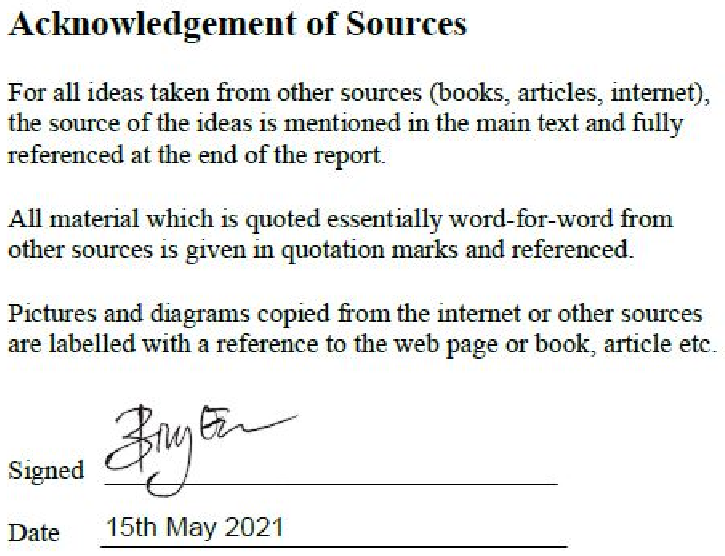
University of BRISTOL

Bayesian Epidemic Modelling

Bing En Gan

Supervised by Daniel Lawson
Level H/6
20 Credit Points

May 16, 2021



Acknowledgement of Sources

For all ideas taken from other sources (books, articles, internet), the source of the ideas is mentioned in the main text and fully referenced at the end of the report.

All material which is quoted essentially word-for-word from other sources is given in quotation marks and referenced.

Pictures and diagrams copied from the internet or other sources are labelled with a reference to the web page or book, article etc.

Signed

Date 15th May 2021

Abstract

In this research project, we study the Susceptible-Infected-Recovered (SIR) compartmental epidemic model in its deterministic and stochastic form in a well-mixed population. The likelihood functions for these models can be intractable or tedious to obtain. For that reason, we study a simulation-based inference method known as approximate Bayesian computation (ABC) to draw inference from these models. Specifically, the computational techniques of ABC is studied in detail to increase computation efficiency. The epidemic models are fitted to Covid-19 data in the UK and are compared against each other.

Contents

1 Introduction

Mathematical modelling is important in epidemiology to understand and predict the spread of infectious diseases. This will allow responsible bodies to make informed decisions to contain the spread of the disease, e.g. through non-pharmaceutical interventions. However, in order to do this, we need to fit the epidemic models to our observed data which can be extremely challenging due to the complexity of the model. Furthermore, the incompleteness of epidemic data makes the modelling process more difficult. Fortunately, one common feature about most epidemic models is that they are relatively easy to sample from. Therefore, we can apply a simulation-based inference method called approximate Bayesian computation (ABC) to draw inference. Like the name suggests, ABC is derived from Bayesian methods whereby the likelihood function is "approximated". We will study more about ABC in a later section.

One of the most common epidemic models are compartmental models which divide the population into different compartments and any individual in the population can only be in one compartment at a time. Additionally, "epidemic models can be classified into various categories depending on their treatment of variability, chance and uncertainty (deterministic or stochastic), time (continuous or discrete intervals), space (non-spatial or spatial) and the structure of the population (homogenous or heterogeneous mixing)." [1] In this research project, we are interested in comparing the differences between a deterministic and a stochastic epidemic model. Therefore, we will be building deterministic and stochastic SIR models and fitting them to Covid-19 data in the UK through ABC methods.

2 SIR Compartmental Epidemic Model

Throughout the history of humankind, numerous infectious disease outbreaks have plagued a great many civilizations and nations, resulting in the death of countless lives. [2] To illustrate the severity of disease outbreaks, the Black Death [3] which happened in the mid 1300s has wiped out almost one third of the European population while the Spanish flu [4] of 1918 was estimated to have caused 20 to 50 million deaths. Even though there has been rapid advancements in technological and theoretical areas of science, it is difficult to predict when a pandemic will arise. Pivotal questions such as how the disease will spread, mortality rate of the disease, types of governmental interventions to be applied to stop the spread remain unanswered until a pandemic has actually happened. This is largely due to the continuous mutations of viruses coupled with the complex transmission mechanism of viruses. For instance, the ongoing Covid-19 pandemic has taken about 2.9 million lives as of 9th of April 2021 [5], and has caused lasting social [6], educational and economic [7] damage to the global population. Therefore, on one hand, there is a need for scientists to focus on vaccine development. On the other hand, there is a need for mathematical models to make swift and accurate assessment on the spread of the pandemic to aid the government and other responsible bodies in the decision-making process.

The very first published work on mathematical models for epidemics was by Daniel Bernoulli in 1766. In his work, Bernoulli developed a mathematical model to analyse the death rate due to smallpox in England. A review and translation of the paper in English can be found at [8]. However, mathematical models of epidemic has not been developed rigorously until the work of Ross in 1911 [9] on Malaria which eventually became the foundation for modern mathematical epidemiology. Following that, McKendrick and Kermack [10] developed the deterministic compartmental epidemic models which will be the foundation of the model that we will be using in this paper.

The compartmental model that will be used in this paper is the Susceptible-Infected-Recovered (SIR) model. The idea of the SIR model is to divide the population into three compartments: "Susceptible"(S), "Infected"(I), and "Recovered"(R). We assume that individuals move from one compartment to the other in the order S → I → R. There are two main ways to run this model, one of them is by using a deterministic framework, and the other a stochastic framework. A deterministic compartmental model is usually governed by a set of ordinary differential equations (ODE) while the stochastic compartmental model can be run with different methods such as Markov chains or stochastic differential equations (SDE) as described in [11] but in this paper we will be using a chain binomial model. Deterministic models are deterministic in nature and always give the same output given the same input whereas stochastic models are random in nature and generate different outputs given the same input. These deterministic and stochastic models will be discussed further in Section 2.1 and 2.2 respectively. The general way that individuals would move in the SIR model is described in figure 1. When a susceptible individual gets infected by the disease, they will be infectious and will move to the I compartment. Lastly, they will move to the R compartment when they have recovered or died and will remain immune to the disease.("R" also known as Removed) At time t, the number of people in each compartment can be represented as a function of time, i.e. $(S(t), I(t), R(t))$. This is just one of the many compartmental epidemic models that were developed. For example, the SI (Susceptible-Infected) model only comprises of two compartments whereas the SEIR model has an additional "Exposed" compartment which represents individuals who are exposed to the disease but has yet to show symptoms.

Assumptions

In this part, we will outline certain assumptions about the SIR model. The accuracy of the model can only be as good as the assumptions they are based on. As the real world has a very complex structure, to simulate the real world as close as possible, certain assumptions that will be outlined will have to change. The assumptions of the SIR model are as follows:

- The number of people in the population is assumed to be constant throughout the epidemic. This can often be justified as epidemics do not last long

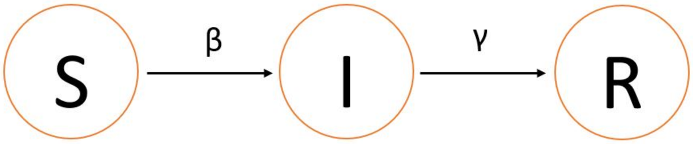


Figure 1: Flow diagram of the SIR model. β represents the rate at which a successful infectious contact occurs per unit time, whereas γ is the rate of recovery of the infected individuals.

enough that would cause significant change in the population size. (e.g. population growth due to birth rate) So, at any time t, $S(t)+I(t)+R(t) = N$, where N is the population size. Also, it is important to note that the R compartment includes people who have recovered and people who died from the disease.

