential dimensionality reduction capabilities of these models.

All pandemic values were scaled to be within the interval $(0, 1)$ and cross-entropy (see Table 6.2) was chosen as the loss function. ReLu activation functions were used at every hidden layer of the network while the sigmoid function was used in the final output layer. The number of epochs was set to 100 and the batch size was 200, meaning that the training data was passed through the model 100 times in random batches of size 200. We include a code snippet detailing how to setup and fit the autoencoder DNN model in Python.

```
# This is our input
input_ = keras.Input(shape=(1086,))
# input gets passed through several hidden layers
encoded = layers.Dense(128, activation = "relu")(input_)
encoded = layers.Dense(64, activation = "relu")(encoded)

# central hidden layer, the encoded data or simply "code"
encoded = layers.Dense(32, activation = "relu")(encoded)

# this next part decodes the data again
decoded = layers.Dense(64, activation = "relu")(encoded)
decoded = layers.Dense(128, activation = "relu")(decoded)
decoded = layers.Dense(1086, activation = "sigmoid")(decoded)

# put model together
autoencoder = keras.Model(input_, decoded)

# encoder
encoder = keras.Model(input_, encoded)

# compile model
autoencoder.compile(optimizer = "adam", loss= "binary_crossentropy")

# fit model to training data (0.8 of simulated pandemics)
autoencoder.fit(x_train, x_train,
                epochs = 100,
                batch_size = 200,
                shuffle = True,
                validation_data = (x_test, x_test))
```

7.6 Conclusion

This concludes our discussion on the application of the theoretical principles discussed in this mini-dissertation. Thanks to proper documentation, easily accessible resources on the Internet and attentive assistance from the package maintainers for the `ABSEIR` package, the analysis was able to proceed at a satisfactory pace with no serious complications or issues preventing analysis.

We do note however that the aggregation of the mobile network to a lower spatial resolution was not done in an optimal fashion and took much longer than it should have. Future attempts to utilize such data will benefit greatly from early consideration of parallelization and distributed computing techniques.

Chapter 8

Results

In this chapter we present the results of the application performed in this mini-dissertation. We outline the results obtained for the various compartmental epidemiological models utilized. We also discuss preliminary modelling results, ABC algorithm performance, spatial autocorrelation, the empirically adjusted reproductive number, comparison of insights derived for the two sources of COVID-19 data and the application of autoencoders for data dimensionality reduction.

8.1 Preliminary modelling results

In this section we present the results for the preliminary models consisting of only subsets of the district municipalities in the study region. The posterior predictive distributions when including only the Cape Winelands and the City of Cape Town Metro with a simple spatial structure are shown in Figure 8.1 and Figure 8.2 respectively. The model clearly does an excellent job of replicating the pattern of daily new infections displayed by the observed data and the resulting intensity process seems to provide a good approximation for the real-world intensity processes in both district municipalities. This supports an initial impression that the intensity processes for these two district municipalities were similar in nature.

The next candidate model to be presented is one wherein the only district municipalities included are the Garden Route and Overberg. This district municipality pair was chosen due to a previously derived impression that their intensity processes could potentially be similar (as was the case with the previous pair of district municipalities). The results of this model are shown in Figure 8.3 and Figure 8.4. A key

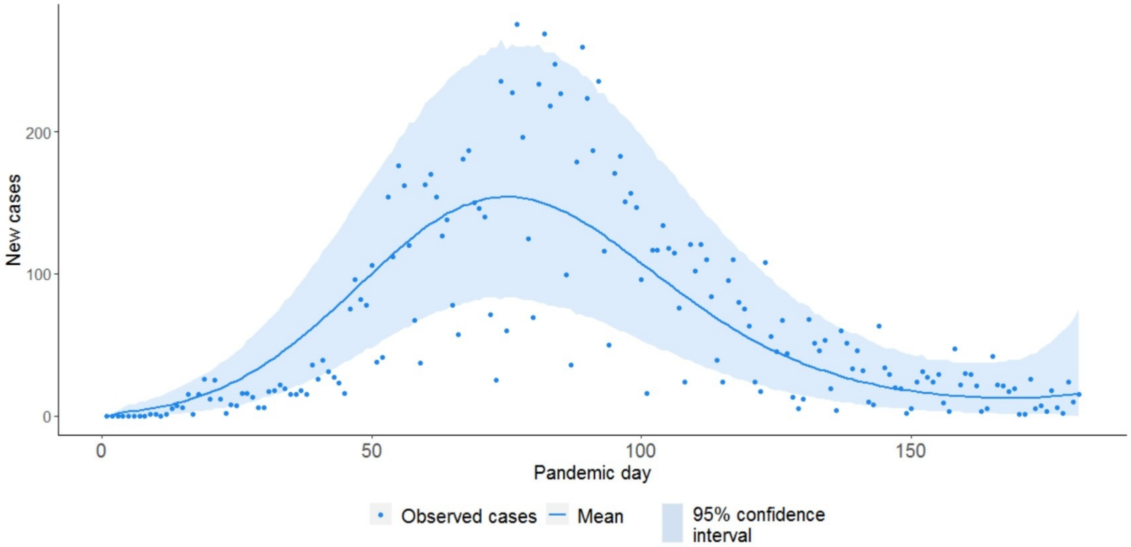


Figure 8.1: Posterior predictive distribution for Cape Winelands (preliminary model)

difference between these two district municipalities and the ones previously considered is the fact that there is a clear lag in initial cases early on in the pandemic, with the number of new cases initially being low or zero. After this initial lag there is a rapid increase in new daily cases followed by a rapid decline. This differs from what we observe for the Cape Winelands and City of Cape Town Metro which is an immediate and gradual increase in new daily cases followed by a gradual decrease. We note that the fitted intensity process replicates the clear lag in initial cases (compared to, for example, that of the model shown in Figure 8.1) and offers a good fit for the data for these two district municipalities. The 95% confidence interval for the new daily cases is quite wide, implying that in some instances the model vastly overestimates the number of new cases. This could perhaps be attributed to the previously mentioned rapidly increasing intensity process increasing too much and leading to far more cases than was actually observed. Nevertheless the average predicted cases align nearly perfectly with the observed data and thus we conclude that this model is satisfactory.

The third and final district municipality pair to be evaluated consists of the Central Karoo and West Coast. There are two reasons why these two district municipalities are the last to be evaluated. Firstly, the daily case data for these two district municipalities shows no clear pattern or “obvious” curvature, making it unclear which of the other district municipalities could potentially have a similar intensity process. Secondly, the West Coast and Central Karoo both exhibit a high degree of missing data, at 15% and 60% respectively. It became apparent through many modelling efforts that the presence of many missing values could greatly affect model performance, particularly when the beginning of the pandemic exhibited many missing observations. These missing observations appeared to mostly be due to the manner in which

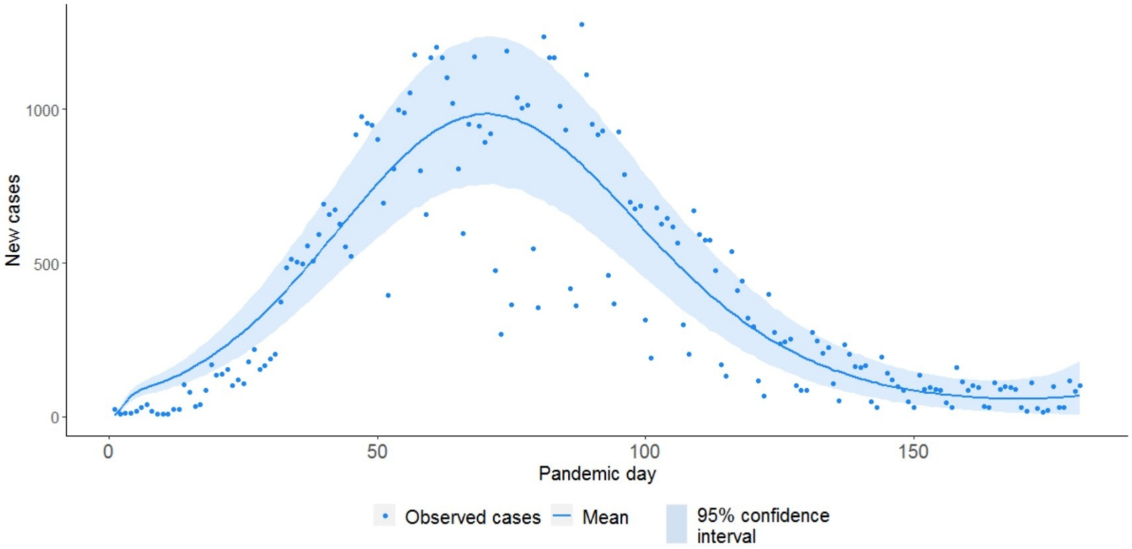


Figure 8.2: Posterior predictive distribution for City of Cape Town Metro (preliminary model)

the data was captured. Specifically, if there were no new cases on a given day there would be no data captured on that day at all. Using this assumption it was possible to replace the missing observations with plausible values. However it still proved impossible to formulate individual models for these two district municipalities that produce a satisfactory fit. Furthermore, a model comprised of both district municipalities fared no better. In an attempt to leverage some degree of analytical power from a district municipality with a clearer intensity process, two more models were formulated utilizing data for Overberg along with the West Coast and Central Karoo respectively. The posterior predictive distributions of these two models (for the West Coast and Central Karoo) are shown in Figure 8.5 and Figure 8.6. In Figure 8.5 we can immediately note that leveraging the Overberg case data allows us to fit a satisfactory model for the West Coast, with an intensity process that seems to replicate the patterns of new daily infections quite well. Unfortunately the results for the Central Karoo, shown in Figure 8.6, are not as promising. However it is rather difficult to imagine what a satisfactory model for this particular dataset would be, given that there is no clear or obvious curve that can be identified.

With all of these results taken into consideration we are able to draw several conclusions that are important to consider for the rest of our analysis. As discussed in a previous chapter, ABC algorithms aim to find the parameter values that replicate the observed data within some tolerance, indicated by ϵ, with the objective being to find the smallest ϵ value possible to improve the approximation. Given that the magnitude of cases within the City of Cape Town Metro is much larger than all other district municipalities in terms of scale, any satisfactory model will be able to achieve a lower tolerance level by minimising its error with regard to this district municipality in particular. This results in the modelling process exhibiting significant bias

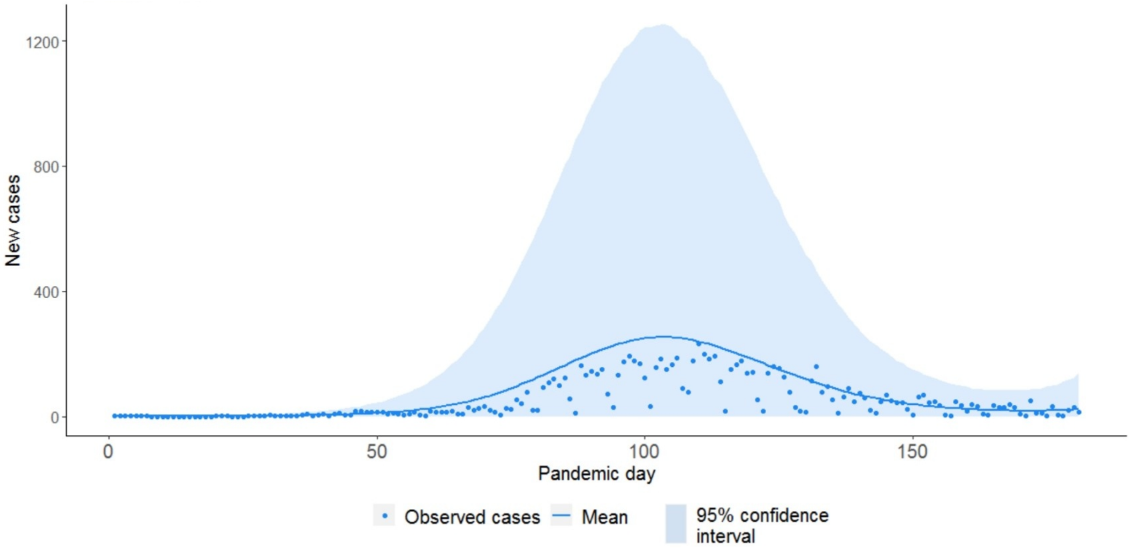


Figure 8.3: Posterior predictive distribution for the Garden Route (preliminary model)

in favor of this district municipality with regards to model goodness of fit. The preliminary modelling results also indicate that the intensity process for the City of Cape Town Metro and the Cape Winelands is similar in nature but different from that of all other district municipalities. Given the previously established bias in favor of the City of Cape Town Metro it is then clear that any model applied to all district municipalities simultaneously using one intensity process parameterization will favour this district municipality pair in terms of model goodness of fit. Attempts at including a separate intensity process for the other district municipalities have not yielded promising results, as the large number of parameters appear to cause issues [24].

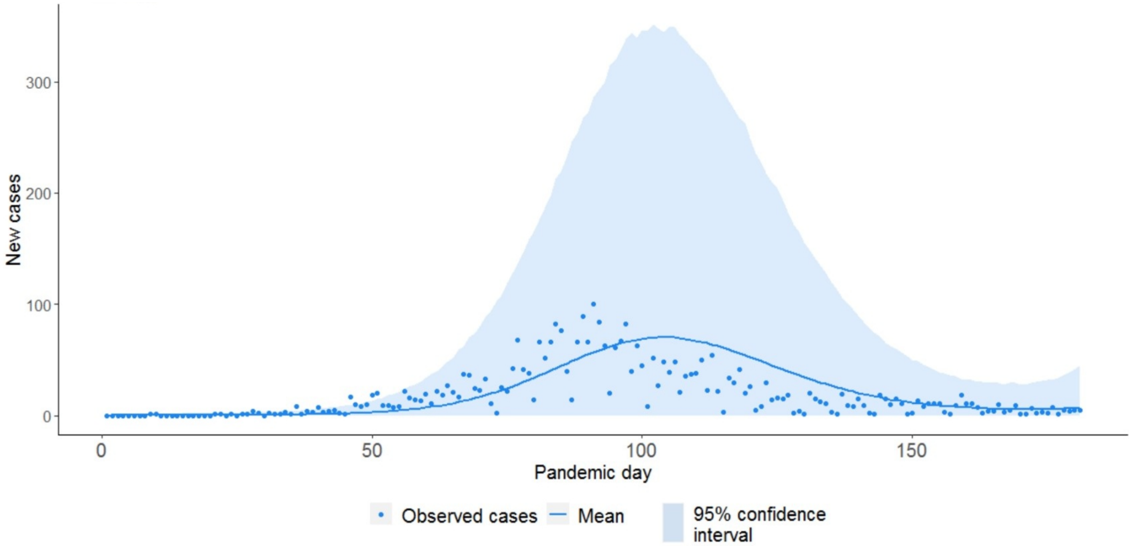


Figure 8.4: Posterior predictive distribution for Overberg (preliminary model)

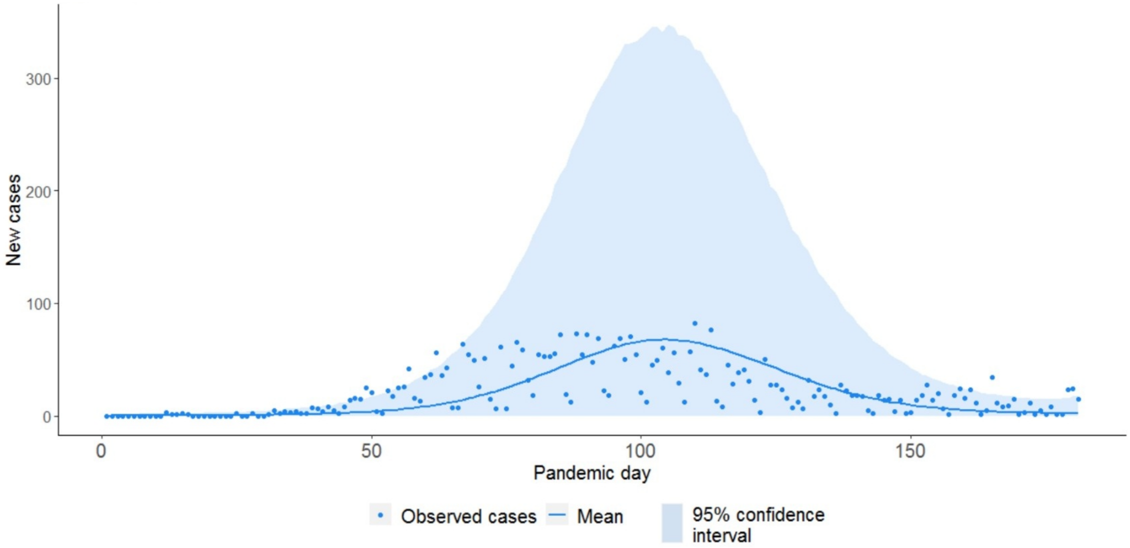


Figure 8.5: Posterior predictive distribution for the West Coast (preliminary model)

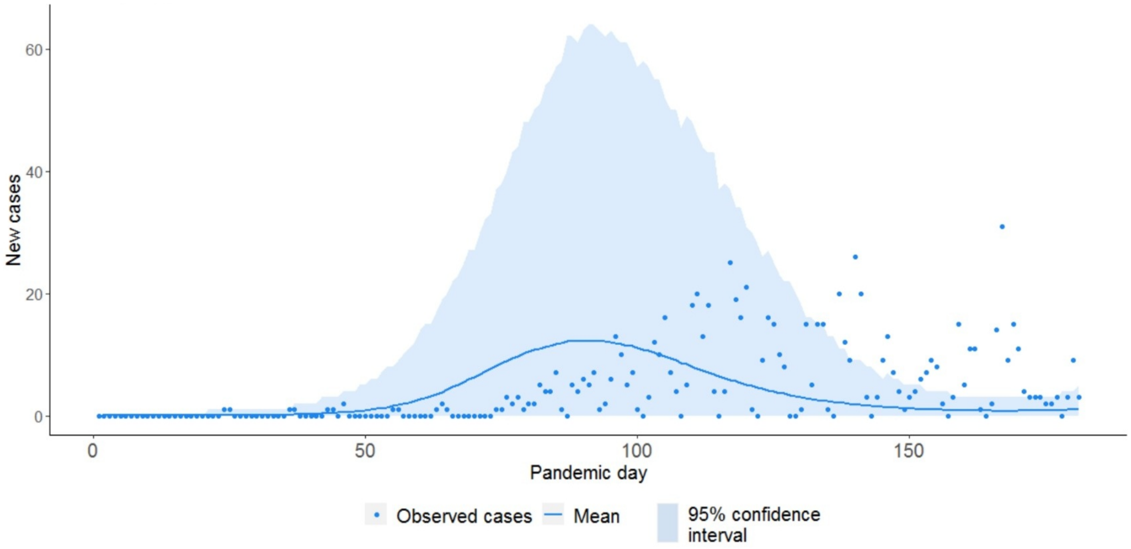


Figure 8.6: Posterior predictive distribution for the Central Karoo (preliminary model)

8.2 ABC rejection algorithm model

In this section we present the results of using the ABC rejection algorithm to fit the full model containing all district municipalities. The latent and infectious times are assumed to follow a gamma distribution.

A total of 100 samples were drawn from the posterior predictive distribution, with a final tolerance value of $\epsilon = 6,566.23$. Unfortunately, the posterior predictive distributions (all given in Appendix D) provide an unsatisfactory fit to the observed data for all district municipalities except the City of Cape Town Metro (shown in Figure 8.7).

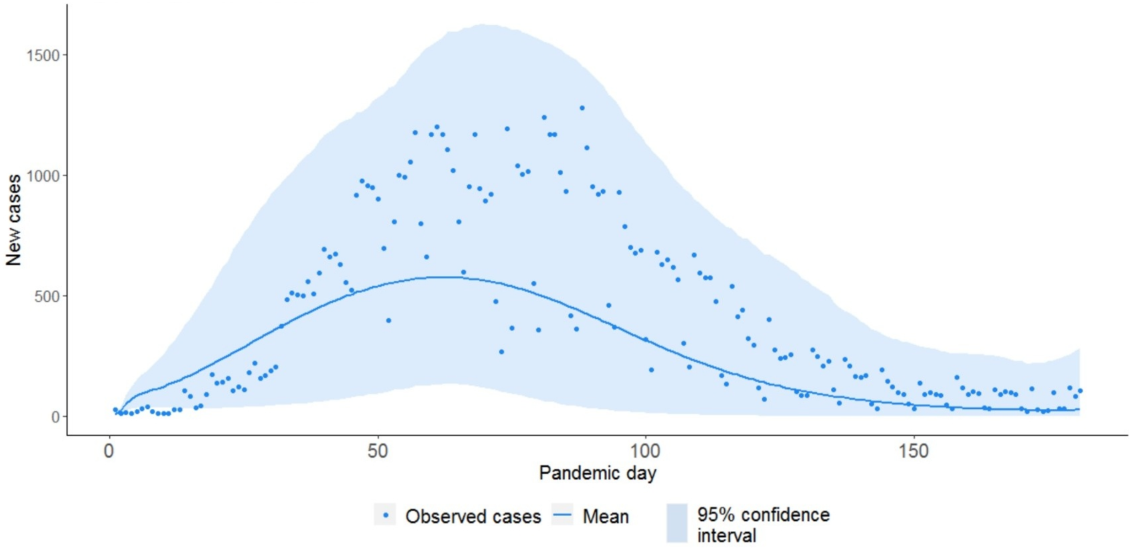


Figure 8.7: Posterior predictive distribution for City of Cape Town Metro (ABC rejection algorithm model)

There could be many reasons for the model's overall poor fit. We speculate that foremost of these is the relative simplicity of the basic ABC rejection algorithm and its many shortcomings when compared to later installments in the ABC family of algorithms (see Chapter 5 for full discussion). Another possible contributor is the previously mentioned bias in favour of the City of Cape Town Metro, which is the only district municipality that the posterior predictive distribution appears to be a good approximation of the observed data. Attempts were made at overcoming this issue by scaling the case data to a similar scale without success. It is unclear whether this scaling would truly reduce bias or instead introduce additional bias since district municipality populations will also need to be scaled in some way which would require the establishment of some baseline value for the scaled values. Furthermore such scaling requires significant rounding and approximations to be made to the data which are most likely not ideal to data that is already deemed noisy and heavily affected by sampling bias.