- It is assumed that the number of people who died naturally is negligible when compared to the total population.
- Homogeneous mixing of the population is assumed. We assume that each person in the population make contact at random. This assumption is usually not justified due to the complex social structure of society. For example, most Bristolians only meet other Bristolians. Furthermore, there are many subcommunities in Bristol, which further complicates the situation.

Although it is more realistic to model the Covid-19 pandemic with a SEIR type model because the virus has an incubation period before the infected starts showing symptoms, our research goal is not to build an epidemic model that best reflect the current Covid-19 situation. Our main goal is to understand how to perform parameter inference on a epidemic model and to compare the SIR model under the deterministic and stochastic framework. Therefore, we will be implementing the SIR model to Covid-19 data despite its limitations.

2.1 Deterministic Framework

If we assume homogeneous mixing of the population, then individuals in the population move at random, so we can assume that each infected individual has a rate of contact c, where c is the number of people contacted per unit time. Since not all contacts will lead to an increase in the number of infected, we let the probability of an effective infectious contact be p. Then each infected individual infects an average of $p \times c \times S/N$ individuals per unit time. So, $p \times c \times I \times S/N$ susceptible individuals will be infected per unit time. We can let $\beta = p \times c$ be the rate of susceptible individuals becoming infected per unit time, also known as the transmission rate. Moreover, we will introduce γ as the recovery rate of the infected individuals. The following set of differential equations are referred to as the SIR deterministic model.

$$\begin{aligned} \frac{dS}{dt} &= -\frac{\beta S(t)I(t)}{N} \\ \frac{dI}{dt} &= \frac{\beta S(t)I(t)}{N} - \gamma I(t) \\ \frac{dR}{dt} &= \gamma I(t) \end{aligned} \tag{1}$$

In epidemiology, the basic reproduction number, often referred to as R_0, is used to quantify the rate of spread of an epidemic. It is defined as the expected number of secondary cases produced by an infected case in a population where everyone is susceptible to the disease. For example, if $R_0 = 3$, then we expect every new case of disease to produce 3 secondary cases on average. For deterministic SIR models, we can calculate the basic reproduction number using the following ratio $R_0 = \frac{\beta}{\gamma}$. If $R_0 > 1$, then an infected infects on average more than one person and hence a pandemic will happen. If $R_0 < 1$, then an infected infects less than one person on average and so the disease will die out. If $R_0 = 1$, the disease will become endemic [12], i.e. the disease will spread through the population but number of infected will not increase or decrease.

2.2 Stochastic Framework

The deterministic model essentially assumes that all individuals from the same compartment behave the same way. In fact, different individuals have different diet, living environment, or even genetic traits, so their bodies may react differently to the disease. Hence, we can think of the spread of disease as a random process. Although for large populations, randomness from individual to individual eventually averages out, making stochastic modelling crucial for small populations. (ref from [10]) Additionally, from an estimation point of view, stochastic models allow us to quantify the uncertainty of the parameters from the data because every simulation run of a stochastic model is different.

Stochastic epidemic models are mostly based on Markov processes. Markov process is defined as a random process where the future state of the process only depends on the current state, i.e. $\mathbb{P}(Y_{t+1}|Y_t, Y_{t-1}, ..., Y_0) = \mathbb{P}(Y_{t+1}|Y_t)$. To have a better understanding, suppose we have two Poisson processes, $I_p(t)$ and $R_p(t)$, where they represent the number of individuals who have been infected and the number of individuals who recovered at time t respectively. Then, for $h > 0$ and $\{S(t), I(t), R(t) : t \geq 0\}$, the following infinitesimal increment probabilities hold:

$$\begin{aligned} \mathbb{P}\big(I_p(t+h) - I_p(t) = 1|I_p(t)\big) &= \frac{\beta S(t)I(t)}{N}h + o(h) \\ \mathbb{P}\big(R_p(t+h) - R_p(t) = 1|R_p(t)\big) &= \gamma I(t)h + o(h) \end{aligned} \tag{2}$$

where N is the population size and $o(h)$ tends to zero as h approaches zero. Therefore, we have that new infections and recoveries occur at a Poisson process with rate $\beta S(t)I(t)/N$ and $\gamma I(t)$ respectively. It is easy to show that for Poisson processes, the amount of time until the event occurs is exponentially distributed. So, using this information, we can construct an algorithm to simulate the stochastic model. The algorithm is called the Gillespie algorithm [13] and one should note that the algorithm is exact and not an approximation. The algorithm is as follows:

1. Get $\{S(t), I(t), R(t)\}$ for time t.
2. Calculate $r_1 = \beta S(t)I(t)/N$ and $r_2 = \gamma I(t)$.
3. Simulate time h until next event happens, i.e. $h \sim \text{Exp}(r_1 + r_2)$.
4. Simulate which of the event happened by using $\mathbb{P}(\text{Event} = i) = \frac{r_i}{r_1+r_2}$, where $i \in \{1, 2\}$, 1 indicating a susceptible is infected, 2 indicating an infected recovered.
5. Update $\{S(t + h), I(t + h), R(t + h)\}$ and set $t = t + h$ and go to step 1.

However, the drawback about this stochastic model is that when the population size N is large, a jump happens very often and therefore it is computationally expensive to run. So, an approximation procedure can be implemented, called the Poisson τ-leaps method. [11] The algorithm goes like this:

1. Set small $\tau > 0$.
2. Get $\{S(t), I(t), R(t)\}$ for time t.
3. Calculate $r_1 = \beta S(t)I(t)/N$ and $r_2 = \gamma I(t)$.
4. Simulate change in each compartment by simulating $R_1 \sim \text{Poisson}(r_1\tau)$ and $R_2 \sim \text{Poisson}(r_2\tau)$.
5. Update $\{S(t + \tau) = S(t) - R_1, I(t + \tau) = I(t) + R_1 - R_2, R(t + \tau) = R(t) + R_2\}$ and set $t = t + \tau$ and return to step 2.

This algorithm performs much faster than the first one and accuracy is achieved for small τ. However, it is possible that some compartments may become negative because the Poisson distribution is unbounded. For instance, say we have 1 individual in the S compartment and 50 infected individuals. Because of the unboundedness of the Poisson distribution, during the next τ time, the Poisson distribution might predict 2 individuals will be infected which results in -1 in the S compartment. To solve this, a Binomial distribution could be used instead of the Poisson distribution. Therefore, we propose the following discrete stochastic model.

$$\begin{aligned} S(t+\tau) &= S(t) - M_I \\ I(t+\tau) &= I(t) + M_I - M_R \\ R(t+\tau) &= R(t) + M_R, \end{aligned} \tag{3}$$

where $M_I \sim \text{Binomial}\big(S(t), 1 - \exp(-\tau\beta I(t)/N)\big)$, $M_R \sim \text{Binomial}\big(I(t), 1 - \exp(-\tau\gamma)\big)$ and $\tau > 0$ is a small change in time as stated above. M_I represents the number of susceptible infected whereas M_R represents the number of infected recovered during that small jump in time. To understand the origin of the probabilities used within the binomial distribution, from the Poisson τ-leaps method above, we have that the probability of having at least one infection within time τ is $\mathbb{P}(T < \tau) = 1 - \exp(-r_1\tau) = 1 - \exp\big(-\tau\beta S(t)I(t)/N\big)$ by the exponential distribution. So the probability of a random individual within the S compartment to be infected within time τ is $1 - \exp\big(-\tau\beta I(t)/N\big)$. A similar explanation can be given for the probability $1 - \exp(-\tau\gamma)$.

We have now introduced the model that we will be implementing in our study, but how do we estimate the parameters β and γ? The next section on Approximate Bayesian Computation(ABC) will be the method we will employ to estimate them.

3 Approximate Bayesian Computation (ABC)

3.1 History and Background

"All models are wrong, but some are useful" is an extremely common aphorism in statistics (originated from the statistician George Box [14]). In reality, this phrase is not only used to describe statistical models but any scientific models in general. However, these so called "wrong" statistical models are capable of providing interesting explanation of the real world and accurate predictions from the data. Statistical modelling generally involves the estimation of parameters using data available based on some statistical assumptions. An ideal version of a statistical model would be one that can replicate the data generating process.

In [15], the paper highlighted two main types of statistical models, those that have known likelihood functions, and those that have intractable likelihood but we can simulate samples from. The latter is often known as a generative model. For likelihood-based models, a very established method to perform statistical inference is the maximum likelihood estimation (MLE) [16]. It is very intuitive to apply and its estimated parameters have many desirable properties. On the other hand, one popular way to perform likelihood-free inference is approximate Bayesian computation (ABC). This section aims to introduce the concept behind ABC and the computational techniques to perfrom ABC.

An approximate Bayesian computation algorithm was first proposed by Simon Tavaré [17] in one of his article by using DNA sequence data to infer the posterior distribution of coalescence times. Following that, Jonathan Pritchard [18] did a study on population genetics using the ABC method. Finally, the term approximate Bayesian computation was popularised by Mark Beaumont [19] who further elaborated the ABC methodology specifically in population genetics. Ever since, ABC methods have been widely used in many fields such as epidemiology, environmental modelling, astronomy and systems biology.

Before introducing the algorithms, let us go through the background of ABC.

(references obtained from [20, 21]) Suppose that we have a n-dimensional observed data such that $y_{\text{obs}} = (y_1, y_2, ..., y_n)^T \in \mathcal{Y}$, and we have a model $\mathcal{M}$ such that $\mathcal{M} : \Theta \longrightarrow \mathcal{Y}$ so it takes in a d-dimensional parameter $\theta \in \Theta \subset \mathbb{R}^d$ and outputs simulated data from the parameter θ. Consider θ to be distributed according to a prior density $\pi(\theta)$ and our goal is to obtain the posterior $\pi(\theta|y)$. However, under the ABC context, we assume that we are unable to access the analytic form for $f_n(y|\theta)$ but it is relatively simple to simulate from $f_n(y|\theta)$. The goal of the ABC method is to obtain

$$\pi_\epsilon(\theta, y|y_{obs}) \propto \pi(\theta) f_n(y|\theta) K_\epsilon(||y - y_{\text{obs}}||)$$

where $K_\epsilon(\cdot)$ is a kernel function with scaling parameter ϵ and $||\cdot||$ is a distance metric. The kernel function is a form of a weighting function that assigns weightings on the parameters based on how "close" the simulated data is to the observed given that $||y - y_{\text{obs}}|| \leq \epsilon$. The ABC posterior is then

$$\pi_\epsilon(\theta|y_{\text{obs}}) = \int \pi_\epsilon(\theta, y|y_{obs}) dy.$$

Refer to Figure 2 for an illustration in one-dimensional θ and y.

The advantage of ABC is that it is very easy to formulate an algorithm to sample from the ABC posterior without the need to compute the likelihood function $f_n(y|\theta)$. A simple example of an algorithm can go as follows:

1. Choose a tolerance level $\epsilon > 0$
2. Sample from the prior $\theta_0 \sim \pi(\theta)$
3. Simulate y_0 from the generative model $\mathcal{M}(\theta_0)$ without using the likelihood function
4. Accept θ_0 with probability proportional to $K_\epsilon(||y - y_{\text{obs}}||)$
5. Repeat steps 1 to 3 until a desired M number of sample size is obtained

When y_{obs} is of high dimension, it is often more suitable to project the high dimensional data to a vector of summary statistics which has a smaller dimension, say p, i.e. $S_{obs} = S(y_{obs}) = (S_1, S_2, ..., S_p)$. The euclidean norm, $||y|| = \sqrt{\sum_{i=1}^n y_i^2}$, is often used as the distance metric. There are a few kernel functions that can be used (see figure 3) such as a uniform kernel, an Epanechnikov kernel [23] or a Gaussian kernel but since all of our algorithms use a uniform kernel, we can substitute the kernel function with an indicator function.

$$K_\epsilon(||y - y_{(obs}||) = \mathbb{1}(||y - y_{obs}|| \leq \epsilon)$$

Intuitively, what the ABC method is trying to achieve is to select a set of parameters from the prior which produces simulated data that are "close" to the observed data through the proposed model. In the next section, we will be discussing some regression adjustment techniques [25] that can be applied to improve the convergence of the ABC posterior.

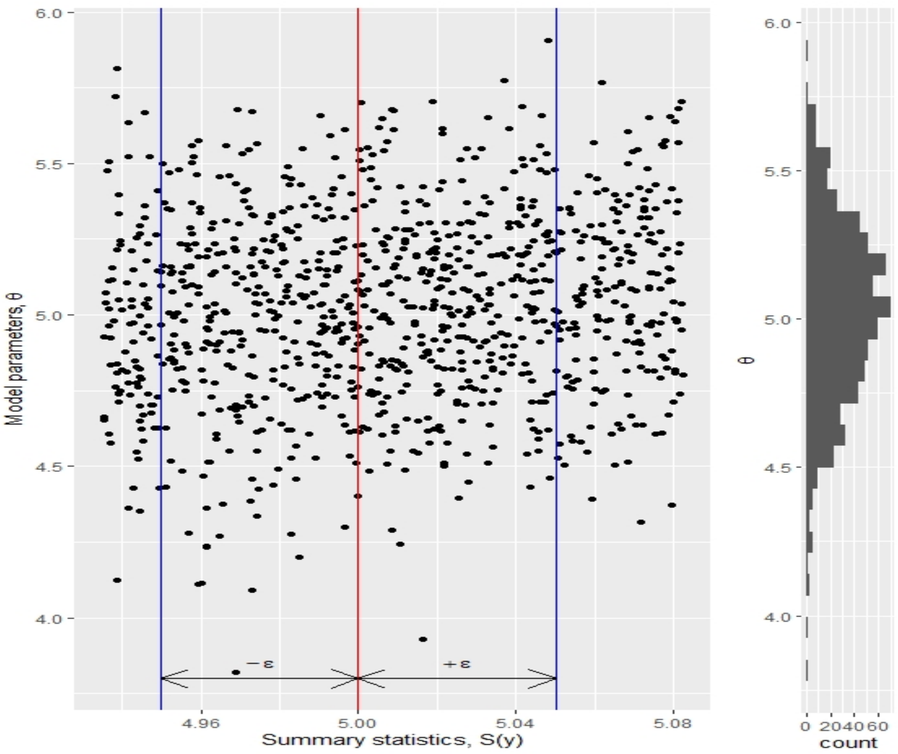


Figure 2: Suppose we have one-dimensional θ and y. Note that we can take $S(\cdot)$ in the plot to be the identity function such that $S(y) = y$ and the red line represents y_{obs}. Each point in the scatter plot represents a (θ, y) draw from the joint distribution $\pi(\theta, y)$. The points which are within the two blue lines, i.e. within the euclidean distance of ϵ are draws from $\pi_\epsilon(\theta, y|y_{\text{obs}})$. Then those θ which corresponds to y with euclidean distance $\leq \epsilon$ forms a marginal distribution which is the ABC posterior. Plot created using ggplot2 [22] in R.