8.3 SMC ABC algorithm model

The model discussed in this section utilises the exact same parameterization as the previous model but is fitted using the SMC ABC algorithm. The final tolerance value for this model was $\epsilon = 2,888.96$. The posterior predictive distributions for this model are shown in Figures 8.8 - 8.13.

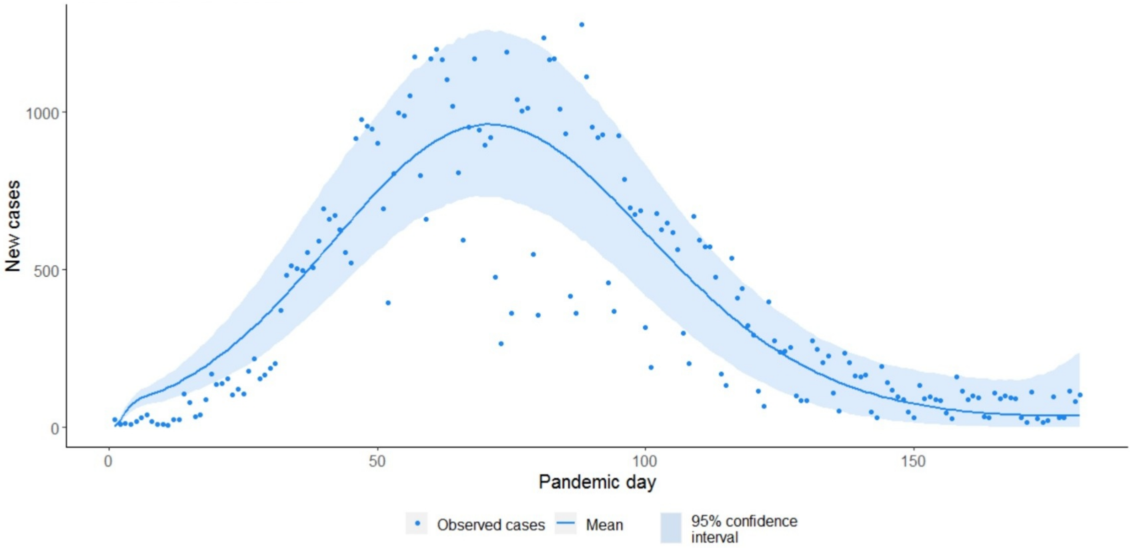


Figure 8.8: Posterior predictive distribution for City of Cape Town Metro (SMC ABC algorithm model)

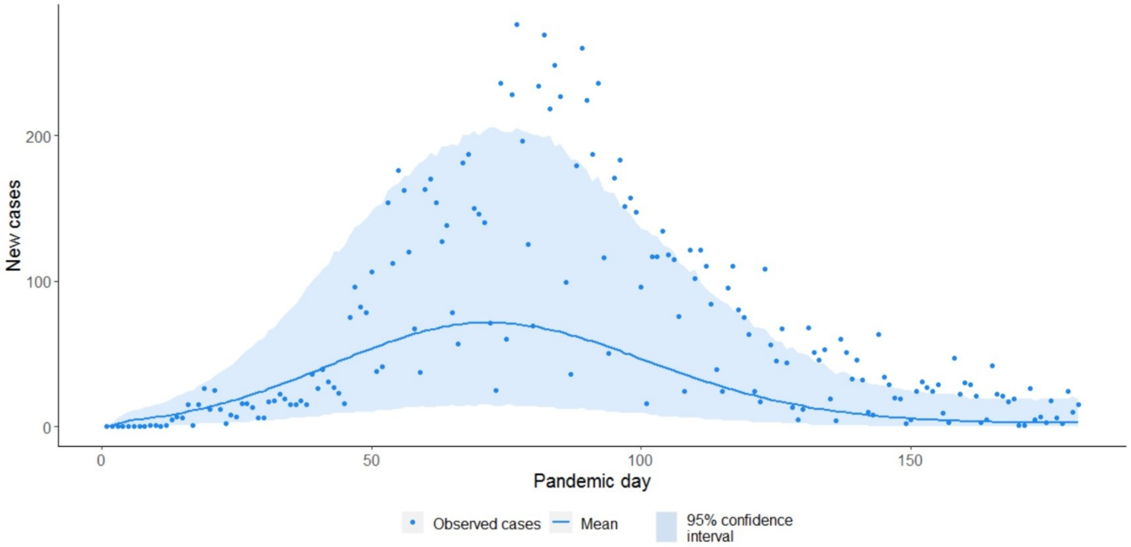


Figure 8.9: Posterior predictive distribution for Cape Winelands (SMC ABC algorithm model)

Through visual inspection alone it is abundantly clear that this model produces posterior predictive distributions that serve as much better approximations to the observed case data compared to those of the ABC rejection algorithm model. Furthermore, the final tolerance level is less than 50% of the threshold that was achieved for the ABC rejection algorithm model.

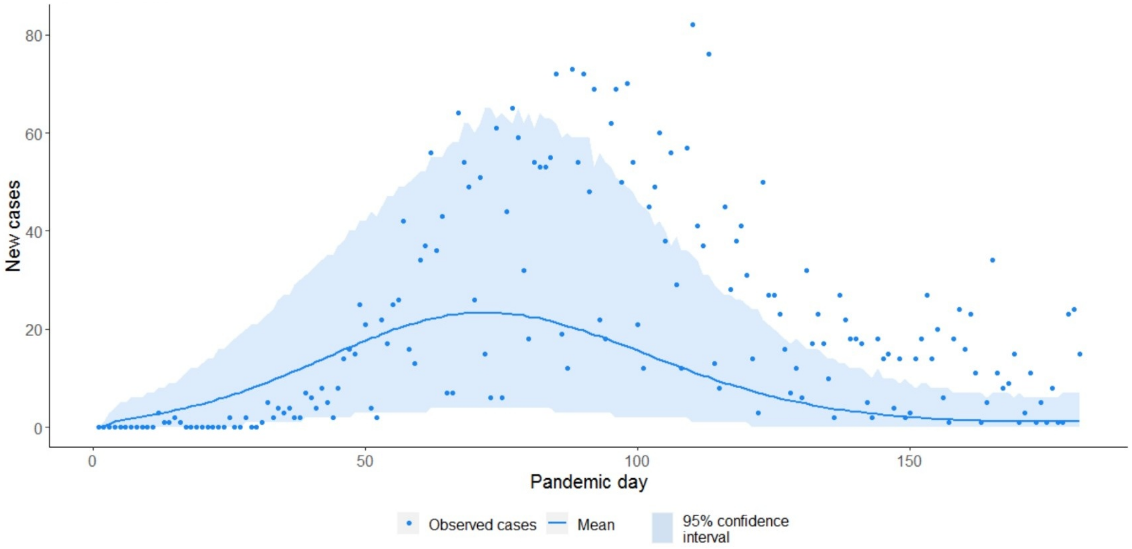


Figure 8.10: Posterior predictive distribution for West Coast (SMC ABC algorithm model)

For the City of Cape Town Metro (posterior predictive distribution shown in Figure 8.8) the model has an improved fit, with significantly smaller confidence intervals than those derived previously. For the Cape Winelands and West Coast (posterior predictive distributions shown in Figure 8.9 and 8.10 respectively) the model provides a good fit but slightly underestimates the peak of the infection curve. A similar observation can be made for Overberg (posterior predictive shown in Figure 8.11), where the peak of the infection curve is significantly lower than that of the observed cases. For the Central Karoo (posterior predictive shown in Figure 8.12) we note that while the overall shape and height of the infection curve appears to be a satisfactory approximation to the observed data, the peak of infections is predicted sooner than what was truly experienced. For the Garden Route (posterior predictive shown in Figure 8.13) we see that not only is the peak of the infection curve significantly lower than that of the observed cases but the peak of infections is also predicted sooner than truly observed.

All of these observations corroborate with earlier observations regarding the true underlying intensity process for the district municipalities. Specifically that the model will prioritise approximating the intensity process of the City of Cape Town Metro, which has a gradual intensity process, while producing a slightly worse fit for other district municipalities that experienced an initial lag in cases.

Other important parameters within this model include the spatial autocorrelation parameters for the mobile network and Facebook data. The distribution of these parameters is shown in Figure 8.14, with a table of summary statistics given in Table 8.1. We note that the distribution of parameter values is more spread out for the mobile network data while that of the Facebook data is more clustered around very low values. This suggests that the spatial weight matrix for the mobile network data implies a larger degree of spatial autocorrelation of the case data between the various district municipalities which is not identified by the Facebook data. This corresponds to results derived in Chapter 3, specifically that the mobile network data spatial weight matrix appears more adept at identifying strong spatial associations over short and medium distances, compared to the Facebook data spatial weight matrix which identifies spatial associations over very long distances and between hubs of human activity such as the four largest cities in the country (all but one of which are removed from the study region).

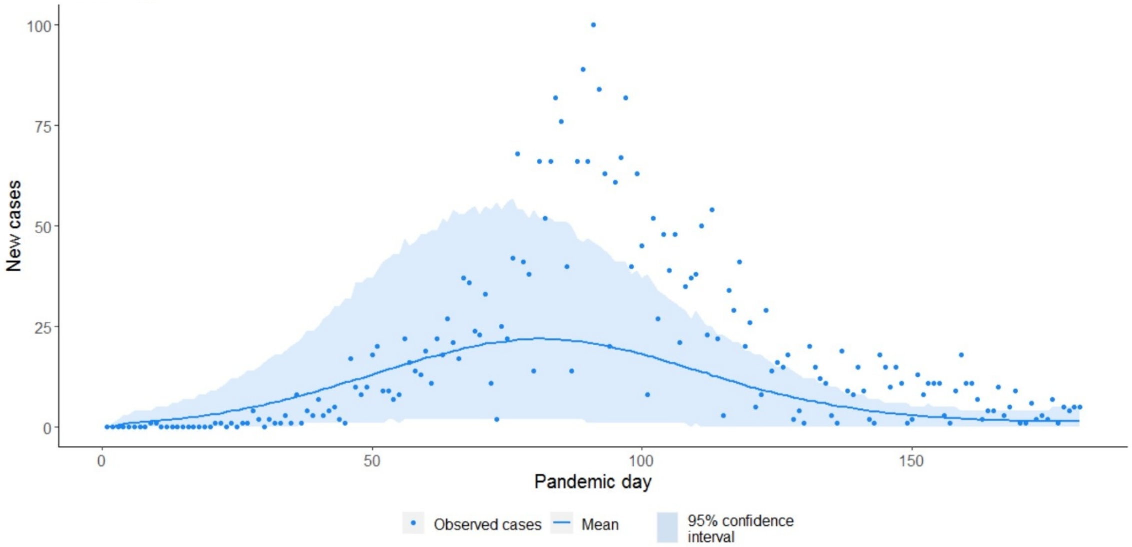


Figure 8.11: Posterior predictive distribution for Overberg (SMC ABC algorithm model)

Note that the minimum spatial autocorrelation for both spatial weight matrices is zero (rounded to 3 decimal places). The mean values for the two spatial autocorrelation parameters is 0.18 and 0.33 for the Facebook and mobile network data respectively. This indicates that while the mobile network spatial weight matrix identifies more spatial autocorrelation of cases than the Facebook one, the spatial autocorrelation is still relatively low. This is expected given that the case data spans the initial length of time wherein South Africa was placed within its most stringent lockdown (see Table 1.1). We note that the probability of the spatial autocorrelation being larger than 0.25 for a given simulation is 0.66 for the mobile network data compared to only 0.23 for the Facebook data. For spatial autocorrelation parameter values above 0.5 we see there is a probability of 0.15 for the mobile network data and 0 for the Facebook data. The results here seem to indicate that utilising the Facebook data as the only indication of spatial association would underestimate the level of spatial autocorrelation within the study region.

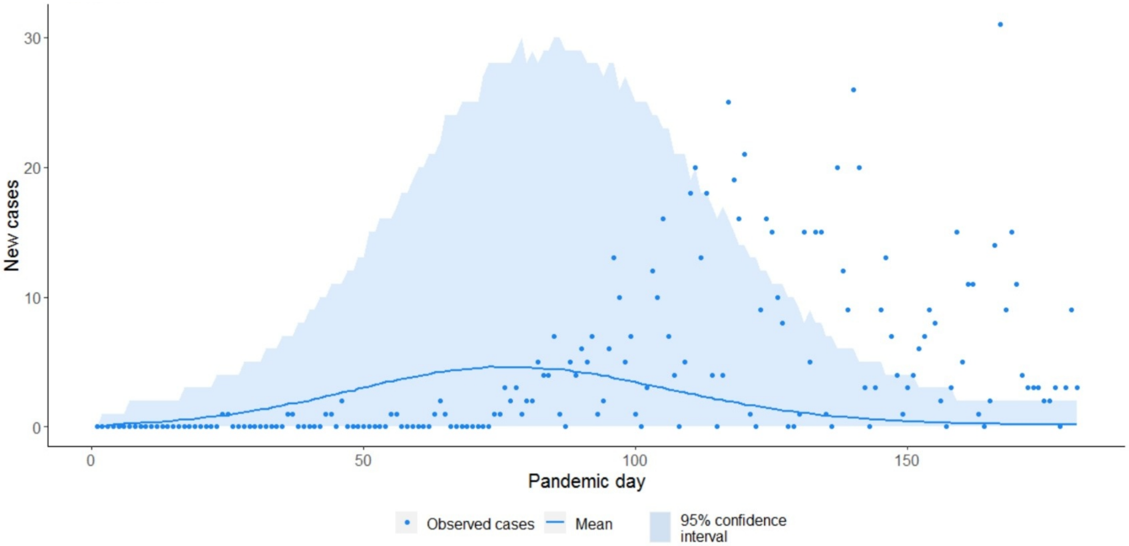


Figure 8.12: Posterior predictive distribution for Central Karoo (SMC ABC algorithm model)

Table 8.1: SMC ABC algorithm model - Spatial autocorrelation parameters summary statistics

Data	Min	Max	Mean	Standard deviation	95% CI	$P(\rho > 0.25)$	$P(\rho > 0.50)$
Facebook	0	0.49	0.18	0.11	[0.04 ; 0.44]	0.23	0
Mobile network	0	0.68	0.33	0.15	[0.06 ; 0.62]	0.66	0.15

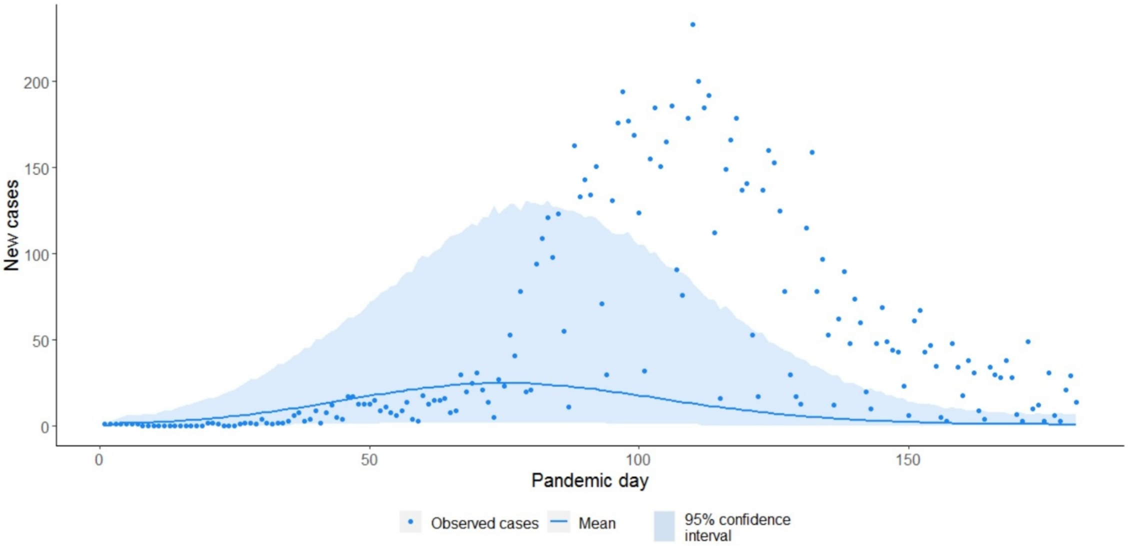


Figure 8.13: Posterior predictive distribution for the Garden Route (SMC ABC algorithm model)

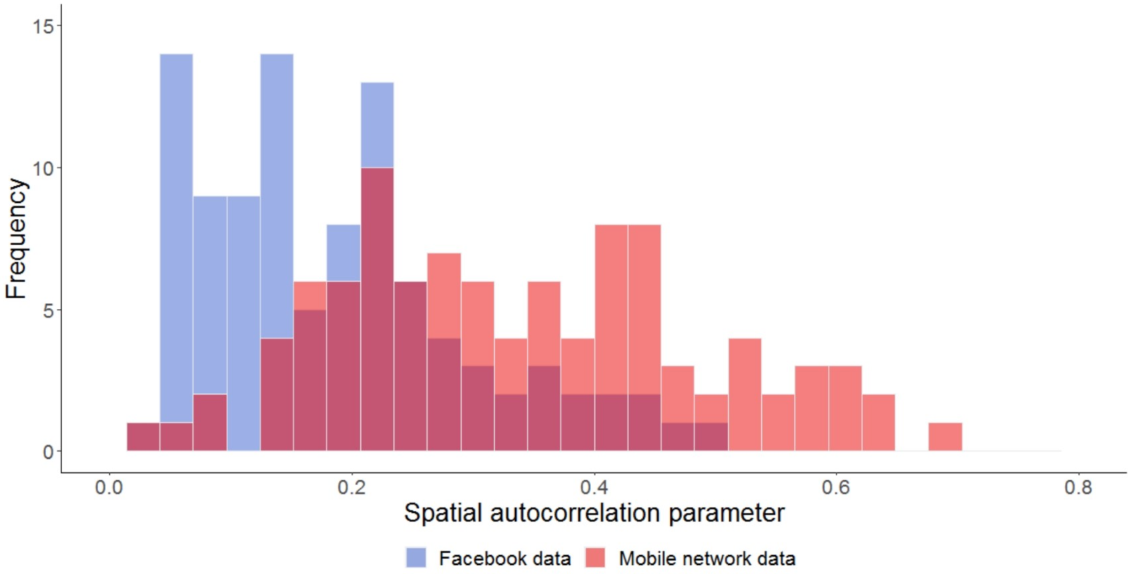


Figure 8.14: Spatial autocorrelation parameter distribution (SMC ABC algorithm model)

8.4 Empirically adjusted reproductive number

In every model presented thus far, the latent and infectious time probability distributions were modelled as random variables following a gamma random variable. This diverges from the usual assumption that is made in compartmental epidemiological modelling, which is that these probabilities follow an exponential distribution [72, 98]. While the exponential parameterization is more simplistic in nature and easier to interpret, the gamma parameterization allows for a larger degree of flexibility when modelling these probabilities. One potential concern for the gamma parameterization however is that the basic reproductive number (discussed in Section 4.2, indicated as R_0) is not unambiguously defined for such models [24]. An equivalent measure that can be interpreted in a similar fashion is the empirically adjusted reproductive number (discussed in Section 4.4, indicated as $R^{(EA)}(t)$) [24]. In this section we present the $R^{(EA)}(t)$ values calculated for two models utilising the gamma and exponential latent and infectious time probability parameterizations respectively.

The first model to be compared is identical to the SMC ABC algorithm model in all facets aside from the removal of the Facebook data spatial weight matrix. This spatial weight matrix was removed due to its relatively low level of importance in the SMC ABC algorithm model (see Table 8.1 and Figure 8.14) as well as the calculation of $R^{(EA)}(t)$ (also implemented in the `ABSEIR` package) not being implemented for more than one spatial weight matrices at present. The second model to be compared is identical to the previously mentioned model except for the latent and infectious times being modelled as exponential rather than gamma random variables. The posterior predictive distributions for this model is included in Appendix E.

Before determining the empirically adjusted reproductive number for these two models, we fit the models to the data and utilise Bayes factors to determine which model provides a better fit to the data. The Bayes factors the two models considered here are given in Table 8.2. We see that the Bayes factor for the gamma model with respect to the exponential model is larger than 1 and thus there is more evidence in favor of the gamma model. However the Bayes factor is not particularly large, indicating that while the gamma model is preferred, the difference in performance is not highly significant (see Table 5.1 for Bayes factor interpretation).