3.2 Regression Adjustment Techniques

The regression adjustment techniques are versatile in the sense that it can be applied to any Monte Carlo method that produces samples from $\pi_\epsilon(\theta, y|y_{obs})$. One of the earliest proposed regression adjustment technique was local-linear regression [19]. Following that, a non-linear regression technique was developed by Blum and Francois in 2010 [26]. In this section, we will focus on the introduction of local-linear regression which is based on the paper from [19].

From the algorithm in Section 3.1, we would have simulated independent pairs of (θ_i, y_i) for $i \in \{1, 2, ..., n\}$. Under Bayes' theorem, we have that $\mathbb{P}(\theta|y) = \mathbb{P}(y|\theta)\mathbb{P}(\theta)/\mathbb{P}(y) = \mathbb{P}(y, \theta)/\mathbb{P}(y)$. From the equation, we can estimate the posterior by dividing $\mathbb{P}(y, \theta)$ by $\mathbb{P}(y)$. The pairs of (θ_i, y_i) are random draws from the joint density, and the rejection method (an ABC algorithm which will

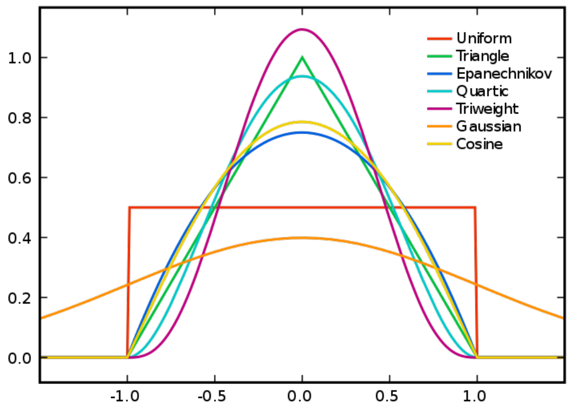


Figure 3: Figure shows a range of different kernel functions. Image obtained from [24].

be elaborated in Section 3.1) is a way to estimate $\mathbb{P}(Y = y)$. The idea of the regression adjustment and weighting proposed in [19] is to first weigh the θ_i according to their distance from the observed, $||y_i - y_{\text{obs}}||$ and then adjust the θ_i by using local-linear regression to diminish the effect of discrepancy between y_i and y_{obs}.

To explain the idea, it would be easier to start with a normal linear regression. Suppose that the following regression model describes the conditional density $\theta|y$,

$$\theta_i = \alpha + (y_i - y_{\text{obs}})^T \beta + \epsilon_i \tag{4}$$

where α is the intercept, β is a vector of coefficients, and ϵ_i are uncorrelated with zero mean and equal variance. Note that the ϵ_i have no distributional assumption and hence θ_i too. So, when $y_i = y_{\text{obs}}$, θ_i are drawn from a distribution with mean $\mathbb{E}(\theta|y = y_{\text{obs}}) = \alpha$. By linear regression literature [27], the least square estimation of α and β can be obtained by minimising

$$\sum_{i=1}^{n} \left(\theta_i - \alpha - (y_i - y_{\text{obs}})^T \beta\right)^2 \tag{5}$$

for which the solution is

$$\begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = (X^T X)^{-1} X^T \theta, \tag{6}$$

where

$$X = \begin{pmatrix} 1 & y_{11} - y_{\text{obs},1} & \cdots & y_{1d} - y_{\text{obs},d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & y_{n1} - y_{\text{obs},1} & \cdots & y_{nd} - y_{\text{obs},d} \end{pmatrix}, \theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_n \end{pmatrix} \tag{7}$$

Now, we can define θ_i^* such that

$$\theta_i^* = \theta_i - (y_i - y_{\text{obs}})^T \hat{\beta}$$

and θ_i^* will be an approximate random sample from the conditional density $\theta|(Y = y_{\text{obs}})$. Note that we can use $\hat{\alpha}$ as a point estimate of θ because $\mathbb{E}(\theta|y = y_{\text{obs}}) = \alpha$ as shown above.

Now we will introduce the notion of local-linear regression. If we recall from Section 3.1, the kernel function applies weightings to the parameters based on how "close" the simulated data are to the observed. Therefore, the linearity assumption of equation 5 is not suitable in general. However, linearity may be applied locally in the neighbourhood of y_{obs}. So, we can replace the minimser as follows, which is called local-linear regression.

$$\sum_{i=1}^{n} \left(\theta_i - \alpha - (y_i - y_{\text{obs}})^T \beta\right)^2 K_\epsilon(||y_i - y_{\text{obs}}||) \tag{8}$$

which can be rewritten as,

$$\sum_{i=1}^{n} \left(\theta_i \sqrt{K_\epsilon} - \alpha\sqrt{K_\epsilon} - (y_i - y_{\text{obs}})^T \beta \sqrt{K_\epsilon}\right)^2 \tag{9}$$

where K_ϵ is the kernel function. To get the solution to equation 8, one can rephrase it as a generalised least squares (GLS) [28] problem because the kernel function outputs value ≥ 0. We can let W be a diagonal matrix with $K_\epsilon(||y_i - y_{\text{obs}}||)$ as the i-th diagonal element and so W is a symmetric and positive definitive matrix. Hence, we can write $W = VV^T$. As we can see from equation 9, we can scale each term by $\sqrt{K_\epsilon(||y_i - y_{\text{obs}}||)}$ to obtain the solution, i.e. $(\hat{\alpha}, \hat{\beta}) = (\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T\tilde{\theta}$, where $\tilde{X} = V^T X$ and $\tilde{\theta} = V^T\theta$. So, we have that

$$\begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = (X^T W X)^{-1} X^T W \theta \tag{10}$$

where W is defined above. The estimate of the posterior mean is then

$$\hat{\alpha} = \frac{\sum_{i=1}^{n} \theta_i^* K_\epsilon(||y_i - y_{\text{obs}}||)}{\sum_{i=1}^{n} K_\epsilon(||y_i - y_{\text{obs}}||)} \tag{11}$$

Note that we can extend the multiple regression approach to using a multiple multivariate regression when θ_i is a vector, in which case β is a matrix while α is a vector. Figure 4 illustrates what the local-linear regression technique does graphically.

Although regression adjustment techniques do improve the convergence of the samples towards the posterior, the results obtained from regression adjustment may be misleading when the model is misspecified. [30] Another problem that might arise due to the regression technique is that θ_i^* may be outside of the support of the prior. For example, θ_i^* may yield a parameter value outside the range of a Uniform$(0, 1)$ prior. In the next section, we will go into the computation techniques revolving ABC and introduce some widely used ABC algorithms.

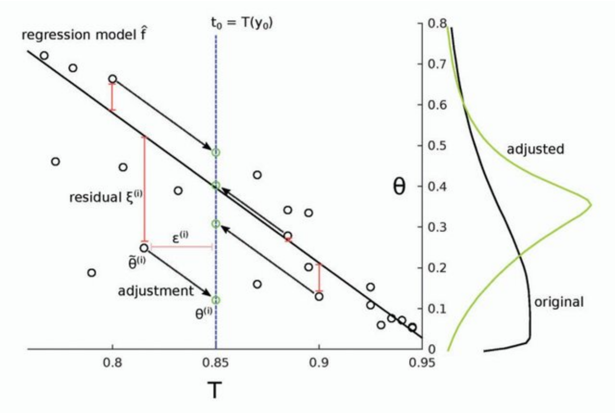


Figure 4: Figure describes the geometrical interpretation of the local-linear regression adjustment technique when the parameters and summary statistics are one-dimensional. The parameters θ_i are adjusted using the regression model as if $y_i = y_{\text{obs}}$. Figure taken from [29].

3.3 Computation Techniques

In this section, we will be talking about 4 different algorithms that we can use to implement ABC. Much research has been done in this area so we will only highlight some of the algorithms that we will be using for our study.

3.3.1 Exact Bayesian Computation

Algorithm 1: Exact Bayesian Computation (EBC)

```
Input: Observed data $y_{\text{obs}}$, prior $\pi(\theta)$, number of samples $N$, model $\mathcal{M}(\cdot)$
Output: $N$ samples from $\pi(\theta|y_{\text{obs}})$
1 Sample $\tilde{\theta}$ from the prior $\pi(\theta)$
2 Simulate $\tilde{y}$ from the model $\mathcal{M}(\tilde{\theta})$
3 Accept $\tilde{\theta}$ if $\tilde{y} = y_{\text{obs}}$ and reject otherwise
4 Repeat steps 1-3 until $N$ samples of $\tilde{\theta}$ is obtained
```

One of the simplest algorithm is called the exact Bayesian computation (EBC) [31] which is described in Algorithm 1. The drawback of the EBC algorithm is that it only works for discrete observed data y_{obs}. This is because for continuous distributions, the acceptance probability at step 3 is zero almost surely. Therefore, for continuous observed data or even discrete data with very low acceptance probability, it is more intuitive to use the ABC-rejection algorithm. In fact, as we will see in Algorithm 2 in the next subsection, EBC is a special case of the ABC rejection algorithm when $\epsilon = 0$.

3.3.2 ABC Rejection

```
Algorithm 2: ABC rejection
  Input: Observed data $y_{\text{obs}}$, prior $\pi(\theta)$, number of samples $N$, model
         $\mathcal{M}(\cdot)$, tolerance level $\epsilon$
  Output: $N$ samples from the ABC posterior
1 Sample $\tilde{\theta}$ from the prior $\pi(\theta)$
2 Simulate $\tilde{y}$ from the model $\mathcal{M}(\tilde{\theta})$
3 Accept $\tilde{\theta}$ if $||\tilde{y} - y_{\text{obs}}|| \leq \epsilon$ and reject otherwise
4 Repeat steps 1-3 until $N$ samples of $\tilde{\theta}$ is obtained
```

For continuous data, it is very unlikely to get an exact match between the simulated data and the observed. Therefore, it is more intuitive to accept parameters that produce simulated data that is ϵ distance away from the observed, where ϵ is specified. Algorithm 2 describes the ABC rejection algorithm. However, it is more practical and computationally efficient to choose to run the simulation N times and choose to keep a proportion $0 < \delta < 1$ of the N simulations with the smallest distance $||y - y_{\text{obs}}||$.

One of the major disadvantage about the ABC rejection algorithm is that it does not exploit information from the already accepted parameters as it continuously draws random samples from the prior in each loop. Therefore, if the prior and the posterior distribution are very different, then the algorithm will be very inefficient because of the low probability of acceptance at step 3. This is often the case when a vague prior is used, e.g. a prior with distribution $\text{Normal}(0, 100^2)$. To counter that drawback, the approximate Bayesian computation Markov chain Monte Carlo (ABC-MCMC) algorithm is proposed. This algorithm was developed by Marjoram [32] by implementing steps in a Metropolis-Hastings algorithm.

3.3.3 Approximate Bayesian Computation Markov Chain Monte Carlo (ABC-MCMC)

Before going into the ABC-MCMC algorithm, it is useful to recall the Metropolis-Hastings algorithm [33] which is a Markov chain Monte Carlo (MCMC) method to obtain samples from a target distribution. The algorithm is shown in Algorithm 3.

In Bayesian inference, the Metropolis-Hastings algorithm is often used to obtain samples from the posterior distribution because very often the posterior has a complex form. Since the posterior is proportional to the likelihood function multiplied by the prior, the target distribution $g(\cdot)$ in Algorithm 3 is often taken to be $g(\theta) = \pi(\theta)f(x|\theta)$, where $\pi(\theta)$ is the prior and $f(x|\theta)$ is the likelihood. However, in the case of ABC, the likelihood is intractable so it is not possible to implement the Metropolis-Hastings algorithm directly. So, could we perhaps

```
Algorithm 3: Metropolis-Hastings algorithm
  Input: Initialisation point $\theta_0$, proposal distribution $q(\cdot|\cdot)$, target
         distribution $g(\cdot)$
  Output: $N$ samples from the target distribution
1 Start with $t = 0$
2 Simulate $\theta^*$ from proposal distribution $q(\cdot|\theta_t)$
3 Set $\theta_{t+1} = \theta^*$ with probability $\alpha$, where
      $\alpha = \frac{g(\theta^*)q(\theta_t|\theta^*)}{g(\theta_t)q(\theta^*|\theta_t)}$
   and $\theta_{t+1} = \theta_t$ otherwise
4 Set $t = t + 1$ and return to step 2 until $N$ samples are obtained
```

somehow approximate the likelihood function such as

$$\hat{f}(x|\theta) = \frac{1}{K}\sum_{i=1}^{K} \mathbb{1}(||y_{\text{obs}} - y_j^*|| \leq \epsilon).$$

If we do so, we can implement the ABC-MCMC algorithm as shown in Algorithm 4.

From the ABC-MCMC algorithm, note that if the model is deterministic, we can set $K = 1$ because of the deterministic nature of the model (i.e we get the same simulated data no matter how many times we run the model). This article [34] showed that repeating simulations of the data at step 3 of the ABC-MCMC algorithm does not contribute significantly to an improved approximation of the ABC posterior even if the model is stochastic. Moreover, this paper [35] proved that $K = 1$ is often very close to the optimal K. Additionally, we can check that for $K = 1$, the chain indeed converges to the ABC posterior by checking the detailed balance equations. This is shown in [32]. Therefore, we can skip step 3 and replace α in step 4 of Algorithm 4 by

$$\alpha = \min\left(1, \frac{\pi(\theta^*)q(\tilde{\theta}|\theta^*)}{\pi(\tilde{\theta})q(\theta^*|\tilde{\theta})}\mathbb{1}\big(||y_{\text{obs}} - y^*|| \leq \epsilon\big)\right).$$

Figure 5 describes the proposal process of a typical ABC-MCMC algorithm. The acceptance rate of the chain is low when it is outside the acceptance region. So, a potential difficulty faced when trying to implement the ABC-MCMC algorithm is that we need a good starting value θ_0 to start the chain because we need $\mathbb{1}(||y_{\text{obs}} - y_0|| \leq \epsilon) = 1$. Selecting a suitable ϵ is also difficult which usually requires long experimental runs to figure out. In addition, the possible low acceptance rate in the proposal process and highly correlated samples of parameter θ may contribute to the chain getting stuck in low probability regions for long periods of time before reaching the acceptance region. In Section 3.4, we will discuss more about the computational techniques to overcome this.