Table 8.2: Gamma vs exponential Bayes factor (district municipality level)

	Gamma	Exponential
Gamma	1	2.152
Exponential	0.465	1

Despite the empirically adjusted reproductive number differing from the basic reproductive number in several ways, we can interpret it in a similar fashion. The empirically adjusted reproductive number estimates the expected number of secondary infections caused by an infectious individual. The threshold value and its interpretation is also identical. If the empirically adjusted reproductive number is larger than 1 then we can expect the disease being studied to spread further amongst the study population and the number of new daily cases to increase in the short-term. This is due to infectious individuals infecting at least one other susceptible individual before recovering. If the empirically adjusted reproductive number is below 1 however, then we can expect the disease to cease spreading in the short-term due to not every infectious individual infecting another susceptible individual. The empirically adjusted reproductive numbers over the study period for each district municipality are shown in Figure 8.15 - 8.20.

The 95% confidence interval for the empirically adjusted reproductive number for all district municipalities is strictly below 1 for the model with gamma transition probabilities. For the exponential parameterization, we note that the 95% confidence interval is strictly below 1 for all district municipalities except the Cape Winelands and the City of Cape Town Metro, shown in Figures 8.15 and 8.17 respectively. However the mean empirically adjusted reproductive number for these two district municipalities is still very low. In general, it appears to be the case that the estimated empirically adjusted reproductive number is higher for the exponential model than for the gamma model.

These results all indicate that the first wave of COVID-19 infections in the Western Cape province died out over the study period, which was indeed the case. Furthermore the results indicate that the disease did not enjoy a particularly rapid spread during the first wave, with the empirically adjusted reproductive number being very low on average. These results indicate that the initially highly stringent lockdown measures that were put into place during the first wave were highly successful in limiting the spread of the disease.

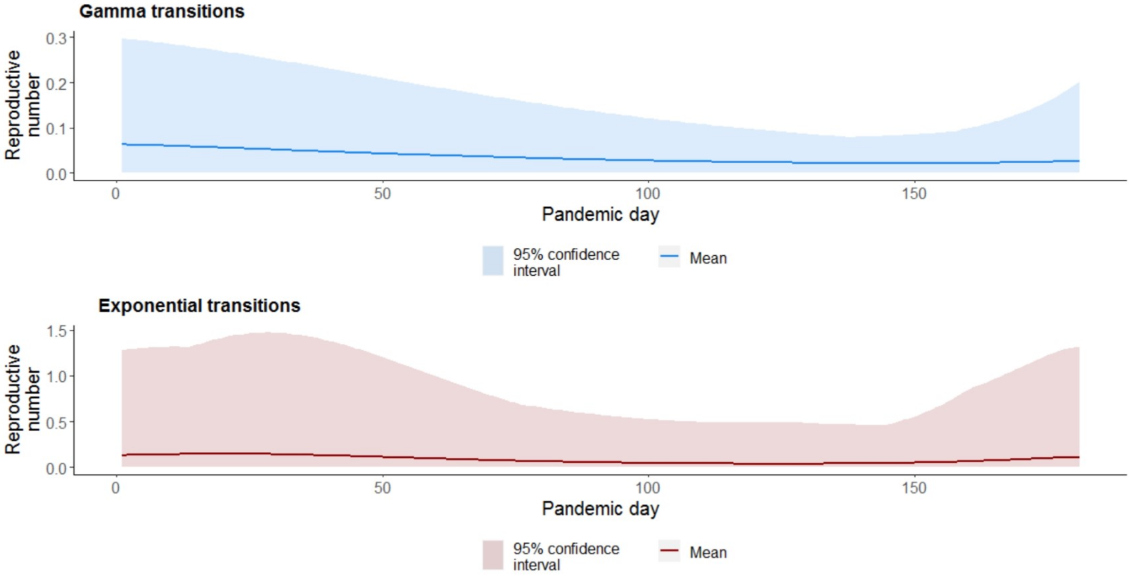


Figure 8.15: Empirically adjusted reproductive number for Cape Winelands

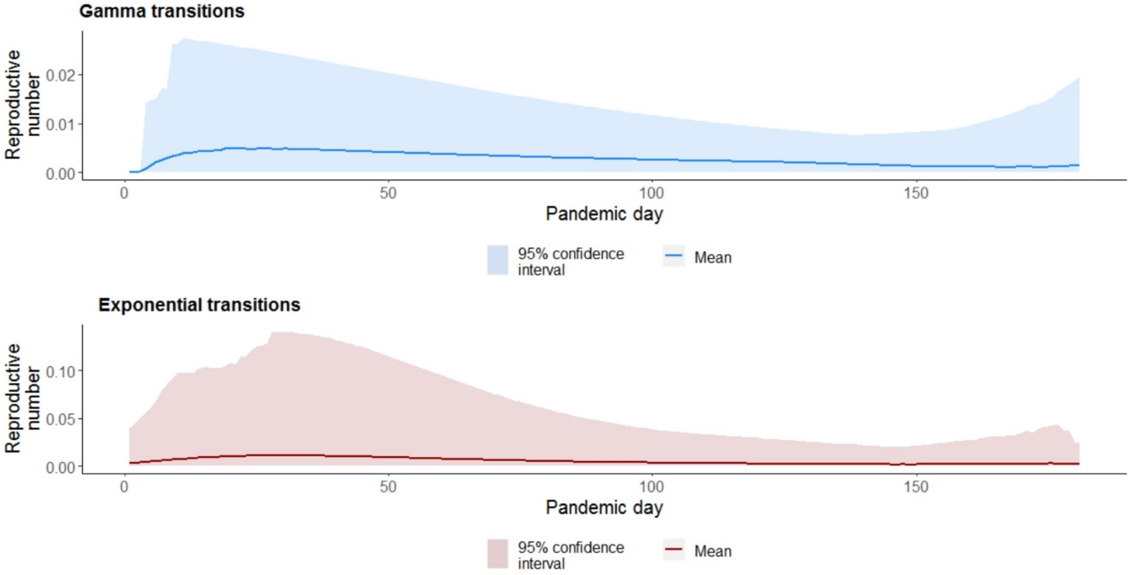


Figure 8.16: Empirically adjusted reproductive number for Central Karoo

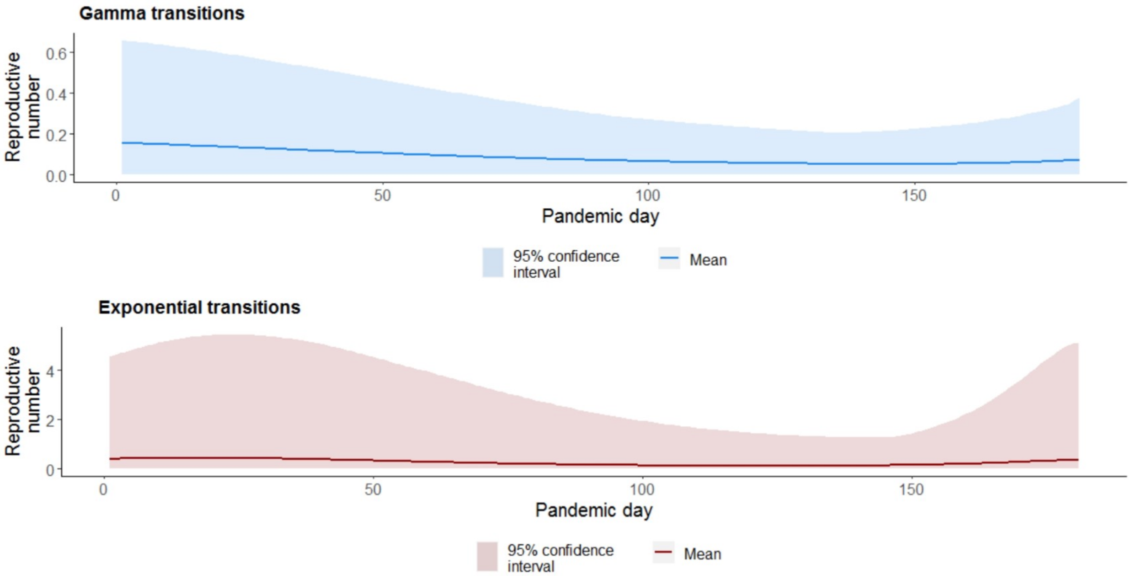


Figure 8.17: Empirically adjusted reproductive number for City of Cape Town Metro

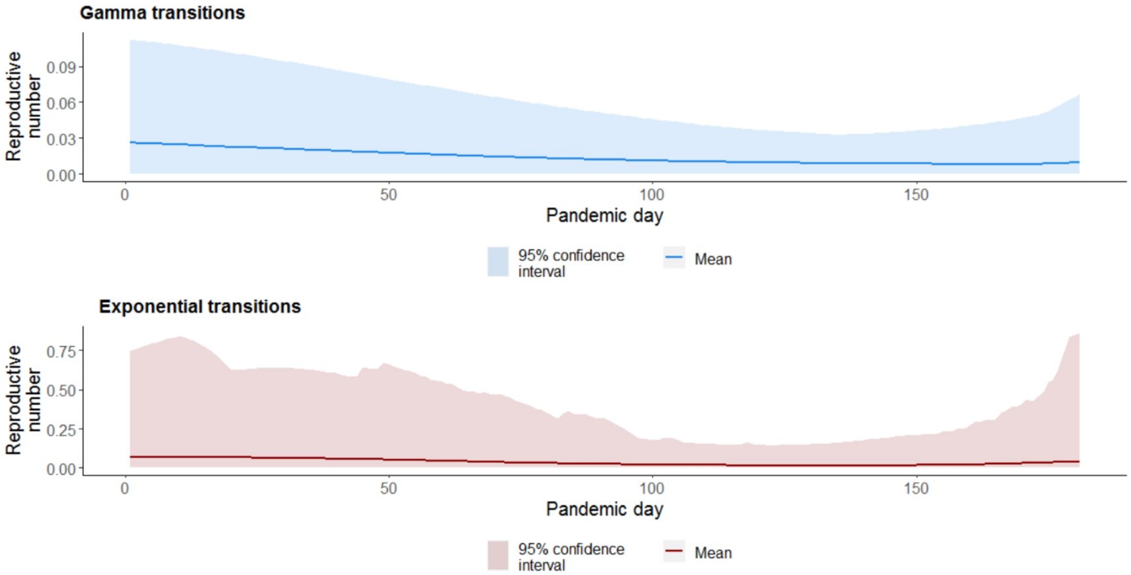


Figure 8.18: Empirically adjusted reproductive number for Garden Route

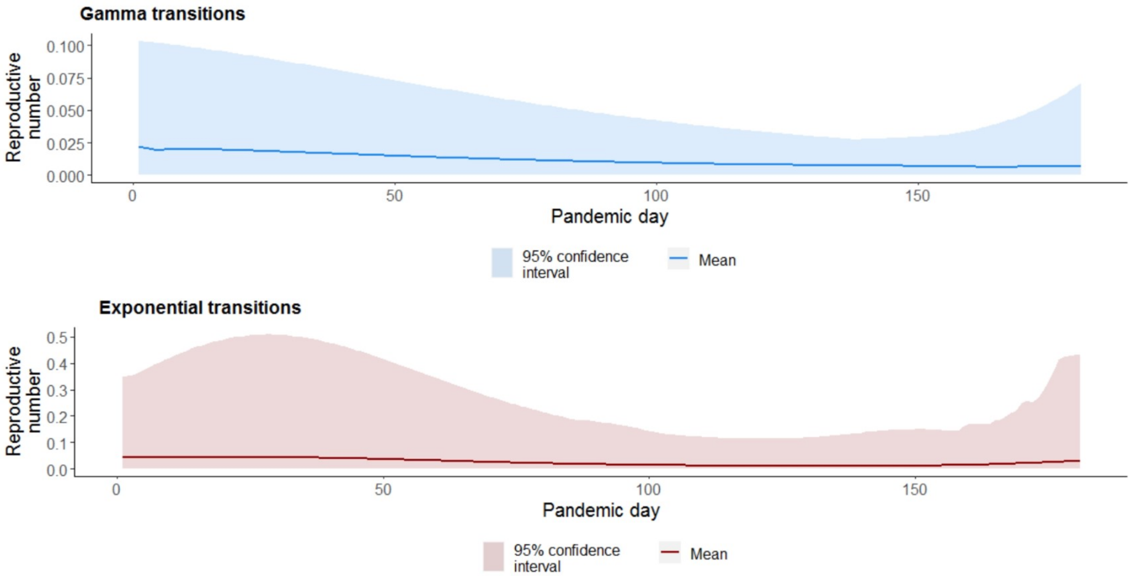


Figure 8.19: Empirically adjusted reproductive number for Overberg

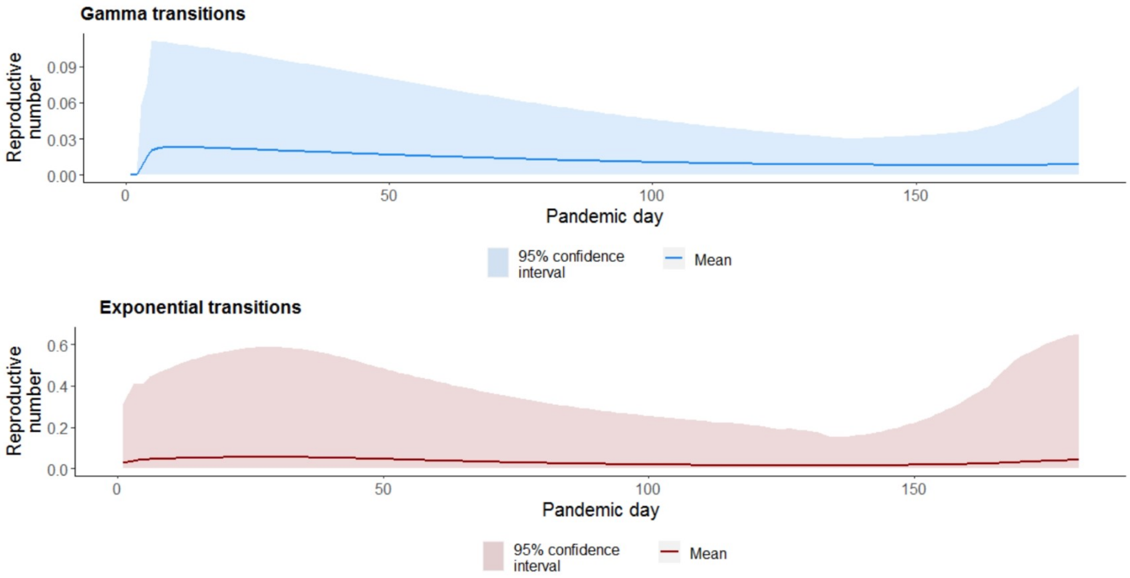


Figure 8.20: Empirically adjusted reproductive number for West Coast

8.5 Single location analysis

We next consider modelling the spread of the disease at provincial level (i.e. aggregating the case data across all district municipalities to produce only one spatial region). We thus disregard all spatial associations in this analysis and model the spread of the disease within the study region as a whole. We again consider the use of two models with different transition probability distributions, gamma and exponential, respectively. After fitting these two models to the province-level data, we derive the Bayes factors as given in Table 8.3. Once again we note that there is more evidence in favour of the gamma model, however the evidence is even less substantial than in the previous section.

Table 8.3: Gamma vs exponential Bayes factor (provincial level)

	Gamma	Exponential
Gamma	1	1.566
Exponential	0.639	1

The empirically adjusted reproductive number for both models is shown in Figure 8.21. For the exponential model, we note that when the case data is aggregated to a provincial level the estimated empirically adjusted reproductive number is much larger than at district municipality level. For the gamma model, we note that the empirically adjusted reproductive number is still very low, well below 1, and has unusually narrow 95% confidence intervals. The posterior predictive distributions for these two models as well as the basic reproductive number distribution for the exponential model are included in Appendix F.

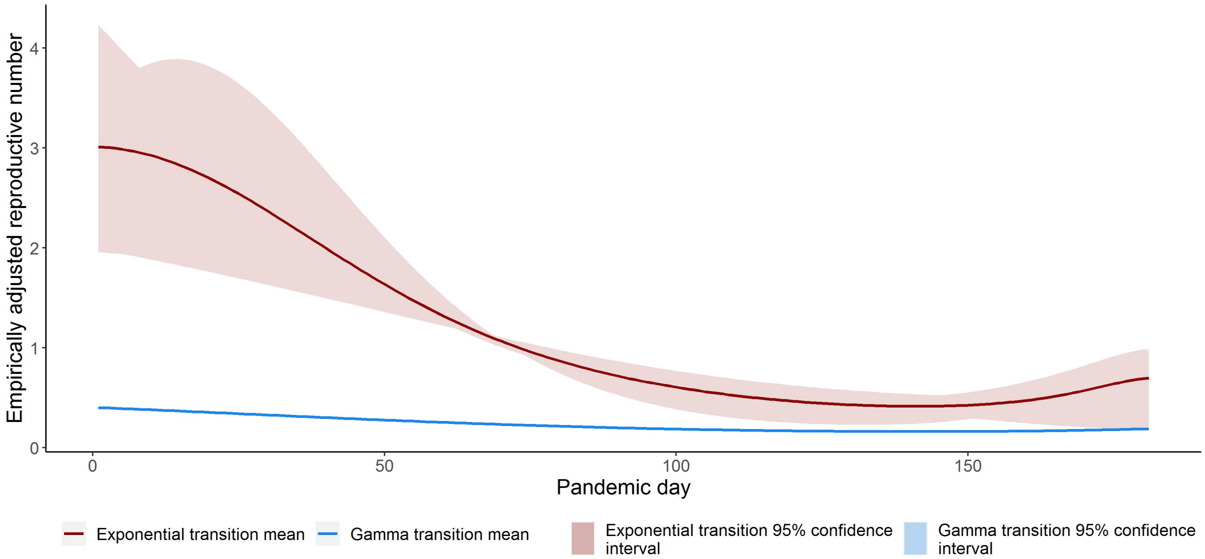


Figure 8.21: Empirically adjusted reproductive number for entire study region

8.6 Data source comparison

In this section we compare the two sources of COVID-19 case data obtained for this mini-dissertation to determine whether they are identical or whether significant changes can be observed. All models discussed thus far were fitted utilising the freely available NICD data. This data was chosen to be used first due to no additional steps having been required to obtain it, while the press release data required web scraping and data cleaning. After performing some simple analysis to check for discrepancies in the data, we fit the SMC algorithm model to the press release data and determine whether any significant differences in model performance and inference are present. The COVID-19 case data for each data source as well as the fitted SMC ABC algorithm model is shown in Figure 8.22 - Figure 8.27 for each district municipality in the study region.

In order to empirically compare the two datasets, we determine the cumulative number of cases for each date. We then resolve to test their similarity through the use of the Kolmogorov-Smirnov (KS) test. We thus consider the case data for each district municipality as an empirical cumulative density function and then determine whether the density functions are statistically different between the two datasets. The results of the KS test are shown in Table 8.4.

Table 8.4: Kolmogorov-Smirnov test for press release and NICD COVID-19 data

District municipality	**p-value**	**Test conclusion**	**Interpretation**
Cape Winelands	0.227	Do not reject	Similar distributions
Central Karoo	<0.001	Reject	Different distributions
City of Cape Town Metro	0.357	Do not reject	Similar distributions
Garden Route	0.011	Reject	Different distributions
Overberg	0.005	Reject	Different distributions
West Coast	0.003	Reject	Different distributions

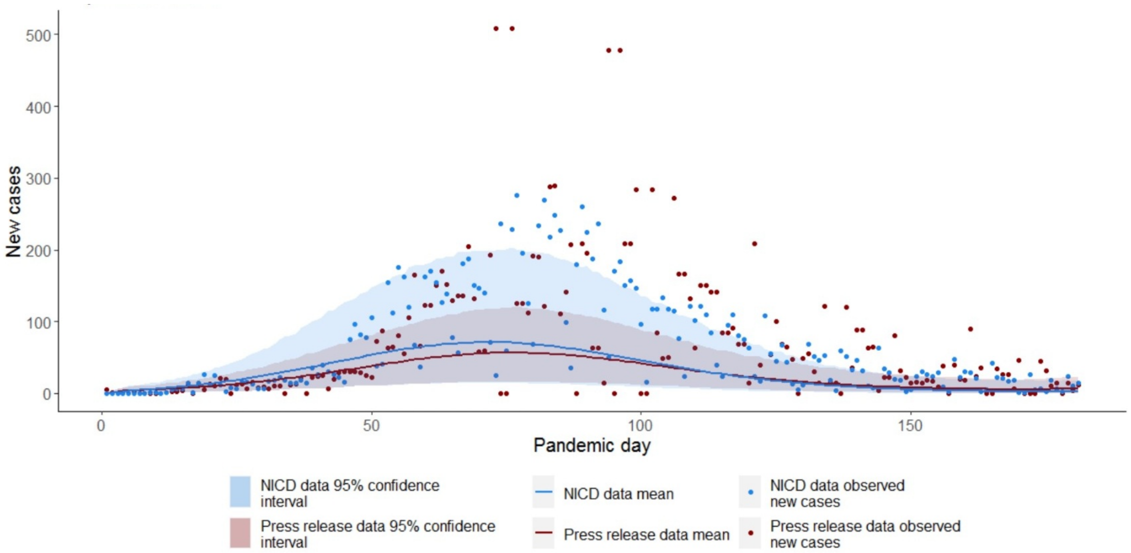


Figure 8.22: COVID-19 case data for two sources with model posterior predictive distribution (Cape Winelands)

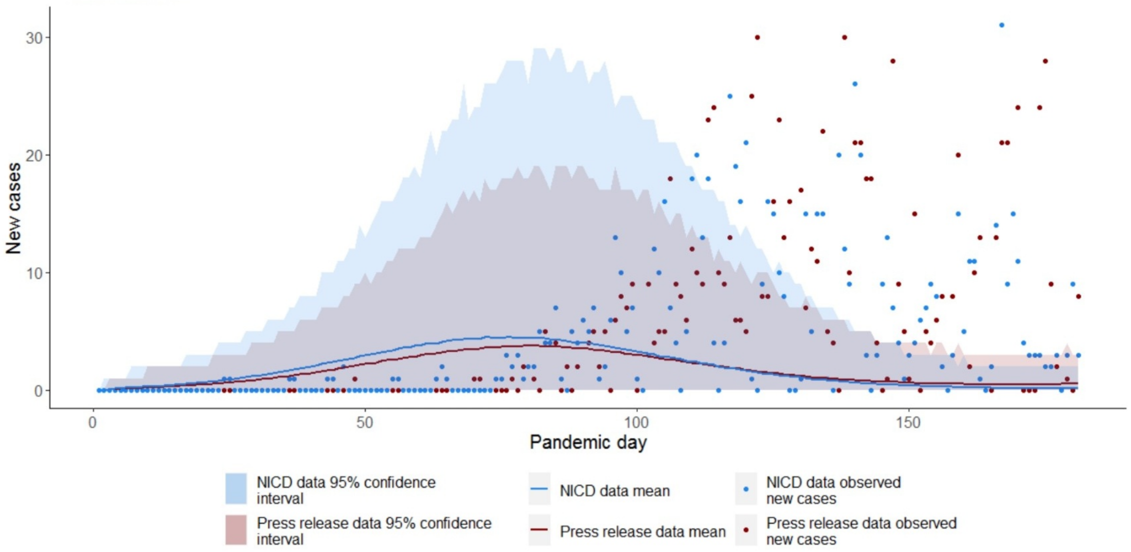


Figure 8.23: COVID-19 case data for two sources with model posterior predictive distribution (Central Karoo)

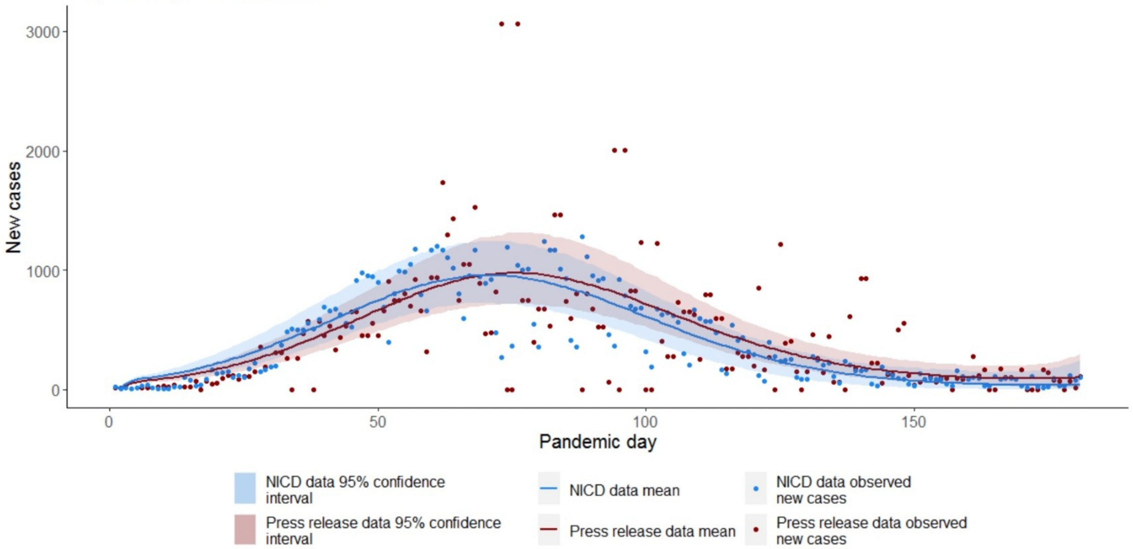


Figure 8.24: COVID-19 case data for two sources with model posterior predictive distribution (City of Cape Town Metro)

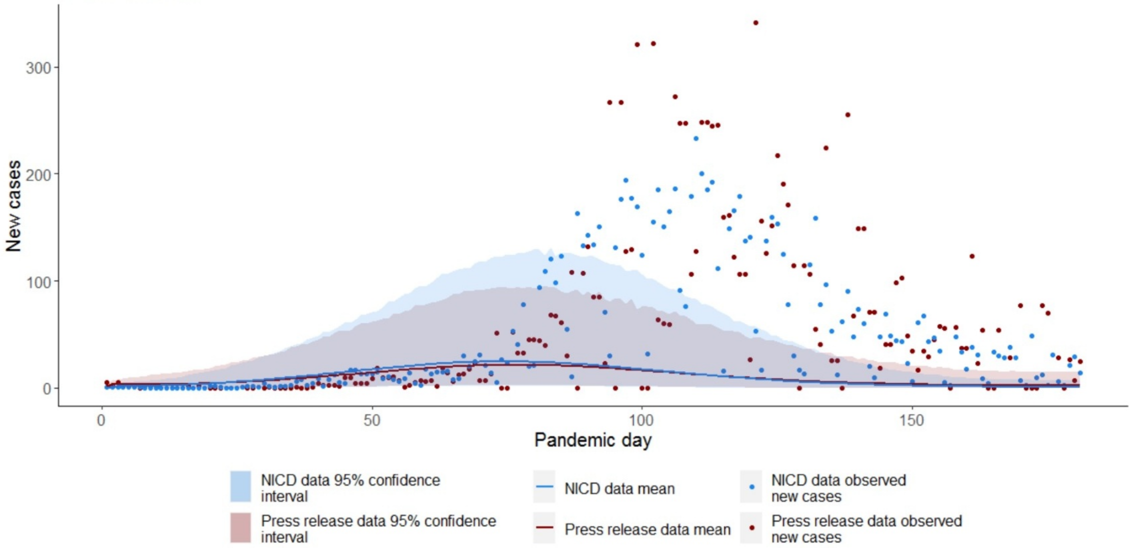


Figure 8.25: COVID-19 case data for two sources with model posterior predictive distribution (Garden Route)

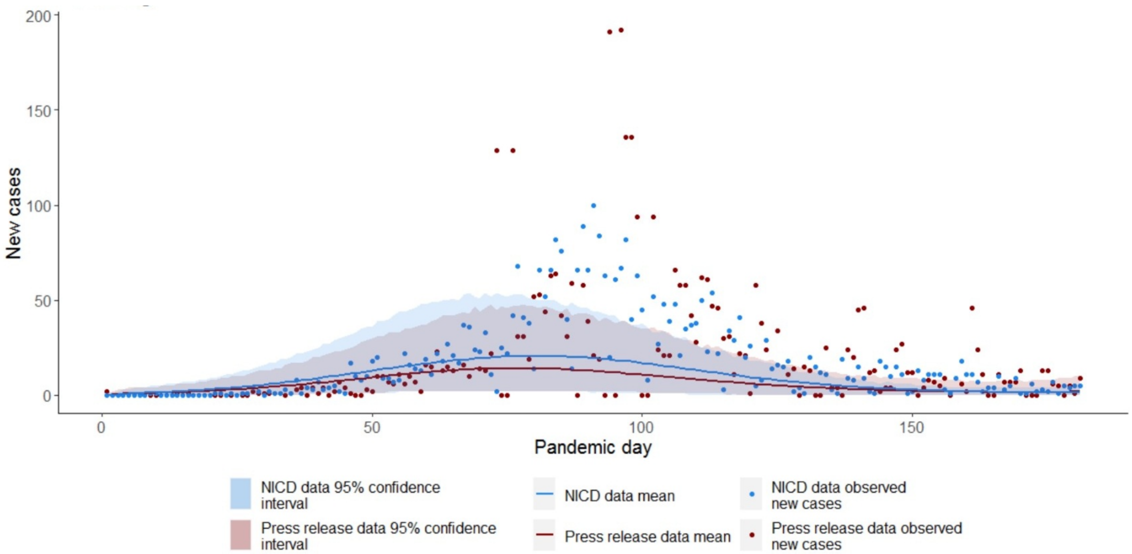


Figure 8.26: COVID-19 case data for two sources with model posterior predictive distribution (Overberg)

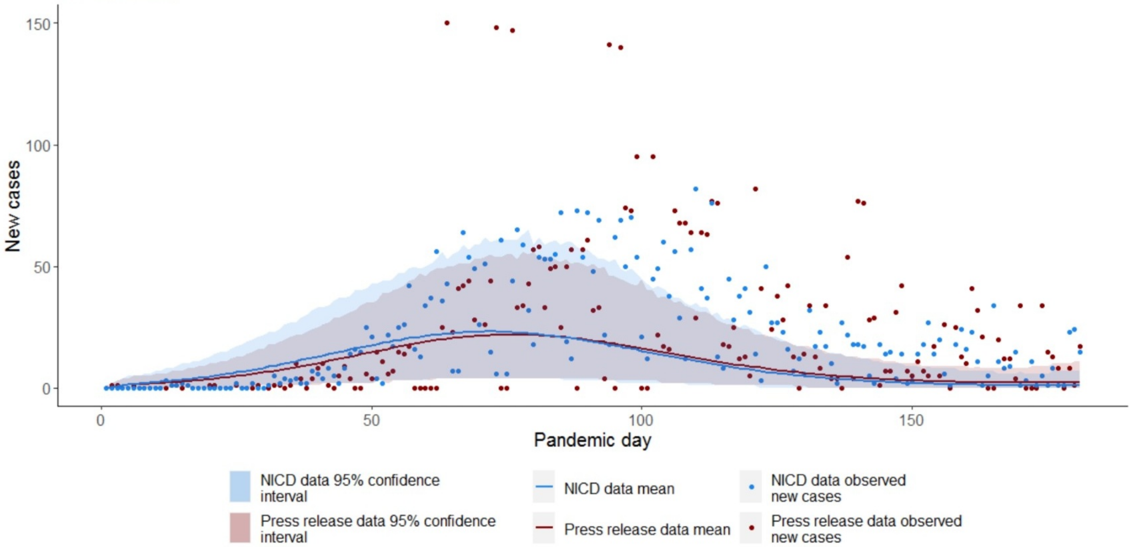


Figure 8.27: COVID-19 case data for two sources with model posterior predictive distribution (West Coast)

8.7 Neural network summary statistic model

In this section we present the results of fitting two autoencoders to the simulated pandemic data. The training and validation losses are shown in Table 8.5.

Table 8.5: Autoencoder training and validation loss

Autoencoder model	Architecture (hidden nodes)	Training loss	Validation loss
ANN	30	0.4351	0.4377
DNN	128, 64, 32, 64, 128	0.4324	0.4353

Unfortunately it was not possible to formulate an autoencoder model that was capable of learning a lower-dimensionality representation of the simulated pandemic data, as can clearly be seen from the relatively high loss function values in Table 8.5. Other architectures for either autoencoder did not show any improvement in the loss function. When opting for less data reduction (for example a data reduction of only 2 when using 500 hidden neurons) the model performance did not improve either.

After using the autoencoders on a second simulated set of pandemics as well as the observed data it was determined that the autoencoders were not able to identify those pandemics that were most similar to the observed data, retaining even those simulations with the least similarity to the observed data. In the next chapter, we discuss possible reasons for the relatively poor performance of the autoencoders.

Chapter 9

Discussion

In this chapter we discuss the results of the analysis conducted in this mini-dissertation. Several different types of statistical analysis and principles were included. These include spatial statistics, compartmental epidemiological modelling, approximate Bayesian computation as well as artificial neural networks. We discuss the results of the analysis performed for each of these in turn.

Spatial associations between the district municipalities of the study region were included using two sources of population mobility data. Data was obtained from a local mobile network provider as well as from Facebook. The spatial associations were captured using various spatial weight matrices as discussed in Section 3.2. When these spatial weight matrices were included within spatial SEIR models it revealed that the spatial autocorrelation of COVID-19 infections across the study region was relatively low. The average spatial autocorrelation parameter for the mobile network and Facebook data was 0.33 and 0.18 respectively (see Table 8.1). This result is unsurprising given that the study period spans South Africa's stringent level 5 lockdown period (see Table 1.1) where most forms of travel and non-essential human interaction was prohibited. These estimates thus confirm that, due almost certainly to a lack of human mobility, there was very correlation between the occurrences of COVID-19 infections between the various district municipalities.

We find it worth noting however that despite both being low on average, the mobile network spatial autocorrelation parameter is almost twice that of the Facebook data (on average). Furthermore, the probability of observing high spatial autocorrelation parameter values is much larger for the mobile network spatial weight matrix. The probability of observing spatial autocorrelation parameters larger than 0.25 was estimated to be 0.23 and 0.66 for the Facebook and mobile network data respectively. For parameter values larger than 0.5 these probabilities were 0 and 0.15 respectively. This illustrates that if one were to use only Facebook spatial weight matrix (which is constructed using only easily available data from the Internet) without the additional incorporation of the mobile network spatial weight matrix (which is constructed using mobile device data that is harder to obtain) there is a serious risk of underestimating the level of spatial autocorrelation present between the COVID-19 cases in the district municipalities within the study region. This illustrates the unfortunate fact that highly censored but freely available data is heavily outclassed by more detailed data that is generally harder to obtain.

These spatial weight matrices were initially calculated at daily time interval using the equations previously given in Section 3.2. These matrices were then averaged to yield single spatial weight matrices for the entire study period for each source of mobility data that were included in the model. An opportunity for future research would be to instead include time-varying spatial weight matrices in order to determine whether there are significant changes in the spatial autocorrelation parameters over time. This could be particularly insightful in light of South Africa's changing lockdown levels.

The spatial SEIR model as proposed by Brown et al. [24] was utilised throughout for the compartmental modelling in this mini-dissertation. A series of initial modelling efforts focused on select pairs of district municipalities revealed that certain district municipalities exhibited similar intensity processes but that these were not homogeneous across all district municipalities. Upon further investigation it was identified that there is a source of bias within the model fitting procedure in favour of spatial regions with observations of larger scale. Future research may be conducted in order to develop a weighted version of the model fitting process in order to improve the fit on other spatial regions.

Even with the observed bias however it appears that the spatial SEIR model is able to deliver a satisfactory approximation of the posterior predictive distribution for the observed data in most instances. The focus of the model fitting performed in this mini-dissertation was on model fit and approximation capability. Extending these models to the prediction of new cases forward in time is a challenge that requires specific and lengthy consideration, which is left to future research.

In many epidemiological studies it is most often assumed that the latent and infectious times follow an exponential distribution. Using the increased flexibility offered by the spatial SEIR model we were able to fit the same model assuming either exponential or gamma transition probabilities. Through the use of Bayes factors we were able to determine that there is evidence to suggest that the gamma model provides a better fit to the data. However the Bayes factor in favour of the gamma model was 2.152, which indicates that the evidence is somewhat weak (see Table 5.1 for interpretation). This would indicate that either model provides a satisfactory fit to the data, with the gamma model having the advantage only slightly in terms of goodness of fit. We determined the empirically adjusted reproductive number, $R^{(EA)}(t)$, for both models over the entire study period. These results can be found in Section 8.4.

The time-varying $R^{(EA)}(t)$ estimates are plotted in Figure 8.16 - Figure 8.20. When inspecting these results we can immediately note that all estimates of $R^{(EA)}(t)$ are relatively low for both models. Recall that the reproductive number in essence indicates the expected number of secondary infections over the study period. The study period spans what is referred to as the "first wave" of COVID-19 infections in the study region. Describing the time period as a "wave" is meant to indicate that it was a period when new daily infections increased drastically for a time before decreasing significantly to a point where new daily infections were low. In Figures 8.8 - 8.13 it is easy to spot that daily new infections were strikingly low at the end of the study period for most district municipalities. This leads us to conclude that the pandemic did all but "die out" in most of the district municipalities, with new infections being introduced at some later point in time. We thus do not find it surprising that the estimated $R^{(EA)}(t)$ values are so low.

For the gamma model we find that $R^{(EA)}(t) < 1 \, \forall \, t$, with the highest estimated value being approximately 0.6 for the City of Cape Town Metro at the start of the study period. While these estimates are accurate as per the definition of the empirically adjusted reproductive number, they are not particularly useful to policymakers since they clearly seem to under-estimate the transmission potential of the disease. Most studies concerning COVID-19 have estimated the basic reproductive number (R_0) to be within the range of $[2, 4]$, with some studies even resulting in estimates larger than 6 [128]. Given their similar interpretation, we would expect $R^{(EA)}(t)$ to exhibit at least somewhat similar behaviour. For the exponential model we achieve estimates that are more in line with these expected outcomes. This model correctly identifies the City of Cape Town Metro as a hot-spot for COVID-19 infections over the study period, with the 95% confidence interval for $R^{(EA)}(t)$ including values as high as 4 at the start of the study period. While it is the case that the mean $R^{(EA)}(t)$ values are still relatively low, we believe that the 95% confidence interval provides more insight given the corresponding confidence intervals for the predicted daily number of cases (see Figure E3). Another district municipality worth mentioning is the Cape Winelands, which also exhibits estimates for $R^{(EA)}(t)$ with 95% confidence intervals exceeding 1, granted only at the start of the study period. As mentioned previously, this district municipality displayed an intensity process similar to the City of Cape Town Metro, but had much less cases of COVID-19.

The analysis discussed above was also performed for two single location models, with the COVID-19 case data for all district municipalities aggregated to provincial level (see Table 2.1 for South Africa administrative boundary definitions). These results are provided in Section 8.5. We once again note that the Bayes factor indicates some evidence in favour of the gamma model instead of the exponential model. The Bayes factor for this comparison is 1.566, which indicates even less evidence of improved model performance than when comparing the district municipality-level models. The posterior predictive distributions for both the provincial-level models are shown in Figures F1 and F2. The estimated empirically adjusted reproductive numbers for both models are shown in Figure 8.21. Here it is immediately apparent that the gamma model retains its earlier behaviour of being highly conservative with regards to the estimated empirically adjusted reproductive number. While technically not incorrect (the pandemic did die out over the study period and so we would expect $R^{(EA)}(t)$ to be less than 1), the estimated values are extremely low for a disease modelled at a provincial level. Furthermore the 95% confidence interval for these estimated values is unusually narrow, indicating little to no variation in this estimation. The estimated empirically adjusted reproductive numbers for the provincial level exponential model on the other hand includes significant variation over time. These estimated $R^{(EA)}(t)$ values offer a much more realistic expression of what the spread of the disease was like over the province as a whole, with an initially rapid spread that decreased over time, only reaching a point where the pandemic could potentially die out ($R^{(EA)}(t) < 1$) at the end of the study period. Despite the fact that these two models differ only with regards to the latent and infectious time distributions, they show significant deviance with respect to these results.

These two models offer different perspectives on the spread of the disease across the study region. The gamma model performs slightly better in producing simulations that accurately mirror the actual observed data but produces an estimated empirically adjusted reproductive number that is conservative and lacks variation. The exponential model performs slightly worse with regards to the posterior predictions but results in an estimated empirically adjusted reproductive number that shows more variation and will be more informative when assessing the affect of non-pharmaceutical interventions on the spread of the disease. These differences could be due to the fact that the current implementation of the `ABSEIR` package allows for any general distribution(s) to be chosen to model the latent and infectious time periods but does not perform posterior inference for any distribution other than the exponential distribution. This could explain why the $R^{(EA)}(t)$ show such little variation due to the package not performing posterior inference on these parameters specifically. Should this particular posterior inference be implemented it is likely that the $R^{(EA)}(t)$ values for the gamma model could be brought more in line with expected outcomes.

When comparing the similarity of the two COVID-19 case datasets obtained from web scraping press releases and from the NICD website, we first employed the Kolmogorov Smirnov (KS) test. The results of the KS test indicates that the case data does indeed differ significantly (at any reasonable level of significance) for at least four of the six district municipalities. These correspond to the district municipalities that had the lowest number of cases over the study period and are the district municipalities that do not share a similar intensity process shape with the City of Cape Town Metro. The KS test also establishes that the shape of the cumulative infection curves are not significantly different for the Cape Winelands and City of Cape Town Metro. This is rather interesting given that we have previously identified these two district municipalities as being very influential in the modelling process. The City of Cape Town Metro benefits greatly from the fitting procedure due to the sheer scale of its case numbers while the Cape Winelands has a similar intensity process as the City of Cape Town Metro. These two district municipalities also have the largest population sizes. We speculate that the reason for the observed differences in the two datasets are due to the data capturing practices and capabilities not being homogeneous across the entire study region. The City of Cape Town Metro is home to the City of Cape Town, one of the largest cities in South Africa and the capital of the Western Cape. Given that this district municipality also has the largest population in the Western Cape, it makes sense that the highest number of COVID-19 cases would be identified here due to increased risk of infection as well as an increased number of COVID-19 testing being performed. The increased resources would also be beneficial towards maintaining consistency in the case numbers for this district municipality. Other district municipalities are comparatively less likely to receive the same amount of resources dedicated towards performing COVID-19 testing, which could lead to inconsistencies being present in case numbers.