Algorithm 4: Approximate Bayesian Computation Markov Chain Monte Carlo (ABC-MCMC)

```
Input: Observed data $y_{\text{obs}}$, prior $\pi(\theta)$, number of samples $N$, model
       $\mathcal{M}(\cdot)$, tolerance level $\epsilon$, proposal distribution $q(\cdot|\cdot)$,
       initialisation point $\theta_0$
Output: $N$ samples from the ABC posterior
1 Set $\tilde{\theta} = \theta_0$
2 Simulate $\theta^*$ from the proposal distribution $q(\cdot|\tilde{\theta})$
3 Simulate $K$ data from the model $\mathcal{M}(\theta^*)$, $y_1^*, ..., y_K^*$ and calculate
      $r^* = \frac{1}{K} \sum_{j=1}^{K} \mathbb{1}(||y_{\text{obs}} - y_j^*|| \leq \epsilon)$
4 Accept $\theta^*$ with probability $\alpha$ where
      $\alpha = \min\left(1, \frac{r^* \pi(\theta^*) q(\tilde{\theta}|\theta^*)}{\tilde{r} \pi(\tilde{\theta}) q(\theta^*|\tilde{\theta})}\right)$
5 If $\theta^*$ accepted, then set $\tilde{\theta} = \theta^*$ and $\tilde{r} = r^*$
6 Store $\tilde{\theta}$ and other information as needed
7 Repeat steps 2-6 until $N$ samples of the ABC posterior is obtained
```

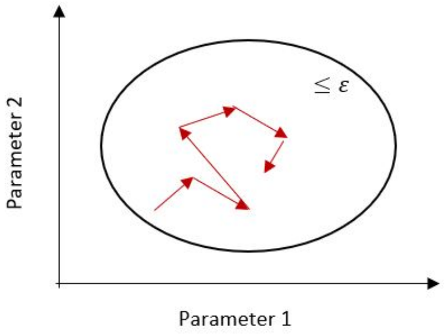


Figure 5: An example of the proposal process of a ABC-MCMC algorithm for a model with two parameters. The circular region indicates the acceptance region where the simulated data corresponds to $||y - y_{\text{obs}}|| \leq \epsilon$. The MCMC chain then moves according to the proposal distribution. Note that the chain can only start when within the acceptance region.

3.3.4 Approximate Bayesian Computation Sequential Monte Carlo (ABC-SMC)

In order to overcome the disadvantages of the rejection and MCMC algorithm, Sisson [36] has developed the ABC-SMC algorithm which was derived from a sequential importance sampling algorithm [37]. The ABC-SMC algorithm was then tweaked in [38] to decrease the computation complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ and to allow the tolerance level to decrease adaptively.

The idea of ABC-SMC is to transform the prior into the posterior through a decreasing sequence of tolerance levels. This can be achieved through: (i) weighing each particle (refers to each draw of the set of parameters) in each population (indicates group of particles corresponding to the same tolerance level), (ii) adaptively choose decreasing sequences of tolerance for each subsequent generation of particles, (iii) resampling the particles if some condition is achieved and (iv) perturb the particles according to a MCMC kernel. Figure 6 describes the process of a typical ABC-SMC algorithm. A simplified version of the ABC-SMC algorithm from [38] is given in Algorithm 5.

3.4 Implementation in R

In this section we will discuss the implementations of the 3 algorithms in R [39] and compare their performance through a case study.

ABC Rejection

The R code for the ABC Rejection algorithm is an implementation of Algorithm 2 with a slight adjustment. Instead of specifying the tolerance level ϵ, a proportion δ is specified such that the output is δN samples from the prior. These δN samples are those that corresponds to simulated data with the smallest distance from the observed. The R code can be seen in figure 7. Note that the function `get_prior` is just a simple function implemented to output a random number based on the prior distribution given in `prior`. To run the function, one needs to specify:

1. `model`: A function to implement the model which would output the set of summary statistics. The summary statistics has to have the same dimension as the observed summary statistics.

2. `prior`: A list of the prior distribution(s) of the parameter(s). Number of prior distributions have to match the number of parameters of the model.

3. `summary_stats`: A vector of observed summary statistics from data available.

4. `simul_number`: An integer which indicates the number of simulations to run.

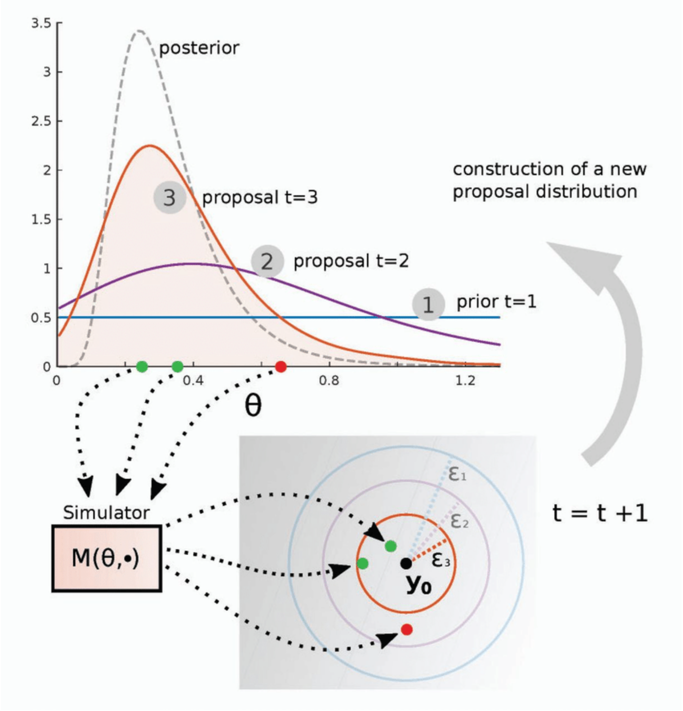
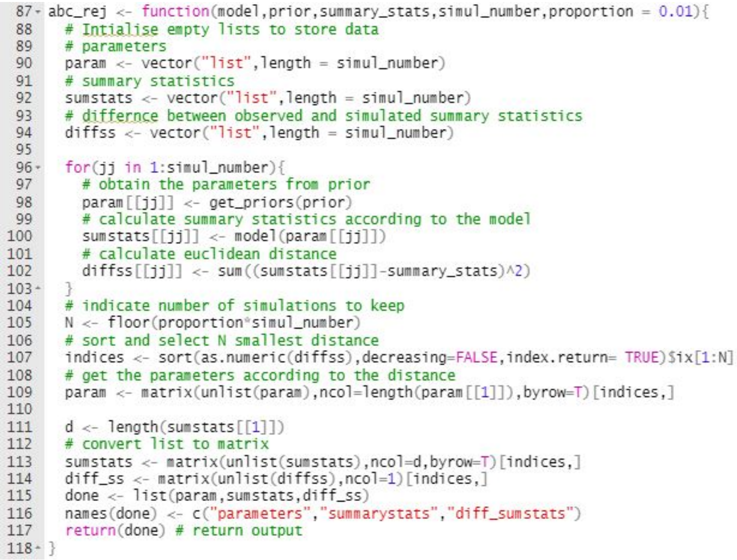


Figure 6: The figure illustrates how ABC-SMC works. The first step is to draw particles from the prior. Then, the particles are put through the simulator to get simulated data. The simulated data are then used to update the tolerance level from ϵ_1 to ϵ_2. The weights of each particles are then computed to update the proposal distribution which is indicated by the purple line. The process is repeated until the distribution resembles the posterior. Figure taken from [29].