When fitting the SMC algorithm model to both datasets we arrive at the posterior predictive results shown in Figures 8.22 - 8.27. We note that despite the KS test identifying significant differences in the data for four of the six district municipalities that the resulting models are not very different in terms of posterior predictive distributions. Despite the press release data (shown in red) exhibiting more outliers than the NICD data (shown in blue) the resulting models produce rather similar predictions. This leads us to speculate that these models are not highly sensitive to the presence of outliers and that subtle differences in data do not lead to drastic differences in model performance which is an ideal feature for a modelling technique to exhibit. Thus, despite the two data sources having been confirmed to not be identical we have observed that the resulting model fit for the two datasets is comparable.

Two different approximate Bayesian computation techniques were considered and utilized for the fitting of the compartmental models in this mini-dissertation. The ABC rejection algorithm (given in Algorithm 1) was shown to not perform adequately, with the estimated posterior predictive distributions providing poor fits to the data for all district municipalities except for the City of Cape Town Metro. The final tolerance value achieved when fitting the model using this algorithm was $\epsilon = 6,566.23$. The sequential Monte Carlo ABC algorithm (given in Algorithm 5) proved far more capable of producing adequate estimates of the posterior predictive distributions. The resulting posterior predictive distributions proved to be good reproductions of the observed data however arguably the best fit overall was still achieved for the City of Cape Town Metro as well as the Cape Winelands which is believed to exhibit a similar intensity process over the study period. The final tolerance achieved when using this algorithm to fit the model was $\epsilon = 2,888.96$, which is less than 50% of the tolerance achieved by the ABC rejection algorithm. This result is not very surprising however given the highly simplistic nature of the rejection algorithm. The rejection algorithm merely simulates a set of parameter values and retains those that produce artificial datasets with sufficient similarity to the observed data. In contrast, the sequential Monte Carlo algorithm sequentially samples new parameter values from previously accepted ones in order to improve upon the estimation. It is evident in the results achieved using both algorithms however that there is a source of bias present in the way which these algorithms fit a model. Due to the difference between the artificial and observed datasets being determined as simply the distance between these datasets (in this case the Euclidean distance), district municipalities with a larger scale to their observations will benefit from increased priority since the model fit is improved most significantly when improved with respect to these regions specifically. Future research may strive to improve upon this by considering and implementing a weighted approach to the distance function evaluation.

As the final analysis conducted in this mini-dissertation we studied the effect of using autoencoder neural networks in conjunction with ABC. We trained both an autoencoder ANN as well as an autoencoder DNN using a large sample of artificially simulated pandemic datasets. All values were scaled to be within the interval $(0, 1)$ during model training (the scaling was done for each district municipality individually to avoid problems of scale). The models were trained by optimising the cross-entropy loss function and the final loss values (both training and validation) for both models was approximately 43%. This loss is somewhat high and consequentially it was observed that when used to summarise a separate set of artificial datasets the models were not able to correctly identify the pandemics with the most similarity to the observed data. Model performance did not improve after increasing the training data sample size or changing the architecture of the autoencoders. It is rather telling that the performance of the autoencoder DNN was not a meaningful improvement over that of the autoencoder ANN. This made it immediately apparent that increasing model complexity would not improve model performance for this particular problem.

There are many potential reasons for why we observe this rather poor performance. Firstly there is the fact that the performance of neural networks is often heavily dependent on the quality and accuracy of the data they are trained upon. Rather than training the models on the actual observed data (which is only a single observation), the models were trained on pandemic data that was simulated using the model parameters of the SMC algorithm model. Thus the data that was used to train the autoencoders are simulations with values determined by a model fitted to the observed data which is assumed to contain some degree of noise and inaccuracy due to inconsistent data capturing practices. The training data thus was not “real” data but instead involved several steps of processing to be derived. Such undesirable data features have the potential to adversely affect any analysis conducted using it and we theorise that this might be the case for the neural networks.

Another contributing factor to the poor performance of the neural networks may be the fact that neither model is able to interpret the data as time-series data. The original dimensionality of the data is 181×6, since we study the spread of the COVID-19 pandemic for 181 days across 6 district municipalities. Despite it being the case that the data is in actuality 6 different time-series datasets, both models simply interpret the data as a sequence of 1086 entries, thus disregarding its bi-dimensional shape. In order to improve upon this, the most immediate solution would be to instead utilize an autoencoder based on a CNN model (see Appendix C), since CNN's are adept at working with bi-dimensional data such as images and time-series. In this mini-dissertation it was hoped that we would be able to utilize such an autoencoder but unfortunately the dimensions of the data are not large enough to support this. Since the data is 181×6, the second dimension (indicating spatial region) is much smaller than the first (indicating time). It is thus not possible to apply a filter to the data of such a size such that it would significantly reduce the size of the first dimension without reducing the second dimension down to just one (and thus losing the bi-dimensional nature of the data in the process anyway). In order to facilitate the application of autoencoder ANN's within the ABC framework of a model for COVID-19 as was attempted in this mini-dissertation, the data would have to be of a spatial resolution higher than district municipality such as local municipality level or even ward level (such as [42]).

Chapter 10

Conclusion

COVID-19 is an infectious respiratory disease that caught the world unawares and introduced the need for innovative and skilful research endeavours in order to improve the limited understanding of the spread of the disease. The spread of COVID-19 is affected by numerous factors that render modelling it very complex, often requiring an understanding of multiple fields of statistical analysis. A key component among these is the need for some spatial structure that approximates the mobility pattern of potentially infectious individuals.

In this mini-dissertation we showed how different spatial structures can be conceptualised using either time-tested or novel approaches and used to determine the degree of spatial autocorrelation of COVID-19 between different regions. We demonstrated how different sources of mobility data can yield spatial structures capable of representing different types of spatial associations. We also discussed how failing to consider or include multiple sources of mobility data could lead to disregarding existing spatial patterns. The inclusion of mobility data for modelling the spread of COVID-19 should not be disregarded and alternative sources of data and chosen spatial structures should be considered for a given task to explore a full range of possible inference.

Working with population mobility data does however introduce challenges such as computational cost, poor spatial resolution, lack of access to quality data and concerns of population representativeness. In this mini-dissertation it was necessary to aggregate certain mobility data to a lower spatial resolution in order to facilitate comparisons with other data. In addition to proving somewhat preventative to analysis, this resulted in the loss of a great deal of information. Future research should focus on developing ways to aggregate spatial covariate data to a higher spatial resolution in order to retain a larger degree of available information.

The spread of a disease such as COVID-19 involves many unknown parameters with inherent uncertainty and so the use of models that are stochastic rather than deterministic in nature is highly warranted. In this mini-dissertation we fit a stochastic spatial compartmental epidemiological model to the number of new daily COVID-19 cases in the Western Cape province in South Africa. We showed that this stochastic model enables a large amount of flexibility with regard to model parameterization and variable uncertainty. We feel that there is a need for more examples of stochastic models being utilized for the modelling of COVID-19 within the South African context. We believe that the research conducted in this mini-dissertation serves as a sufficient contribution to the currently existing literature.

In order to fit and evaluate these stochastic models we relied on approximate Bayesian techniques. While two algorithms were considered, analysis was conducted prominently using sequential Monte Carlo methods. The resulting models produced satisfactory posterior predictive estimates however some sources of bias were identified, particularly with regards to vastly different observation scales for study regions. Future research should focus on developing model evaluation techniques that are unaffected by these sources of bias. Alternatively, the establishment of a satisfactory data pre-processing method could also rectify this issue. Overall however these approximate Bayesian techniques represent a great tool for the purpose of fitting and evaluating possibly highly complex models in an intuitive and easily understood way.

We discussed and compared inference derived from models fitted to two sources of COVID-19 case data. The first dataset was made freely available by the NICD on their website while the second was web scraped from the press releases of the Western Cape premier. We confirmed that despite both being from reputable sources, the two datasets were significantly different for the majority of the district municipalities in the study region. However due to the modelling procedure being insensitive to these differences in the data it was shown that the resulting model performance was still comparable even for district municipalities with vastly different data. In earlier chapters we discuss the situation surrounding COVID-19 case data in South Africa at length. We feel it is crucial for researchers to constantly discuss such matters and treat available data with scrutiny whenever possible. Future research attempts should continue to consider and compare alternative sources of data in order to express the importance of data reliability to policymakers.

We also considered the use of artificial neural networks for the purpose of summarising pandemic data in this mini-dissertation. While it is the case that unfortunately the data did not support the use of these techniques we hold fast that there is room for the application of deep learning techniques to aid in the fight against COVID-19. Future research should attempt to utilize the ideas presented here for datasets that possess the necessary properties we have previously listed. Future research should also invest even further consideration into neural network architecture and modelling choices to uncover the full inferential power of these models.

Lastly, we urge that future research should be founded on the idea that solutions to complex real-world problems may be formulated as a fusion of distinct and perhaps unrelated fields of statistical analysis. We feel that a diverse set of methods were selected and applied in this mini-dissertation, each of which contributed in some way towards improving our understanding of the COVID-19 modelling process. Only through collaboration and exploration can the true discovery potential of research be fully realized.

Appendix

Appendix A - South African government website scraping permissions

Here we provide the “robots.txt” file for the South African government’s website (https:www.gov.za). Note that the file does not explicitly disallow web scraping aside from a few select directories such as `user`, `admin`, `comments`, `filter` etc. Furthermore the file specifies that a crawl-delay of 10 seconds must be implemented when scraping from the site and thus all web scraping implemented in this mini-dissertation incorporates a 10 second delay between successive requests.

```
#
# robots.txt
#
# This file is to prevent the crawling and indexing of certain parts # of your site by web crawlers and spiders run by sites like Yahoo! # and Google. By telling these "robots" where not to go on your site, # you save bandwidth and server resources.
#
# This file will be ignored unless it is at the root of your host:
# Used: http://example.com/robots.txt
# Ignored: http://example.com/site/robots.txt
#
# For more information about the robots.txt standard, see:
# http://www.robotstxt.org/robotstxt.html

User-agent: *
Crawl-delay: 10
# CSS, JS, Images
Allow: /misc/*.css$
Allow: /misc/*.css?
Allow: /misc/*.js$
Allow: /misc/*.js?
Allow: /misc/*.gif
Allow: /misc/*.jpg
Allow: /misc/*.jpeg
```

```
Allow: /misc/*.png
Allow: /modules/*.css$
Allow: /modules/*.css?
Allow: /modules/*.js$
Allow: /modules/*.js?
Allow: /modules/*.gif
Allow: /modules/*.jpg
Allow: /modules/*.jpeg
Allow: /modules/*.png
Allow: /profiles/*.css$
Allow: /profiles/*.css?
Allow: /profiles/*.js$
Allow: /profiles/*.js?
Allow: /profiles/*.gif
Allow: /profiles/*.jpg
Allow: /profiles/*.jpeg
Allow: /profiles/*.png
Allow: /themes/*.css$
Allow: /themes/*.css?
Allow: /themes/*.js$
Allow: /themes/*.js?
Allow: /themes/*.gif
Allow: /themes/*.jpg
Allow: /themes/*.jpeg
Allow: /themes/*.png
# Directories
Disallow: /includes/
Disallow: /misc/
Disallow: /modules/
Disallow: /profiles/
Disallow: /scripts/
Disallow: /themes/
# Files
```

```
Disallow: /CHANGELOG.txt
Disallow: /cron.php
Disallow: /INSTALL.mysql.txt
Disallow: /INSTALL.pgsql.txt
Disallow: /INSTALL.sqlite.txt
Disallow: /install.php
Disallow: /INSTALL.txt
Disallow: /LICENSE.txt
Disallow: /MAINTAINERS.txt
Disallow: /update.php
Disallow: /UPGRADE.txt
Disallow: /xmlrpc.php
# Paths (clean URLs)
Disallow: /admin/
Disallow: /comment/reply/
Disallow: /filter/tips/
Disallow: /node/add/
Disallow: /search/
Disallow: /user/register/
Disallow: /user/password/
Disallow: /user/login/
Disallow: /user/logout/
# Paths (no clean URLs)
Disallow: /?q=admin/
Disallow: /?q=comment/reply/
Disallow: /?q=filter/tips/
Disallow: /?q=node/add/
Disallow: /?q=search/
Disallow: /?q=user/password/
Disallow: /?q=user/register/
Disallow: /?q=user/login/
Disallow: /?q=user/logout/
Sitemap: https://www.gov.za/sitemap.xml
```

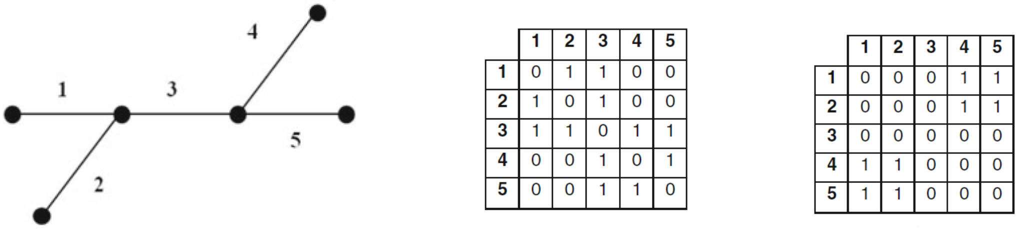
Appendix B - Contiguity on a network

While not explicitly relevant to the application done in this mini-dissertation, we discuss the application of spatial contiguity to cases where spatial locations are defined on a network. The concept of contiguity can be readily applied to spatial locations defined on a network [29, 30, 94]. A network can be viewed as a graph $G = (N, E)$ that contains a set of nodes labeled $N = \{1, 2, \ldots, n\}$ and a set of edges E that links pairs of nodes [29].

Unlike how we determined contiguity when considering areal spatial units, with network data we rather evaluate the *adjacency* between nodes based on whether or not there exists an edge that links them [94]. Rather than the nodes, it's also possible to instead regard the edges between them as the spatial units. This is particularly relevant when working with, for example, road traffic data [29, 30]. In this case, edges are considered first-order adjacent if there is a node at which the two edges intersect [29, 30]. An illustration of this is given at Figure B1 (a) and (b) where each pair of edges that intersect at a node has a weight of 1 in $\boldsymbol{W}$ and all other pairs have a 0. The second-order adjacency matrix can be determined in a similar manner as the second-order contiguity matrix before, which is shown at Figure B1(c).

A special property of networks is that they can either be undirected or directed [29]. A directed version of the network just discussed is given at Figure B2 where the arrows indicate the flow of some phenomenon (possibly traffic [29]). What differs from the previous case is that influence between spatial units only flows in the direction indicated by the arrows. So for example, edges 1 and 3 intersect at a node, but since the network flows from 1 *to* 3 we set $w_{1,3} = 1$ and $w_{3,1} = 0$. Once again the second-order adjacency matrix follows in a similar manner.

In their 2014 study Wang et al. utilized a directed spatial network to create a dynamic spatial weight



(a) Undirected network

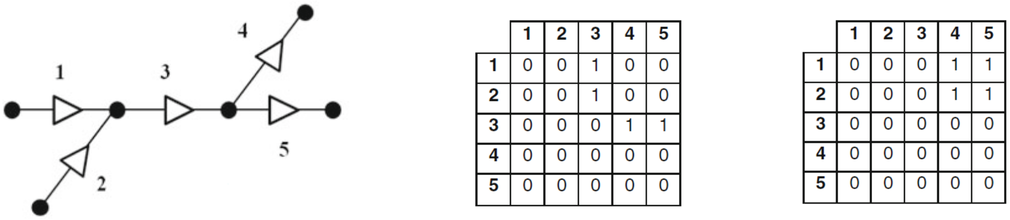
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	1	0	0
3	1	1	0	1	1
4	0	0	1	0	1
5	0	0	1	1	0

(b) First-order adjacency matrix

	1	2	3	4	5
1	0	0	0	1	1
2	0	0	0	1	1
3	0	0	0	0	0
4	1	1	0	0	0
5	1	1	0	0	0

(c) Second-order adjacency matrix

Figure B1: Adjacency of undirected network



(a) Directed network

	1	2	3	4	5
1	0	0	1	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	0	0	0	0	0
5	0	0	0	0	0

(b) First-order adjacency matrix

	1	2	3	4	5
1	0	0	0	1	1
2	0	0	0	1	1
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

(c) Second-order adjacency matrix

Figure B2: Adjacency of directed network

matrix that incorporates both the average traffic speed of different edges roads as well the propensity for vehicles to travel from one road to another within a 5-minute time period [30]. They achieved this by checking the travel times for various vehicles and determining how many nodes vehicles tend to cross in a single 5-minute period (and hence how many edges do they encounter). They found that at most a vehicle can travel over 3 edges within a 5 minute period and thus spatial weight matrices up to third-order were included in the final model [30]. The spatial weights were assigned as:

$$w_{ij}^{(h)}(t) = \begin{cases} \frac{v_j(t)-v_i(t)}{v_i(t)} & \text{if } j \text{ is upstream of } i \\ \frac{v_i(t)-v_j(t)}{v_i(t)} & \text{if } j \text{ is downstream of } i \end{cases}$$

where $v_i(t)$ and $v_j(t)$ are the average traffic speeds for road i and j respectively. Note that this allows for negative weights which was deemed appropriate for the particular model used since this would imply that the average traffic speed on one road decreases the travel time on another [30].

Appendix C - Convolutional neural networks

In this mini-dissertation it was not possible to utilized CNN's for our intended purposes due to data constraints. However given their potential to be applied in some future attempt at similar research, we include a discussion on their characteristics in this section.

A CNN is a specific type of neural network that is adept at processing data that can be arranged into a grid-like structure such as images or time series [52]. CNN derives its name from the *convolutional* operation that is performed (at least once) within its hidden layers [20, 52]. Within this context, the convolutional operator is defined as:

$$s(t) = (y * w)(t) = \sum_{a=-\infty}^{\infty} y(a) \cdot w(t-a) \tag{1}$$

for input data y, kernel w and time point t [52, 2]. The kernel w is essentially a filter of adaptive weights that is applied over data that is arranged in a grid-like structure [2]. For example, consider some input data that is arranged in a 3×3 grid and a 2×2 kernel of adaptive weights. The result of applying the kernel to the input data is shown at Figure C1. Note that each entry in the resulting feature map [52] is the dot product of a 2×2 subset of the inputs and the convolutional kernel plus a bias term.

Each node within the hidden convolutional layers of the CNN has its own weight filter and bias term and thus produces its own feature map. Once the feature map for a node is determined, each entry is passed through a continuous activation function as was done for FFNNs and DNNs. Each entry of the resulting feature map is thus a non-linear transformation of a weighted linear combination of the input data plus a bias term. The entries within the CNN feature maps are thus similar to the FFNNs and DNNs in terms of output, but with the output arranged into a grid. After passing through the activation function, we can then either feed forward the data to another convolution layer, a standard hidden layer or we can use what is known as a *pooling layer* to reduce the dimensionality of the data [52, 2]. A pooling layer replaces every $m \times m$ subset of values (where m is chosen by the user) with a summary statistic such as their maximum or mean value [52, 2]. Aside from reducing the dimensionality of the data, this also renders the data invariant to small translations in the input since these values' activation function responses will either be discarded or smoothed out [52].

There are many benefits to using a CNN for high-dimensional data. By using a weight kernel that has smaller dimensions than the input data we achieve a model with sparse connectivity [52]. Note how at Figure C1 we see that not all data points interact with one another in the convolutional operation (for

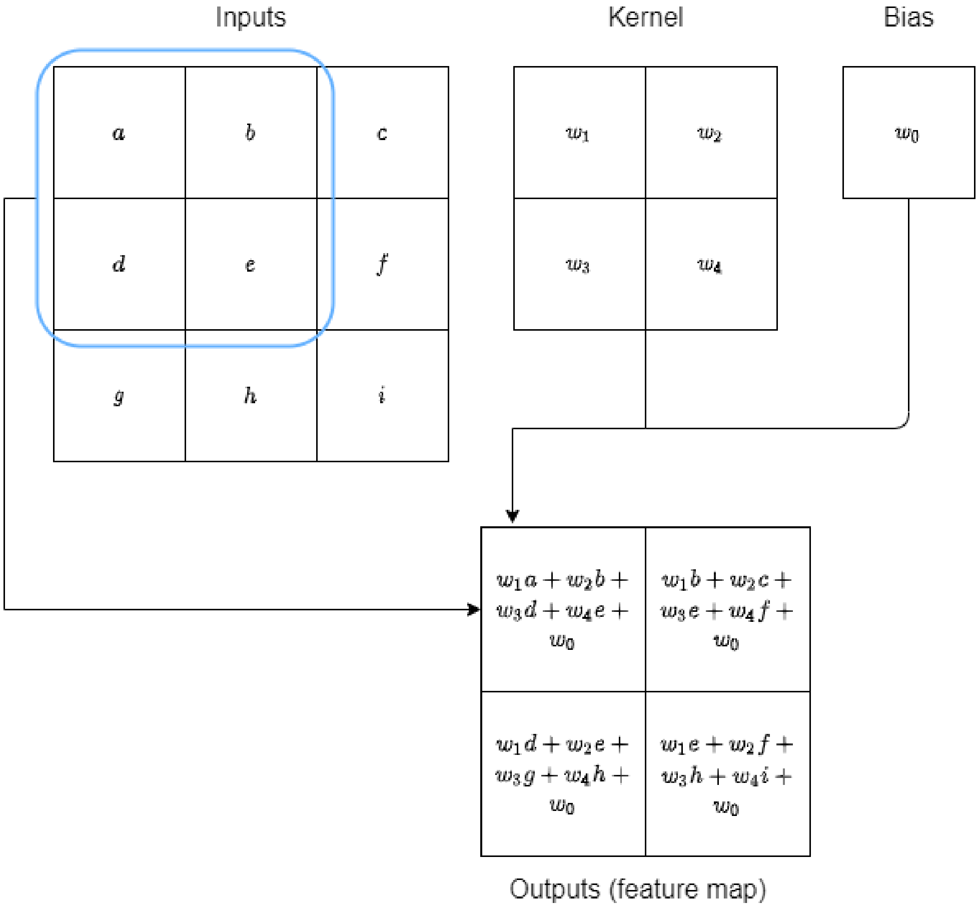

Inputs
a
b
c
d
e
f
g
h
i
Kernel
w_1
w_2
w_3
w_4
Bias
w_0
$w_1 a + w_2 b + w_3 d + w_4 e + w_0$
$w_1 b + w_2 c + w_3 e + w_4 f + w_0$
$w_1 d + w_2 e + w_3 g + w_4 h + w_0$
$w_1 e + w_2 f + w_3 h + w_4 i + w_0$
Outputs (feature map)

Figure C1: Diagram of convolutional operation

example data points a and i are never placed within the same linear combination). This differs from previously discussed neural networks where all data points interacted with one another. This allows us to identify important data features over relatively small areas or timespans by focusing on observations that are situated more closely together in space or time [52]. Furthermore, while not every observation is used for each convolution, each kernel weight is (note how all four kernel weights are used for each convolution in Figure C1). This means that instead of determining a weight for every single observation as we would with previously discussed neural networks (which would result in a total of nine weights), we only need to determine the smaller number of weights (in this case four) that are shared by all observations, thus reducing the number of parameters that need to be estimated [52]. Lastly, by using a CNN we achieve a model that is invariant to many transformations, producing a model with a degree of robustness [20]. Kesson et al. [2] compared the performance of a CNN, DNN and PEN and showed that CNNs performed better in cases of high-dimensional and increasingly complex problems for likelihood-free inference [2].