Algorithm 5: Approximate Bayesian Computation Sequential Monte Carlo (ABC-SMC)

```
   Input: Observed data $y_{\text{obs}}$, prior $\pi(\theta)$, number of samples $N$, model
          $\mathcal{M}(\cdot)$, perturbation kernal $K_t(\cdot|\cdot)$, target tolerance level $\epsilon_T$,
          proportion of particles to keep $\alpha$
   Output: $N$ samples from the ABC posterior
1  Set $t = 0$ (population indicator)
2  For $i = 1, ..., N$, sample $\theta_i^{(t)}$ from the prior $\pi(\cdot)$ and get $s_i^{(t)}$ from
   $\mathcal{M}(\theta_i^{(t)})$. Set weight $w_i^{(t)} = 1/N$ and $\epsilon_t = \infty$
3  Set $t = t + 1$. If $\epsilon_{t-1} = \epsilon_T$ then stop
4  Compute the next tolerance level $\epsilon_t$ by using a proportion $\alpha$ particles
     from the previous generation which can be represented as
       $$\alpha \sum_{i=1}^{N} \mathbb{1}\big(||y_i^{(t-1)} - y_{\text{obs}}|| \leq \epsilon_{t-1}\big) = \sum_{i=1}^{N} \mathbb{1}\big(||y_i^{(t)} - y_{\text{obs}}|| \leq \epsilon_t\big).$$
     If $\epsilon_t < \epsilon_T$, then set $\epsilon_t = \epsilon_T$
5  For $i = 1, ..., N$, set $w_i^{(t)} = 0$ if $||y_i^{(t-1)} - y_{\text{obs}}|| > \epsilon_t$
6  Renormalise the weight such that $\sum_{i=1}^{N} w_i^{(t)} = 1$
7  Calculate the effective sample size (ESS), where
     $\text{ESS} = \big(\sum_{i=1}^{N}(w_i^{(t)})^2\big)^{-1}$
8  If ESS $< N/2$, then resample $\theta_i^{(t-1)}$ from $\{\theta_i^{(t-1)}\}$ according to weights
     $\{w_i^{(t)}\}$ and set $w_i^{(t)} = 1/N$ for all $i$
9  Perturb all particles with $w_i^{(t)} > 0$ such that $\theta_i^* \sim q_t(\cdot|\theta_i^{(t-1)})$ where
       $$q(\cdot|\theta_i^{(t-1)}) = \frac{\sum_{i=1}^{N} w_i^{(t-1)} K_t(\cdot|\theta_i^{(t-1)})}{\sum_{i=1}^{N} w_i^{(t-1)}}$$
10 Set $\theta_i^{(t)} = \theta_i^*$ with probability
       $$\min\left(1, \frac{\pi(\theta_i^*)q(\theta_i^{(t-1)}|\theta_i^*)}{\pi(\theta_i^{(t-1)})q(\theta_i^*)|\theta_i^{(t-1)}\theta_i^*)} \mathbb{1}\big(||y_i^* - y_{\text{obs}}|| \leq \epsilon\big)\right)$$
     and set $\theta_i^{(t)} = \theta_i^{(t-1)}$ otherwise
11 Go to step 3
```



```r
abc_rej <- function(model,prior,summary_stats,simul_number,proportion = 0.01){
  # Intialise empty lists to store data
  # parameters
  param <- vector("list",length = simul_number)
  # summary statistics
  sumstats <- vector("list",length = simul_number)
  # differnce between observed and simulated summary statistics
  diffss <- vector("list",length = simul_number)

  for(jj in 1:simul_number){
    # obtain the parameters from prior
    param[[jj]] <- get_priors(prior)
    # calculate summary statistics according to the model
    sumstats[[jj]] <- model(param[[jj]])
    # calculate euclidean distance
    diffss[[jj]] <- sum((sumstats[[jj]]-summary_stats)^2)
  }
  # indicate number of simulations to keep
  N <- floor(proportion*simul_number)
  # sort and select N smallest distance
  indices <- sort(as.numeric(diffss),decreasing=FALSE,index.return= TRUE)$ix[1:N]
  # get the parameters according to the distance
  param <- matrix(unlist(param),ncol=length(param[[1]]),byrow=T)[indices,]

  d <- length(sumstats[[1]])
  # convert list to matrix
  sumstats <- matrix(unlist(sumstats),ncol=d,byrow=T)[indices,]
  diff_ss <- matrix(unlist(diffss),ncol=1)[indices,]
  done <- list(param,sumstats,diff_ss)
  names(done) <- c("parameters","summarystats","diff_sumstats")
  return(done) # return output
}
```

Figure 7: ABC rejection algorithm implemented in R

5. `proportion`: A positive number between 0 and 1 which indicates the proportion of simulations with the smallest distance between the observed and simulated summary statistics to be retained. Default is set to 0.01.

ABC-MCMC

As we have mentioned, there are a few issues with implementing the ABC-MCMC algorithm in Section 3.3.3 directly. The problems, along with our proposed solutions (adopted from [40]) are as follows:

- A tolerance level ϵ has to be specified as one of the inputs of the algorithm. Therefore, we propose an initial tuning step by running the ABC rejection algorithm with N simulations and keep a proportion δ of the "closest" simulations to the observed data. Then, ϵ is chosen to be the largest distance among the "closest" simulations.

- A starting point, θ_0 such that $||y_0 - y_{\text{obs}}|| \leq \epsilon$ has to be provided in order to have an efficient algorithm because acceptance rate will be low when θ_0 generates y_0 that is far away from y_{obs}. So, we propose to randomly set θ_0 as one of the $N\delta$ parameters from the previous bullet point.

- It is not clear as of how to choose the range of the proposal distribution $q(\cdot|\cdot)$. If the proposal range is too large, then the chain might jump to parameters that generate simulations outside the range of ϵ. If the proposal range is too small, the chain might not explore the whole parameter space. Therefore, we propose to select the proposal range by using the same idea as the first bullet point. Among the $N\delta$ simulations from the ABC rejection method, we take the range, r, as the largest parameter value subtracted by the smallest parameter value as the range. For instance, we will set the proposal distribution $q(\cdot|\theta_0)$as Uniform$(\theta_0 - r/2, \theta_0 + r/2)$ if the prior is uniform (of course we have to take into account of the range of the prior which we have to stay within) and as Normal$(\theta_0, (r/2)^2)$ if the prior is normally distributed.

An example code snippet of the ABC-MCMC algorithm is presented in Figure 8. The inputs for the ABC-MCMC algorithm are `model`, `prior`, `summary_stats` and `proportion` which are the same as the ABC Rejection algorithm with some additional inputs:

1. `n_sample`: An integer that determines the number of samples to be simulated from the ABC posterior.

2. `n_trials`: An integer that indicates the number of simulations to run for the ABC rejection algorithm as the initial calibration step. Default is set to 10000.

ABC-SMC

We will not be constructing an algorithm for ABC-SMC from scratch in R. Instead, we will be using the function `ABC_sequential` implemented in the EasyABC package [41]. The method used will be “Delmoral” [38], where the simplified version of the algorithm was given previously in Section 3.3.4.

Local-linear regression

It is straightforward to apply the local-linear regression postsampling technique in Section 3.2. We can do so by using the `lm` function in R. A code snippet is provided in Figure 9. The inputs are:

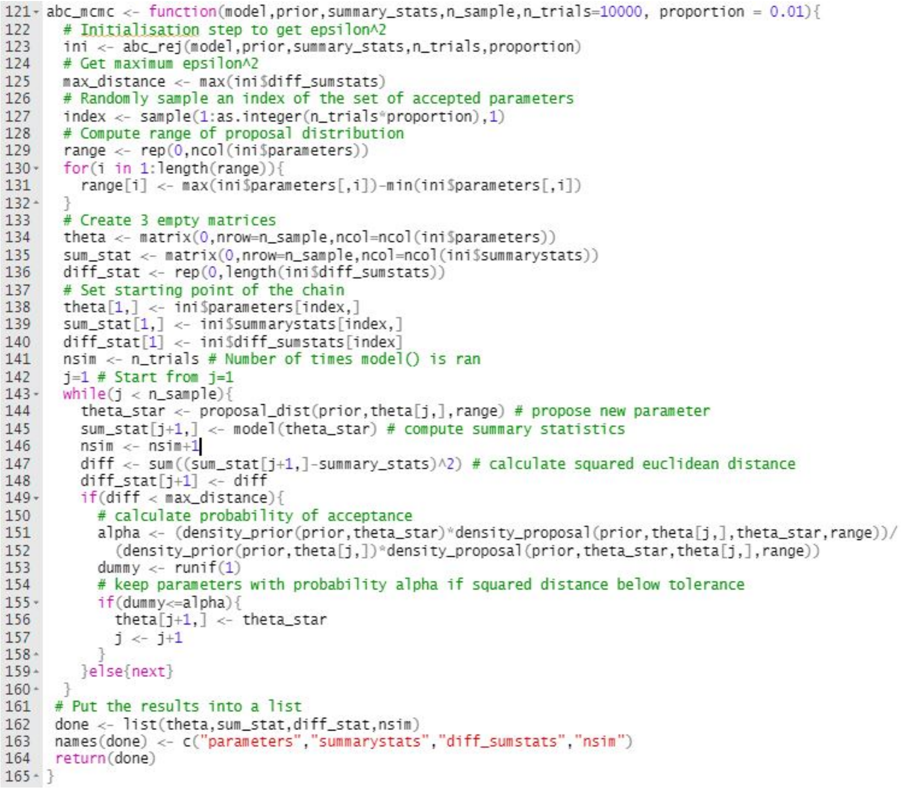
1. `parameters`: A matrix that represents parameter samples from the ABC posterior. Each column corresponds to a parameter.