Appendix D - ABC rejection algorithm model posterior predictive distributions

In this section we present the modelling results for the ABC rejection algorithm model. This model included all six district municipalities within the study region with their spatial association being modelled using both mobile network and Facebook data, utilized gamma transitional probabilities and was fitted using the basic ABC rejection algorithm. As mentioned in Chapter 8, the fit of this model is not ideal for any district municipality other than the City of Cape Town Metro.

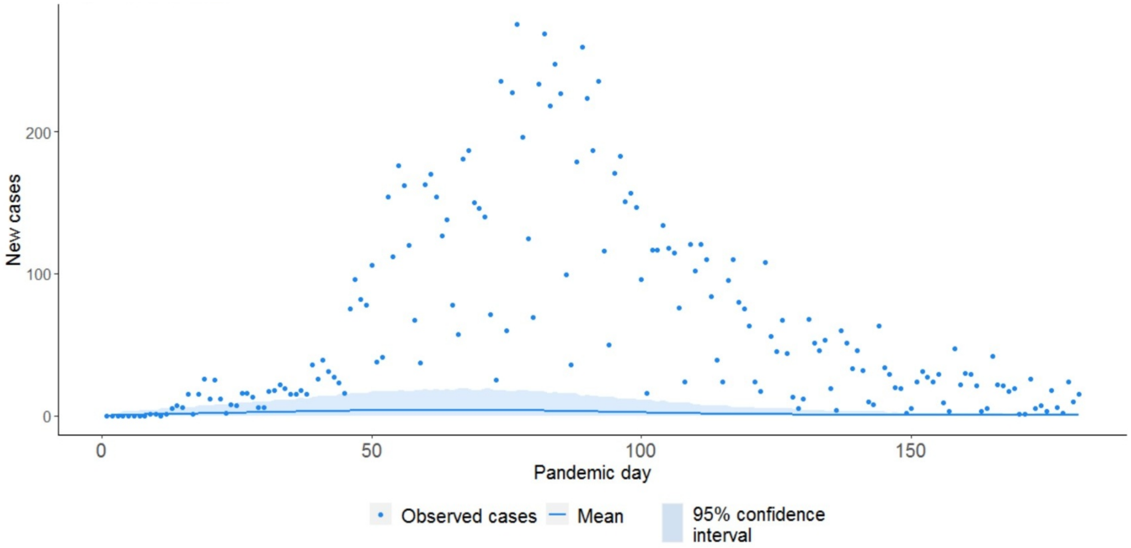


Figure D1: Posterior predictive distribution for Cape Winelands (ABC rejection algorithm model)

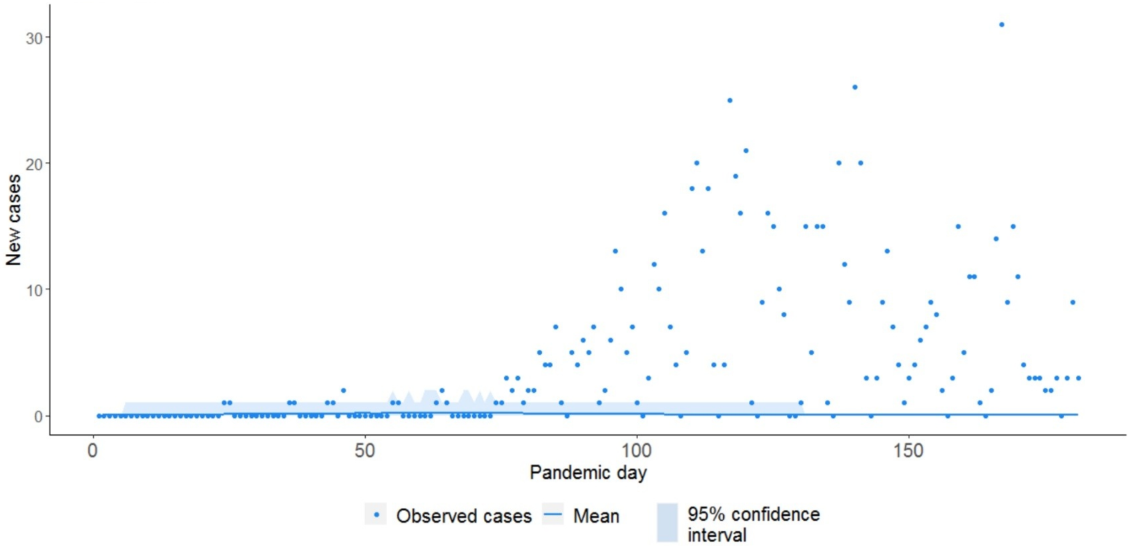


Figure D2: Posterior predictive distribution for Central Karoo (ABC rejection algorithm model)

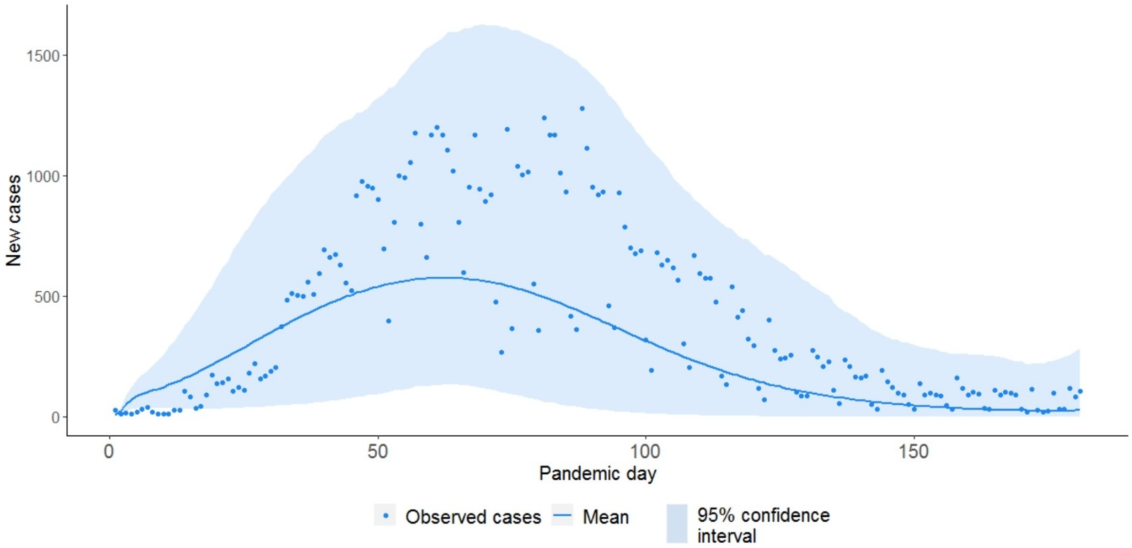


Figure D3: Posterior predictive distribution for City of Cape Town Metro (ABC rejection algorithm model)

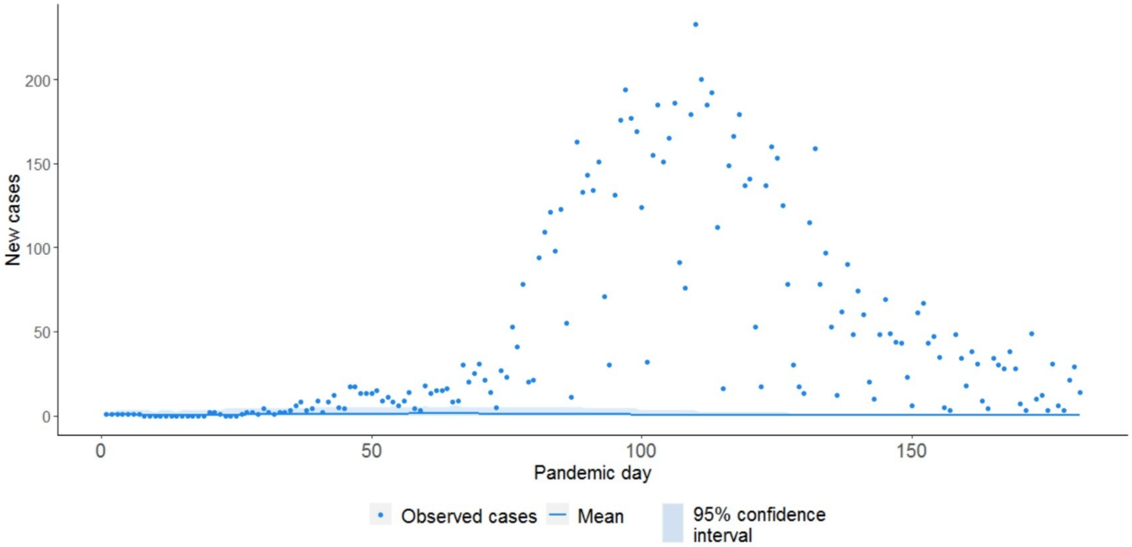


Figure D4: Posterior predictive distribution for Garden Route (ABC rejection algorithm model)

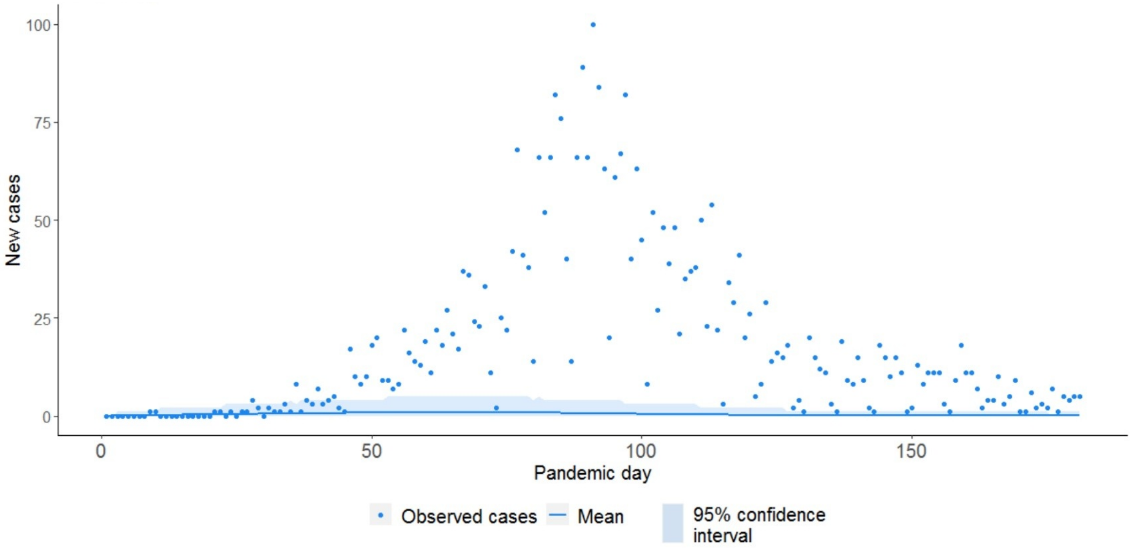


Figure D5: Posterior predictive distribution for Overberg (ABC rejection algorithm model)

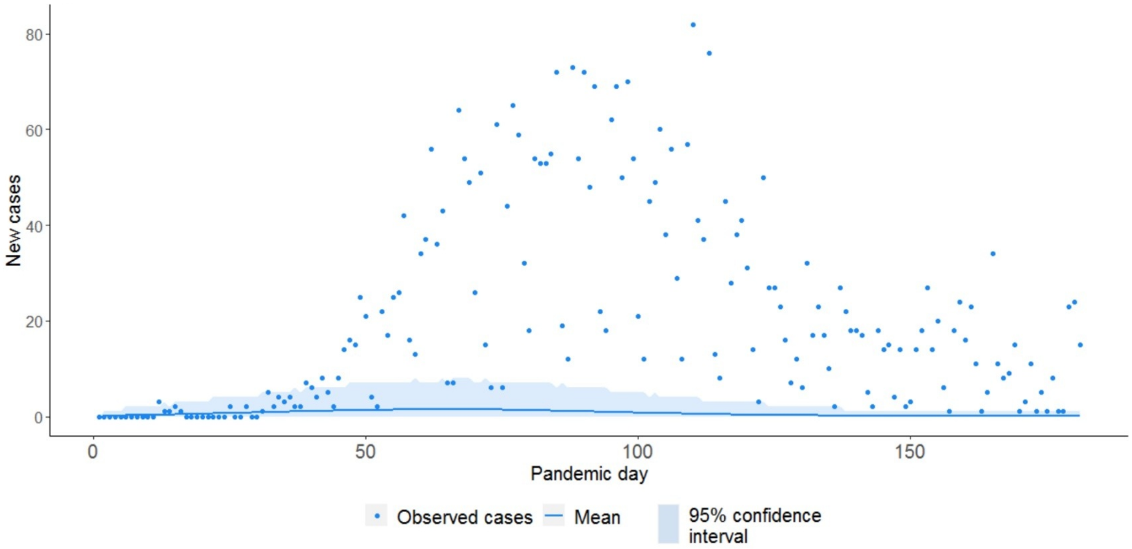


Figure D6: Posterior predictive distribution for West Coast (ABC rejection algorithm model)

Appendix E - Exponential transition model posterior predictive distributions

In this section we present the posterior predictive results for the exponential transition model. This model differs from the SMC ABC algorithm model by having the transition probabilities from the "Exposed" and "Infectious" compartments follow an exponential distribution as is often assumed in compartmental modelling literature [72, 98]. The mean of the transition probability distributions were chosen identical to previous models (i.e. 4 and 10 days for latent and infectious period respectively).

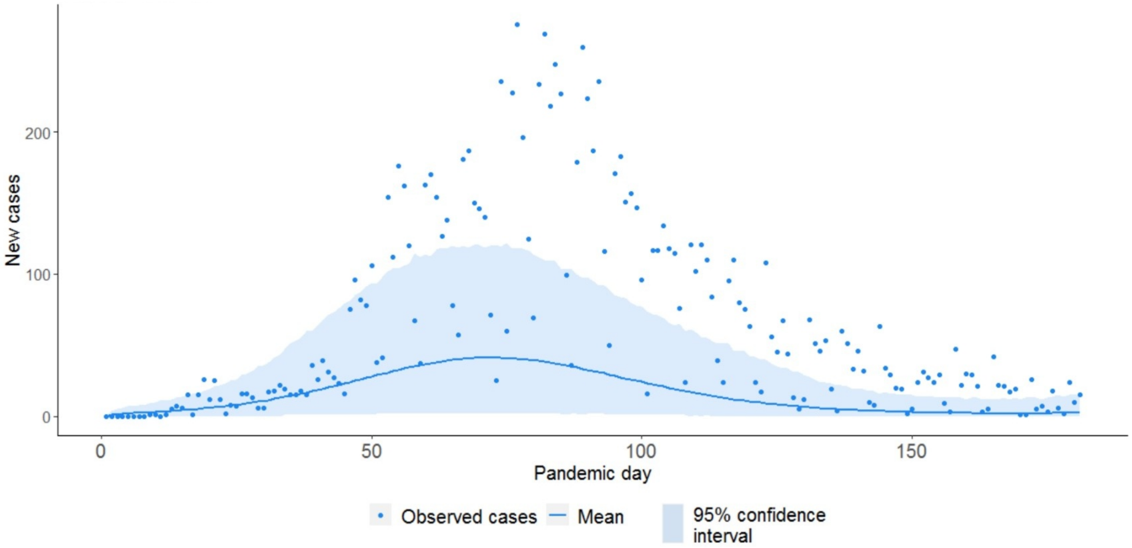


Figure E1: Posterior predictive distribution for Cape Winelands (exponential transition model)

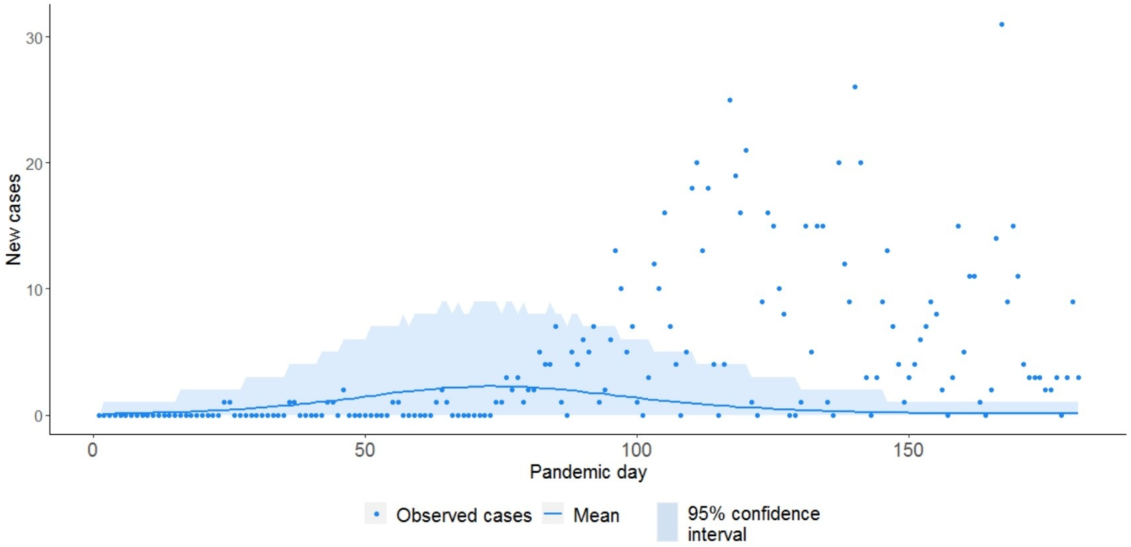


Figure E2: Posterior predictive distribution for Central Karoo (exponential transition model)

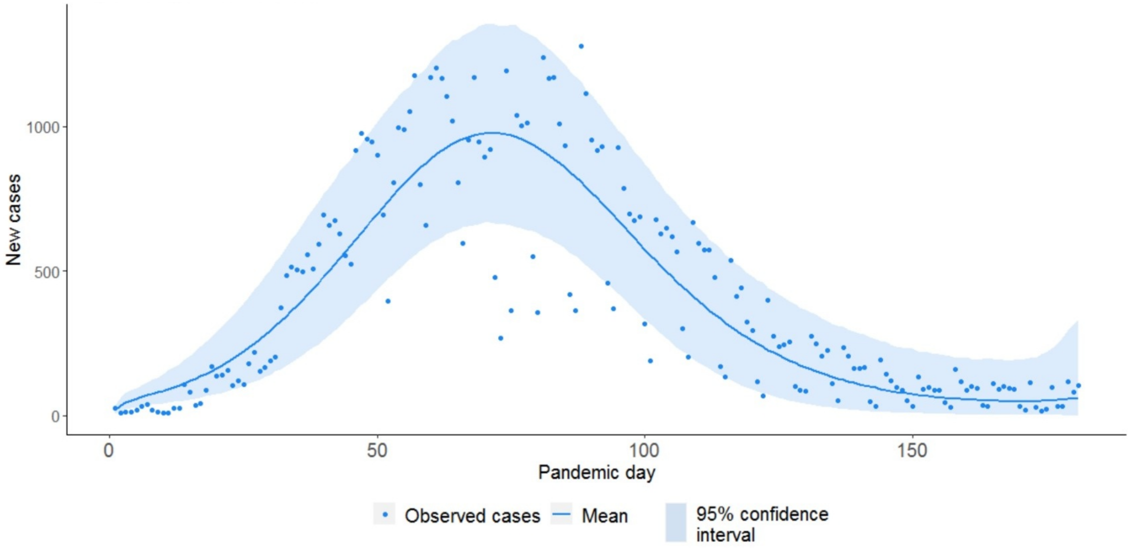


Figure E3: Posterior predictive distribution for City of Cape Town Metro (exponential transition model)

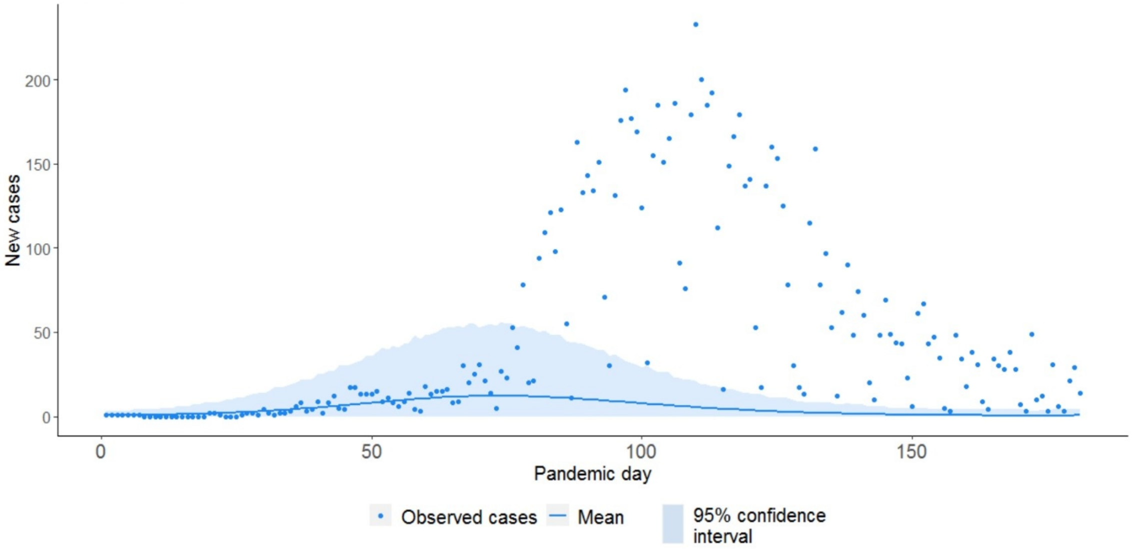


Figure E4: Posterior predictive distribution for Garden Route (exponential transition model)

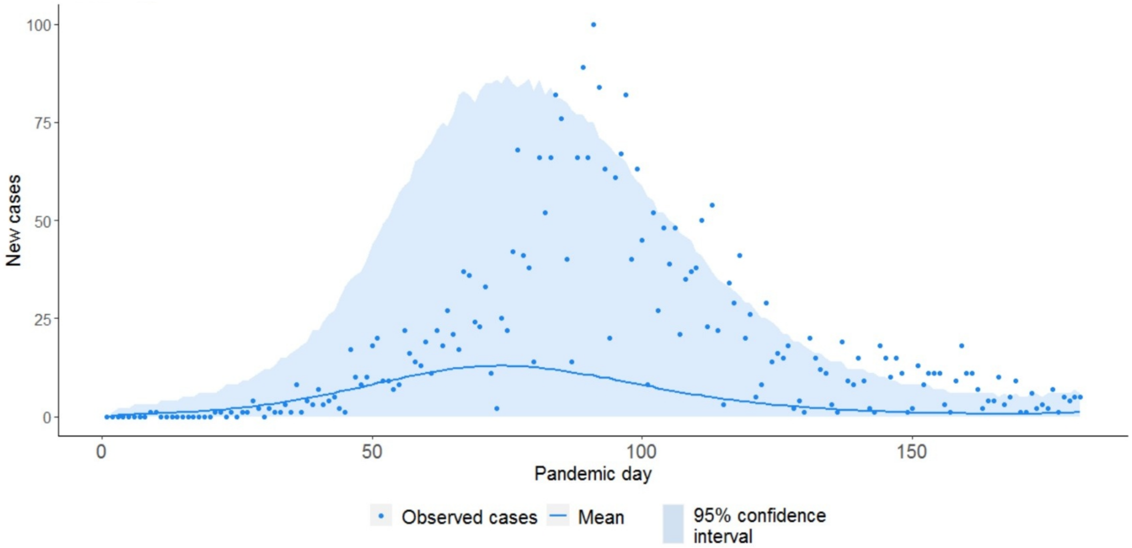


Figure E5: Posterior predictive distribution for Overberg (exponential transition model)

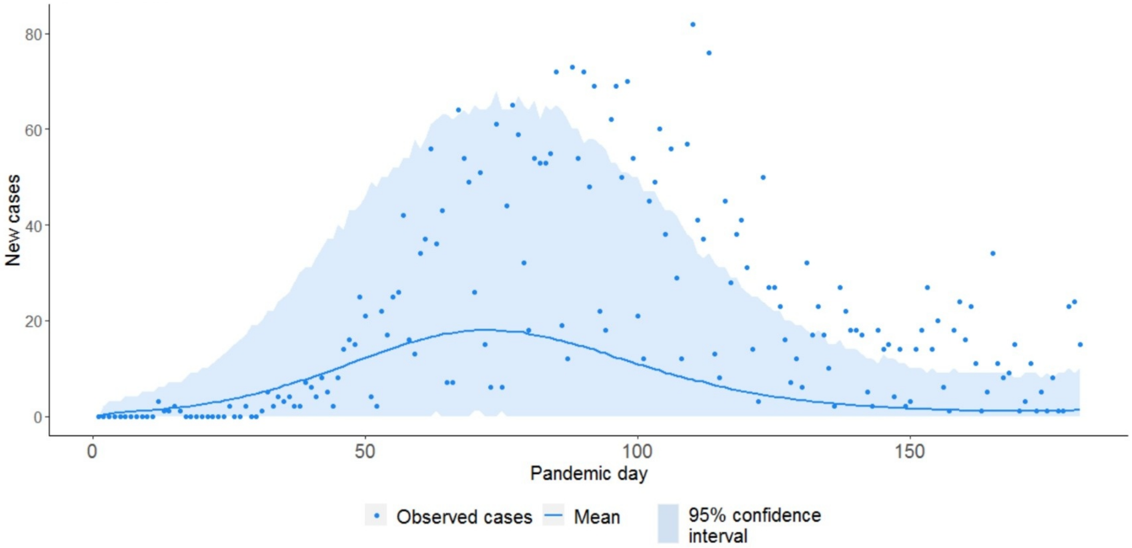


Figure E6: Posterior predictive distribution for West Coast (exponential transition model)

Appendix F - Single location model posterior predictive distributions

In this section we present the modelling results when aggregating case data to a provincial level. We present the posterior predictive distributions for both the gamma and exponential transition models. Note that the mean of the transition probability distributions were chosen identical to previous models (i.e. 4 and 10 days for latent and infectious period respectively). It is clear from Figure F1 and Figure F2 that these models differ only superficially with respect to goodness of fit to the data.

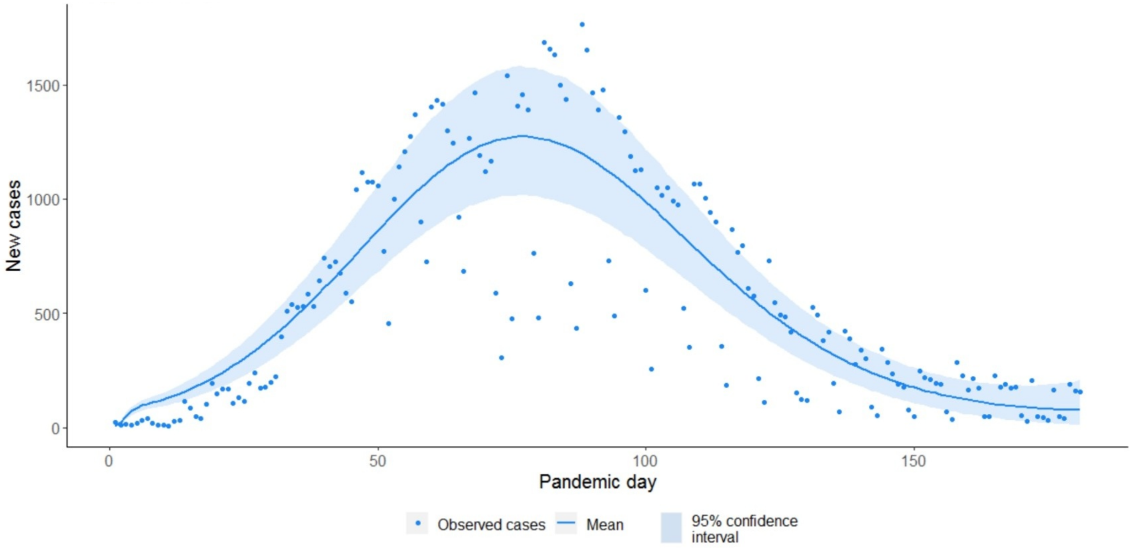


Figure F1: Gamma transition model posterior predictive distribution

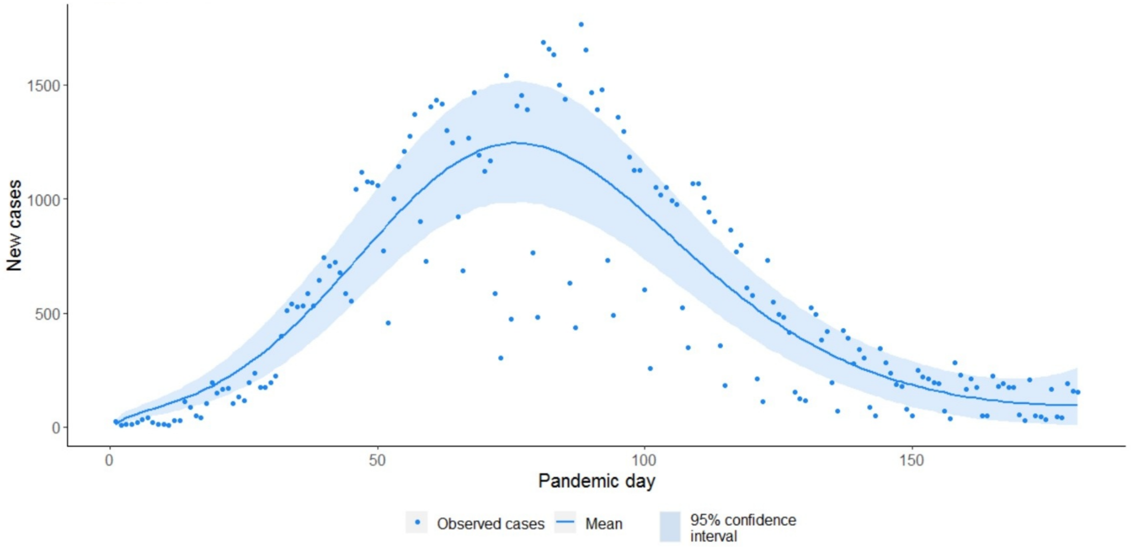


Figure F2: Exponential transition model posterior predictive distribution

The basic reproductive number (R_0) is not defined unambiguously for all compartmental models, particularly not for more complex models. However it is defined for the exponential transition model utilized here. In Figure F3 we provide the distribution of the basic reproductive number for the exponential model. It is interesting to note that the basic reproductive number is more closely centred around 1, with a maximum value of approximately 1.2, as opposed to the empirically adjusted reproductive number for this model which reaches values in excess of 4.

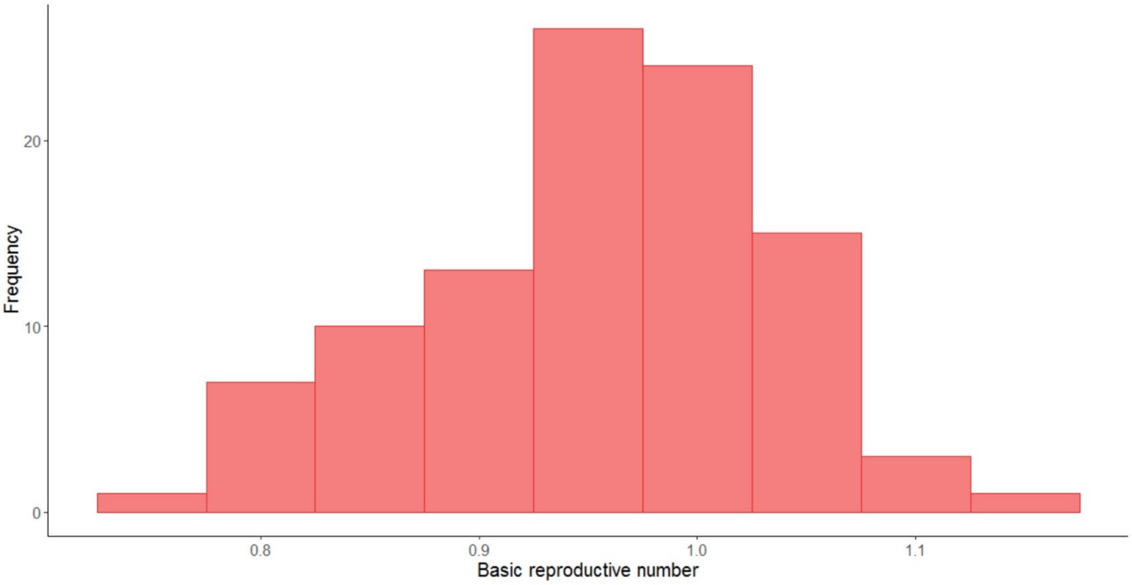


Figure F3: Exponential transition model basic reproductive number distribution

Bibliography

[1] Steven Abrams, James Wambua, Eva Santermans, Lander Willem, Elise Kuylen, Pietro Coletti, Pieter Libin, Christel Faes, Oana Petrof, Sereina A Herzog, Philippe Beutels, and Niel Hens. Modeling the early phase of the Belgian COVID-19 epidemic using a stochastic compartmental model and studying its implied future trajectories. *MedRXiv*, 2020.

[2] Mattias Åkesson, Prashant Singh, Fredrik Wrede, and Andreas Hellander. Convolutional neural networks as summary statistics for approximate Bayesian computation. *arXiv preprint arXiv:2001.11760*, 2020.

[3] Jared Aldstadt and Arthur Getis. Using AMOEBA to create a spatial weights matrix and identify spatial clusters. *Geographical Analysis*, 38(4):327–343, 2006.

[4] James E. Andersen. The gravity model. *Annual Review of Economics*, 3:133–160, 2011.

[5] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43, 2003.

[6] Luc Anselin. *Spatial Econometrics: Methods and Models*. Springer, 1988.

[7] Luc Anselin and Sergio Joseph Rey. *Modern Spatial Econometrics in Practice: A Guide to GeoDa, GeoDaSpace and PySAL*. GeoDa Press LLC, illustrated edition, 2014.

[8] Alex Arenas, Wesley Cota, Jesús Gómez-Gardenes, Sergio Gómez, Clara Granell, Joan T Matamalas, David Soriano-Panos, and Benjamin Steinegger. A mathematical model for the spatiotemporal epidemic spreading of COVID-19. *MedRxiv*, 2020.

[9] Channing Arndt, Rob Davies, Sherwin Gabriel, Laurence Harris, Konstantin Makrelov, Sherman Robinson, Stephanie Levy, Witness Simbanegavi, Dirk van Seventer, and Lillian Anderson. Covid-

19 lockdowns, income distribution, and food security: An analysis for South Africa. *Global Food Security*, 26:100410, 2020.

[10] John E Ataguba. COVID-19 pandemic, a war to be won: understanding its economic implications for Africa. *Applied Health Economics and Health Policy*, 18:325–328, 2020.

[11] Dimitris Ballas, Graham Clarke, Danny Dorling, Heather Eyre, Bethan Thomas, and David Rossiter. SimBritain: A spatial microsimulation approach to population dynamics. *Population, Space and Place*, 11(1):13–34, 2005.

[12] MS Bartlett. Some evolutionary stochastic processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 11(2):211–229, 1949.

[13] F. Bavaud. Models for spatial weights: A systematic look. *Geographical Analysis*, 30:153–171, 1998.

[14] Mark A Beaumont. Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41:379–406, 2010.

[15] Mark A Beaumont. Approximate Bayesian computation. *Annual Review of Statistics and its Application*, 6:379–403, 2019.

[16] Mark A Beaumont, Jean-Marie Cornuet, Jean-Michel Marin, and Christian P Robert. Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990, 2009.

[17] Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

[18] Linus Bengtsson, Jean Gaudart, Xin Lu, Sandra Moore, Erik Wetter, Kankoe Sallah, Stanislas Rebaudet, and Renaud Piarroux. Using mobile phone data to predict the spatial spread of cholera. *Scientific Reports*, 5(1):1–5, 2015.

[19] Christopher M Bishop. *Neural Networks for Pattern Recognition.* Oxford university press, 1995.

[20] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[21] Michael GB Blum and Olivier François. Non-linear regression models for approximate Bayesian computation. *Statistics and Computing*, 20(1):63–73, 2010.

[22] Paola Bortot, Stuart G Coles, and Scott A Sisson. Inference for stereological extremes. *Journal of the American Statistical Association*, 102(477):84–92, 2007.

[23] Seppe Vanden Broucke and Bart Baesens. *Practical web scraping for data science: best practices and examples with Python.* CreateSpace, 2017.

[24] Grant D Brown, Jacob J Oleson, and Aaron T Porter. An empirically adjusted approach to reproductive number estimation for stochastic compartmental models: A case study of two Ebola outbreaks. *Biometrics*, 72(2):335–343, 2016.

[25] Grant D. Brown, Aaron T. Porter, Jacob J. Olsen, and Jessica A. Hinman. Approximate Bayesian computation for spatial SEIR(S) epidemic models. *Spatial and Spatio-temporal Epidemiology*, 24:27–37, 2018.

[26] Alonso Cancino, Carla Castillo, Pedro Gajardo, Rodrigo Lecaros, Claudio Munoz, César Naranjo, Jaime Ortega, Héctor Ramırez, and Santa Marıa. Report 2: Estimation of maximal ICU beds demand for COVID-19 outbreak in Santiago, Chile. Technical report, CMM-AM2V-CEPS, 2020.

[27] Joel M Caplan, Leslie W Kennedy, and Christine H Neudecker. Cholera deaths in Soho, London, 1854: Risk terrain modeling for epidemiological investigations. *PloS ONE*, 15(3):e0230725, 2020.

[28] Jeffrey Chan, Valerio Perrone, Jeffrey P Spence, Paul A Jenkins, Sara Mathieson, and Yun S Song. A likelihood-free inference framework for population genetic data using exchangeable neural networks. *Advances in Neural Information Processing Systems*, 31:8594, 2018.

[29] Tao Cheng, James Haworth, and Jiaqiu Wang. Spatio-temporal autocorrelation of road network data. *Journal of Geographical Systems*, 14(4):389–413, 2012.

[30] Tao Cheng, Jiaqiu Wang, James Haworth, Benjamin Heydecker, and Andy Chow. A dynamic spatial weight matrix and localized space–time autoregressive integrated moving average for network modeling. *Geographical Analysis*, 46(1):75–97, 2014.

[31] Irina Chis Ster and Neil M Ferguson. Transmission parameters of the 2001 foot and mouth epidemic in Great Britain. *PloS ONE*, 2(6):e502, 2007.

[32] G Chowell, CE Ammon, NW Hengartner, and JM Hyman. Transmission dynamics of the great influenza pandemic of 1918 in Geneva, Switzerland: Assessing the effects of hypothetical interventions. *Journal of Theoretical Biology*, 241(2):193–204, 2006.

[33] Gerardo Chowell, Paul W Fenimore, Melissa A Castillo-Garsow, and Carlos Castillo-Chavez. SARS

outbreaks in Ontario, Hong Kong and Singapore: the role of diagnosis and isolation as a control mechanism. *Journal of Theoretical Biology*, 224(1):1–8, 2003.

[34] Gerardo Chowell, Hiroshi Nishiura, and Luis MA Bettencourt. Comparative estimation of the reproduction number for pandemic influenza from daily case notification data. *Journal of the Royal Society Interface*, 4(12):155–166, 2007.

[35] 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.

[36] Derek AT Cummings, Rafael A Irizarry, Norden E Huang, Timothy P Endy, Ananda Nisalak, Kumnuan Ungchusak, and Donald S Burke. Travelling waves in the occurrence of dengue haemorrhagic fever in Thailand. *Nature*, 427(6972):344–347, 2004.

[37] Rob Deardon, Stephen P Brooks, Bryan T Grenfell, Matthew J Keeling, Michael J Tildesley, Nicholas J Savill, Darren J Shaw, and Mark EJ Woolhouse. Inference for individual-level models of infectious diseases in large populations. *Statistica Sinica*, 20(1):239, 2010.

[38] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22(5):1009–1020, 2012.

[39] Arnaud Doucet, Nando De Freitas, and Neil James Gordon. *Sequential Monte Carlo Methods in Practice*, volume 1. Springer, 2001.

[40] Bedilu Alamirie Ejigu and Eshetu Wencheko. Introducing covariate dependent weighting matrices in fitting autoregressive models and measuring spatio-environmental autocorrelation. *Spatial Statistics*, 38, 2020.

[41] Iniobong Ekong, Emeka Chukwu, and Martha Chukwu. Covid-19 mobile positioning data contact tracing and patient privacy regulations: exploratory search of global response strategies and the use of digital tools in Nigeria. *JMIR mHealth and uHealth*, 8(4):e19139, 2020.

[42] Inger Fabris-Rotelli, Jenny Holloway, Zaid Kimmie, Sally Archibald, Pravesh Debba, Raeesa Docrat, Alize le Roux, Nontembeko Dudeni-Tlhone, Charl Janse Van Rensburg, Renate Thiede, Nada Abdelatiff, Arminn Potgieter, and Sibusisiwe Makhanya. A spatial SEIR model for COVID-19 in South Africa. 2021.

[43] Paul Fearnhead and Dennis Prangle. Constructing summary statistics for approximate Bayesian

computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):419–474, 2012.

[44] Flavio Finger, Tina Genolet, Lorenzo Mari, Guillaume Constantin de Magny, Noël Magloire Manga, Andrea Rinaldo, and Enrico Bertuzzo. Mobile phone data highlights the role of mass gatherings in the spreading of cholera outbreaks. *Proceedings of the National Academy of Sciences*, 113(23):6421–6426, 2016.

[45] A. S. Fotheringham, M. Charlton, and C. Brundson. The geography of parameter space: an investigation of spatial non-stationarity. *Geographical Information Systems*, 10(5):605–627, 1996.

[46] Yun-Xin Fu and Wen-Hsiung Li. Estimating the age of the common ancestor of a sample of DNA sequences. *Molecular Biology and Evolution*, 14(2):195–199, 1997.

[47] Tapiwa Ganyani, Christel Faes, Gerardo Chowell, and Niel Hens. Assessing inference of the basic reproduction number in an SIR model incorporating a growth-scaling parameter. *Statistics in Medicine*, 37(29):4490–4506, 2018.

[48] Song Gao, Jinmeng Rao, Yuhao Kang, Yunlei Liang, Jake Kruse, Dorte Dopfer, Ajay K Sethi, Juan Francisco Mandujano Reyes, Brian S Yandell, and Jonathan A Patz. Association of mobile phone location data indications of travel and stay-at-home mandates with Covid-19 infection rates in the US. *JAMA Network Open*, 3(9):e2020485–e2020485, 2020.

[49] Salisu M Garba, Jean M-S Lubuma, and Berge Tsanou. Modeling the transmission dynamics of the COVID-19 pandemic in South Africa. *Mathematical Biosciences*, 328:108441, 2020.

[50] Arthur Getis and Jared Aldstadt. Constructing the spatial weights matrix using a local statistic. *Geographical Analysis*, 36(2), May 2004.

[51] Arthur Getis and J Keith Ord. The analysis of spatial association by use of distance statistics. In *Perspectives on Spatial Data Analysis*, pages 127–145. Springer, 2010.

[52] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016.

[53] Kyra H Grantz, Hannah R Meredith, Derek AT Cummings, C Jessica E Metcalf, Bryan T Grenfell, John R Giles, Shruti Mehta, Sunil Solomon, Alain Labrique, Nishant Kishore, Caroline O Buckee, and Amy Wesolowski. The use of mobile phone data to inform analysis of COVID-19 pandemic epidemiology. *Nature Communications*, 11(1):1–8, 2020.

[54] Mevin B Hooten, Jessica Anderson, and Lance A Waller. Assessing North American influenza dynamics with a statistical SIRS model. *Spatial and Spatio-Temporal Epidemiology*, 1(2-3):177–185, 2010.

[55] Haijun Hu, Xupu Yuan, Lihong Huang, and Chuangxia Huang. Global dynamics of an SIRS model with demographics and transfer from infectious to susceptible on heterogeneous networks. *Mathematical Biosciences and Engineering*, 16(5):5729–5749, 2019.

[56] Rui Huang, Miao Liu, and Yongmei Ding. Spatial-temporal distribution of COVID-19 in China and its prediction: A data-driven modeling analysis. *The Journal of Infection in Developing Countries*, 14(03):246–253, 2020.

[57] Jayson S Jia, Xin Lu, Yun Yuan, Ge Xu, Jianmin Jia, and Nicholas A Christakis. Population flow drives spatio-temporal distribution of COVID-19 in China. *Nature*, 582(7812):389–394, 2020.

[58] Bai Jiang, Tung-yu Wu, Charles Zheng, and Wing H Wong. Learning summary statistic for approximate Bayesian computation via deep neural network. *Statistica Sinica*, pages 1595–1618, 2017.

[59] Youngji Jo, Lise Jamieson, Ijeoma Edoka, Lawrence Long, Sheetal Silal, Juliet RC Pulliam, Harry Moultrie, Ian Sanne, Gesine Meyer-Rath, and Brooke E Nichols. Cost-effectiveness of remdesivir and dexamethasone for COVID-19 treatment in South Africa. In *Open Forum Infectious Diseases*, volume 8, page ofab040. Oxford University Press US, 2021.

[60] Paul Joyce and Paul Marjoram. Approximately sufficient statistics and Bayesian computation. *Statistical Applications in Genetics and Molecular Biology*, 7(1), 2008.

[61] David G Kendall. Deterministic and stochastic epidemics in closed populations. In *Contributions to Biology and Problems of Health*, pages 149–166. University of California Press, 2020.

[62] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772):700–721, 1927.

[63] Ivan Korolev. Identification and estimation of the SEIRD epidemic model for COVID-19. *Binghamton University. http://dx. doi. org/10.2139/ssrn*, 3569367, 2020.

[64] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.

[65] Vlad Krotov, Leigh Johnson, and Leiser Silva. Tutorial: Legality and ethics of web scraping. *Communications of the Association for Information Systems*, 47(1):22, 2020.

[66] Vlad Krotov and Leiser Silva. Legality and ethics of web scraping. In *Twenty-fourth Americas Conference on Information Systems*, New Orleans, 2018.

[67] Toshikazu Kuniya. Prediction of the epidemic peak of coronavirus disease in Japan, 2020. *Journal of Clinical Medicine*, 9(3):789, 2020.

[68] Theodore Kypraios, Peter Neal, and Dennis Prangle. A tutorial introduction to Bayesian inference for stochastic epidemic models using approximate Bayesian computation. *Mathematical Biosciences*, 287:42–53, 2017.

[69] Lesley Le Grange. Covid-19 pandemic and the prospects of education in South Africa. *Prospects*, pages 1–12, 2020.

[70] Lung-fei Lee and Jihai Yu. Some recent developments in spatial panel data models. *Regional Science and Urban Economics*, 40(5):255–271, 2010.

[71] Roger Th AJ Leenders. Modeling social influence through network autocorrelation: constructing the weight matrix. *Social Networks*, 24(1):21–47, 2002.

[72] Phenyo E Lekone and Bärbel F Finkenstädt. Statistical inference in a stochastic epidemic SEIR model with control intervention: Ebola as a case study. *Biometrics*, 62(4):1170–1177, 2006.

[73] James LeSage and R. Kelly Pace. *Introduction to Spatial Econometrics*. Chapman and Hall/CRC, first edition, 2009.

[74] Benjamin F Maier and Dirk Brockmann. Effective containment explains subexponential growth in recent confirmed COVID-19 cases in China. *Science*, 368(6492):742–746, 2020.

[75] Rajat Malik, Rob Deardon, and Grace PS Kwong. Parameterizing spatial models of infectious disease transmission that incorporate infection time uncertainty using sampling-based likelihood approximations. *PloS ONE*, 11(1):e0146253, 2016.

[76] Jean-Michel Marin, Pierre Pudlo, Christian P Robert, and Robin J Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

[77] 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.

[78] Pierre-Alexandre Mattei and Samuel Wiqvist. Partially exchangeable networks and architectures for learning summary statistics in approximate bayesian computation.

[79] Rendani Mbuvha and Tshilidzi Marwala. Bayesian inference of COVID-19 spreading rates in South Africa. *PloS ONE*, 15(8):e0237126, 2020.

[80] Rendani Mbuvha and Tshilidzi Marwala. Bayesian inference of COVID-19 spreading rates in South Africa. *PloS one*, 15(8):e0237126, 2020.

[81] Trevelyan McKinley, Alex R Cook, and Robert Deardon. Inference in epidemic models without likelihoods. *The International Journal of Biostatistics*, 5(1), 2009.

[82] Trevelyan J McKinley, Joshua V Ross, Rob Deardon, and Alex R Cook. Simulation-based Bayesian inference for epidemic models. *Computational Statistics & Data Analysis*, 71:434–447, 2014.

[83] Trevelyan J McKinley, Ian Vernon, Ioannis Andrianakis, Nicky McCreesh, Jeremy E Oakley, Rebecca N Nsubuga, Michael Goldstein, and Richard G White. Approximate Bayesian computation and simulation-based inference for complex stochastic epidemic models. *Statistical science*, 33(1):4–18, 2018.

[84] Miryam S Merk and Philipp Otto. Estimation of the spatial weighting matrix for regular lattice data–an adaptive lasso approach with cross-sectional resampling. *arXiv preprint arXiv:2001.01532*, 2020.

[85] David Mhlanga and Tankiso Moloi. COVID-19 and the digital transformation of education: What are we learning on 4IR in South Africa? *Education sciences*, 10(7):180, 2020.

[86] AG M'Kendrick. Applications of mathematics to medical problems. *Proceedings of the Edinburgh Mathematical Society*, 44:98–130, 1925.

[87] Zindoga Mukandavire, Farai Nyabadza, Noble J Malunguza, Diego F Cuadros, Tinevimbo Shiri, and Godfrey Musuka. Quantifying early COVID-19 outbreak transmission in South Africa and exploring vaccine efficacy scenarios. *PloS ONE*, 15(7):e0236003, 2020.

[88] Kevin P Murphy. *Machine learning: a probabilistic perspective.* MIT press, 2012.

[89] Thirusha Naidu. The COVID-19 pandemic in South Africa. *Psychological Trauma: Theory, Research, Practice, and Policy*, 12(5):559, 2020.

[90] Brooke E Nichols, Lise Jamieson, Sabrina RC Zhang, Gabriella A Rao, Sheetal Silal, Juliet RC Pulliam, Ian Sanne, and Gesine Meyer-Rath. The role of Remdesivir in South Africa: Preventing COVID-19 deaths through increasing intensive care unit capacity. *Clinical Infectious Diseases*, 72(9):1642–1644, 2021.

[91] Nuria Oliver, Bruno Lepri, Harald Sterly, Renaud Lambiotte, Sébastien Deletaille, Marco De Nadai, Emmanuel Letouzé, Albert Ali Salah, Richard Benjamins, Ciro Cattuto, et al. Mobile phone data for informing public health actions across the COVID-19 pandemic life cycle, 2020.

[92] J Keith Ord and Arthur Getis. Local spatial autocorrelation statistics: distributional issues and an application. *Geographical analysis*, 27(4):286–306, 1995.

[93] Amenaghawon C Osemwinyen and Aboubakary Diakhaby. Mathematical modelling of the transmission dynamics of Ebola virus. *Applied and Computational Mathematics*, 4(4):313–320, 2015.

[94] Dominique Peeters and Isabelle Thomas. Network autocorrelation. *Geographical Analysis*, 41(4):436–443, 2009.

[95] Pedro S Peixoto, Diego Marcondes, Cláudia Peixoto, and Sérgio M Oliva. Modeling future spread of infections via mobile geolocation data and population dynamics. an application to COVID-19 in Brazil. *PloS ONE*, 15(7):e0235732, 2020.

[96] Danny Pfeffermann et al. New important developments in small area estimation. *Statistical Science*, 28(1):40–68, 2013.

[97] Elena Loli Piccolomini and Fabiana Zama. Preliminary analysis of COVID-19 spread in Italy with an adaptive SEIRD model. *arXiv preprint arXiv:2003.09909*, 2020.

[98] Aaron T Porter and Jacob J Oleson. A path-specific SEIR model for use with general latent and infectious time distributions. *Biometrics*, 69(1):101–108, 2013.

[99] Aaron T Porter and Jacob J Oleson. A spatial epidemic model for disease spread over a heterogeneous spatial support. *Statistics in Medicine*, 35(5):721–733, 2016.

[100] Arminn Potgieter, Inger Fabris-Rotelli, Zaid Kimmie, Nontembeko Dudeni-Tlhone, Jenny Holloway, Charl Janse Van Rensburg, Renate Thiede, Pravesh Debba, Raeesa Docrat, Nada Abdelatif, et al.

Modelling representative population mobility for COVID-19 spatial transmission in South Africa. 2021.

[101] Mark P Pritchard, Mark E Havitz, and Dennis R Howard. Analyzing the commitment-loyalty link in service contexts. *Journal of the academy of marketing science*, 27(3):333–348, 1999.

[102] Xi Qu, Lung-fei Lee, and Jihai Yu. Qml estimation of spatial dynamic panel data models with endogenous time varying spatial weights matrices. *Journal of econometrics*, 197(2):173–201, 2017.

[103] Oliver Ratmann, Christophe Andrieu, Carsten Wiuf, and Sylvia Richardson. Model criticism based on likelihood-free inference, with an application to protein network evolution. *Proceedings of the National Academy of Sciences*, 106(26):10576–10581, 2009.

[104] Oliver Ratmann, Anton Camacho, Adam Meijer, and Gé Donker. Statistical modelling of summary values leads to accurate approximate Bayesian computations. *arXiv preprint arXiv:1305.4283*, 2013.

[105] Jonathan M Read, Jessica RE Bridgen, Derek AT Cummings, Antonia Ho, and Chris P Jewell. Novel coronavirus 2019-nCoV: early estimation of epidemiological parameters and epidemic predictions. *MedRxiv*, 2020.

[106] Stuart Rennie, Mara Buchbinder, Eric Juengst, Lauren Brinkley-Rubinstein, Colleen Blue, and David L Rosen. Scraping the web for public health gains: Ethical considerations from a 'big data' research project on HIV and incarceration. *Public Health Ethics*, 13(1):111–121, 2020.

[107] Christian P Robert, Jean-Marie Cornuet, Jean-Michel Marin, and Natesh S Pillai. Lack of confidence in approximate Bayesian computation model choice. *Proceedings of the National Academy of Sciences*, 108(37):15112–15117, 2011.

[108] Ronald Ross. An application of the theory of probabilities to the study of a priori pathometry.-part i. *Proceedings of the Royal Society of London. Series A, Containing papers of a mathematical and physical character*, 92(638):204–230, 1916.

[109] Ronald Ross and Hilda P Hudson. An application of the theory of probabilities to the study of a priori pathometry.-part ii. *Proceedings of the Royal Society of London. Series A, Containing papers of a mathematical and physical character*, 93(650):212–225, 1917.

[110] Donald B Rubin. Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, pages 1151–1172, 1984.

[111] Nick W Ruktanonchai, Patrick DeLeenheer, Andrew J Tatem, Victor A Alegana, T Trevor Caughlin, Elisabeth zu Erbach-Schoenberg, Christopher Lourenço, Corrine W Ruktanonchai, and David L Smith. Identifying malaria transmission foci for elimination using human mobility data. *PLoS Computational Biology*, 12(4):e1004846, 2016.

[112] Benjamin Sakarovitch, Marie-Pierre de Bellefon, Pauline Givord, and Maarten Vanhoof. Estimating the residential population from mobile phone data, an initial exploration. *Economie et Statistique*, 505(1):109–132, 2018.

[113] B Sartorius, AB Lawson, and RL Pullan. Modelling and predicting the spatio-temporal spread of COVID-19, associated deaths and impact of key risk factors in England. *Scientific reports*, 11(1):1–11, 2021.

[114] Oliver Schabenberger and Carol A Gotway. *Statistical methods for spatial data analysis*. CRC press, 2017.

[115] Malte Schröder, Andreas Bossert, Moritz Kersting, Sebastian Aeffner, Justin Coetzee, Marc Timme, and Jan Schlüter. COVID-19 in South Africa: outbreak despite interventions. *Scientific Reports*, 11(1):1–9, 2021.

[116] Tom Seymour, Dean Frantsvog, Satheesh Kumar, et al. History of search engines. *International Journal of Management & Information Systems (IJMIS)*, 15(4):47–58, 2011.

[117] Daniel Shriner, Yi Liu, David C Nickle, and James I Mullins. Evolution of intrahost HIV-1 genetic diversity during chronic infection. *Evolution*, 60(6):1165–1176, 2006.

[118] Sheetal Silal, Juliet Pulliam, Gesine Meyer-Rath, Brooke Nichols, Lise Jamieson, Zaid Kimmie, and Harry Moultrie. Estimating cases for COVID-19 in South Africa update: 19 May 2020. *Update*, 2020.

[119] Daniel Silk, Saran Filippi, and Michael PH Stumpf. Optimizing threshold-schedules for approximate Bayesian computation sequential Monte Carlo samplers: applications to molecular systems. *arXiv preprint arXiv:1210.3296*, 2012.

[120] Scott A Sisson, Yanan Fan, and Mark M Tanaka. Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007.

[121] Stanislav Stakhovych and Tammo H. A. Bijmolt. Specification of spatial models: A simulation study on weights matrices. *Papers in Regional Science*, 88(2):389–408, June 2008.

[122] Irina Chis Ster, Brajendra K Singh, and Neil M Ferguson. Epidemiological inference for partially observed epidemics: the example of the 2001 foot and mouth epidemic in Great Britain. *Epidemics*, 1(1):21–34, 2009.

[123] Nancy Stiegler and Jean-Pierre Bouchard. South Africa: Challenges and successes of the COVID-19 lockdown. In *Annales Médico-psychologiques, revue psychiatrique*, volume 178, pages 695–698. Elsevier, 2020.

[124] Mikael Sunnåker, Alberto Giovanni Busetto, Elina Numminen, Jukka Corander, Matthieu Foll, and Christophe Dessimoz. Approximate Bayesian computation. *PLoS Comput Biol*, 9(1):e1002803, 2013.

[125] E Tagliazucchi, Pablo Balenzuela, M Travizano, GB Mindlin, and Pablo Daniel Mininni. Lessons from being challenged by COVID-19. *Chaos, Solitons & Fractals*, 137:109923, 2020.

[126] Mark Tanaka, Andrew Francis, Fabio Luciani, and Scott Sisson. Estimating tuberculosis transmission parameters from genotype data using approximate Bayesian computation. *Genetics*, 2006.

[127] Simon Tavaré, David J Balding, Robert C Griffiths, and Peter Donnelly. Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518, 1997.

[128] Renate Thiede, Nada Abdelatif, Inger Fabris-Rotelli, Raeesa Manjoo-Docrat, Jennifer Holloway, Charl Janse van Rensburg, Pravesh Debba, Nontembeko Dudeni-Tlhone, Zaid Kimmie, and Alize le Roux. Spatial variation in the basic reproduction number of COVID-19: A systematic review. *arXiv preprint arXiv:2012.06301*, 2020.

[129] W. R. Tobler. A computer movie simulating urban growth in the Detroit region. *Economic Geography*, 46:234–240, 1970.

[130] 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.

[131] Don Van Ravenzwaaij, Pete Cassey, and Scott D Brown. A simple introduction to Markov chain Monte–Carlo sampling. *Psychonomic bulletin & review*, 25(1):143–154, 2018.

[132] Thomas Varsavsky, Mark S Graham, Liane S Canas, Sajaysurya Ganesh, Joan Capdevila Pujol,

Carole H Sudre, Benjamin Murray, Marc Modat, M Jorge Cardoso, Christina M Astley, et al. Detecting COVID-19 infection hotspots in England using large-scale self-reported data from a mobile application: a prospective, observational study. *The Lancet Public Health*, 6(1):e21–e29, 2021.

[133] Jay M Ver Hoef, Noel AC Cressie, and David C Glenn-Lewin. Spatial models for spatial statistics: some unification. *Journal of Vegetation Science*, 4(4):441–452, 1993.

[134] Vincenzina Vitale, Pierpaolo D'Urso, and Livia De Giovanni. Spatio-temporal object-oriented bayesian network modeling of the Covid-19 Italian outbreak data. *Spatial Statistics*, page 100529, 2021.

[135] Eric-Jan Wagenmakers, Michael Lee, Tom Lodewyckx, and Geoffrey J Iverson. Bayesian versus frequentist inference. In *Bayesian evaluation of informative hypotheses*, pages 181–207. Springer, 2008.

[136] Gunter Weiss and Arndt von Haeseler. Inference of population history using a likelihood approach. *Genetics*, 149(3):1539–1546, 1998.

[137] Amy Wesolowski, Nathan Eagle, Andrew J Tatem, David L Smith, Abdisalan M Noor, Robert W Snow, and Caroline O Buckee. Quantifying the impact of human mobility on malaria. *Science*, 338(6104):267–270, 2012.

[138] Amy Wesolowski, Taimur Qureshi, Maciej F Boni, Pål Roe Sundsøy, Michael A Johansson, Syed Basit Rasheed, Kenth Engø-Monsen, and Caroline O Buckee. Impact of human mobility on the emergence of dengue epidemics in Pakistan. *Proceedings of the National Academy of Sciences*, 112(38):11887–11892, 2015.

[139] Bernard Widrow and Michael A Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442, 1990.

[140] Bo Zhao. Web scraping. *Encyclopedia of big data*, pages 1–3, 2017.

[141] Zebin Zhao, Xin Li, Feng Liu, Gaofeng Zhu, Chunfeng Ma, and Liangxu Wang. Prediction of the COVID-19 spread in African countries and implications for prevention and control: A case study in South Africa, Egypt, Algeria, Nigeria, Senegal and Kenya. *Science of the Total Environment*, 729:138959, 2020.

[142] Ying Zhou, Renzhe Xu, Dongsheng Hu, Yang Yue, Qingquan Li, and Jizhe Xia. Effects of human

mobility restrictions on the spread of COVID-19 in Shenzhen, China: a modelling study using mobile phone data. *The Lancet Digital Health*, 2(8):e417–e424, 2020.