2. `simulated`: A matrix that represents simulated outputs of the model from the parameters in 1. Each row corresponds to the set of simulated statistics.

3. `summary_stats`: A vector that represents the observed statistics.

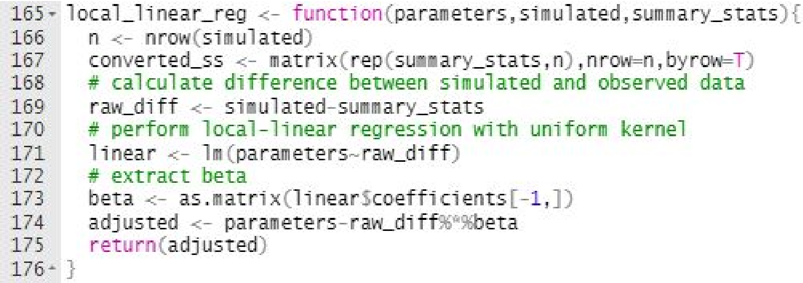
3.4.1 Case Study

In this section, we will be doing a simple case study to demonstrate the use of ABC algorithms. First, we will simulate data from a deterministic SIR model as described in Section 2 with known parameter values $\beta = 2.0$, $\gamma = 0.3$ and



```r
abc_mcmc <- function(model,prior,summary_stats,n_sample,n_trials=10000, proportion = 0.01){
  # Initialisation step to get epsilon^2
  ini <- abc_rej(model,prior,summary_stats,n_trials,proportion)
  # Get maximum epsilon^2
  max_distance <- max(ini$diff_sumstats)
  # Randomly sample an index of the set of accepted parameters
  index <- sample(1:as.integer(n_trials*proportion),1)
  # Compute range of proposal distribution
  range <- rep(0,ncol(ini$parameters))
  for(i in 1:length(range)){
    range[i] <- max(ini$parameters[,i])-min(ini$parameters[,i])
  }
  # Create 3 empty matrices
  theta <- matrix(0,nrow=n_sample,ncol=ncol(ini$parameters))
  sum_stat <- matrix(0,nrow=n_sample,ncol=ncol(ini$summarystats))
  diff_stat <- rep(0,length(ini$diff_sumstats))
  # Set starting point of the chain
  theta[1,] <- ini$parameters[index,]
  sum_stat[1,] <- ini$summarystats[index,]
  diff_stat[1] <- ini$diff_sumstats[index]
  nsim <- n_trials # Number of times model() is ran
  j=1 # Start from j=1
  while(j < n_sample){
    theta_star <- proposal_dist(prior,theta[j,],range) # propose new parameter
    sum_stat[j+1,] <- model(theta_star) # compute summary statistics
    nsim <- nsim+1
    diff <- sum((sum_stat[j+1,]-summary_stats)^2) # calculate squared euclidean distance
    diff_stat[j+1] <- diff
    if(diff < max_distance){
      # calculate probability of acceptance
      alpha <- (density_prior(prior,theta_star)*density_proposal(prior,theta[j,],theta_star,range))/
        (density_prior(prior,theta[j,])*density_proposal(prior,theta_star,theta[j,],range))
      dummy <- runif(1)
      # keep parameters with probability alpha if squared distance below tolerance
      if(dummy<=alpha){
        theta[j+1,] <- theta_star
        j <- j+1
      }
    }else{next}
  }
 # Put the results into a list
 done <- list(theta,sum_stat,diff_stat,nsim)
 names(done) <- c("parameters","summarystats","diff_sumstats","nsim")
 return(done)
}
```

Figure 8: ABC-MCMC algorithm implemented in R



```r
local_linear_reg <- function(parameters,simulated,summary_stats){
  n <- nrow(simulated)
  converted_ss <- matrix(rep(summary_stats,n),nrow=n,byrow=T)
  # calculate difference between simulated and observed data
  raw_diff <- simulated-summary_stats
  # perform local-linear regression with uniform kernel
  linear <- lm(parameters~raw_diff)
  # extract beta
  beta <- as.matrix(linear$coefficients[-1,])
  adjusted <- parameters-raw_diff%*%beta
  return(adjusted)
}
```

Figure 9: R code to implement local-linear regression adjustment

initial values ($S_0 = 99, I_0 = 1, R_0 = 0$) for time $t = 0, 1, ..., 10$. We will see in detail how to construct a SIR model in R later on in Section 4. Figure 10 describes the change in the number of people in each compartment over time. Suppose that we want to predict the parameters β and γ but only have access to the number of infected across the time scale. We can do so by applying ABC techniques. The priors for all three algorithms are set to be Uniform$(0.5, 2.5)$ for β and Uniform$(0, 0.5)$ for γ. Each algorithms are ran such that they produce 1000 samples of the posterior. The ABC posterior of each method is shown in Figure 11.

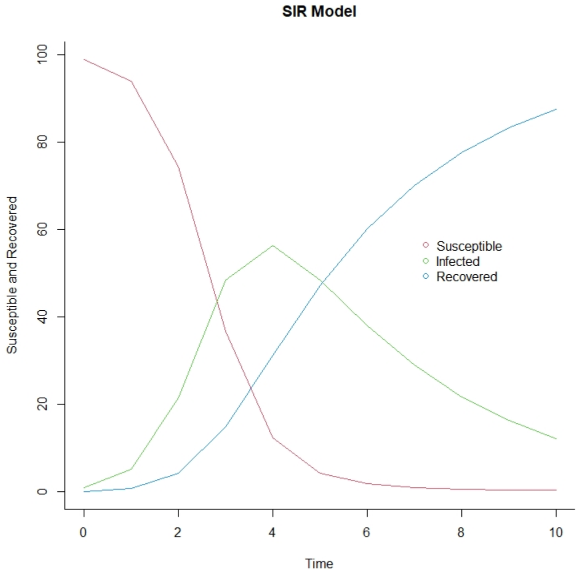


Figure 10: A simulated deterministic SIR model with $\beta = 2.0$ and $\gamma = 0.3$

Note that in Figure 11, local-linear regression is not applied to ABC-SMC because the algorithm is designed to decrease the tolerance level adaptively to an acceptable level. Here, the ABC-SMC algorithm decreased the tolerance level to $\epsilon = 0.0.001$ as we set the tolerance target as $\epsilon_T = 0.001$. We can observe that the local-linear regression technique greatly improves the convergence of the posterior. We see that all algorithms seems to be able to approximate the posterior mean fairly well. Table 1 summarises our findings. The ABC-MCMC method took the least amount of time whereas the ABC rejection algorithm have the longest computation time. The ABC-MCMC algorithm ran the SIR model the least amount of time while the ABC Rejection algorithm ran the model the most amount of times. We can also observe that the ABC-SMC algorithm results in a ABC posterior with a very narrow range and hence has the best accuracy. On the other hand, the ABC-MCMC algorithm uses an initial calibration step which results in a larger ϵ than ABC-SMC and therefore the

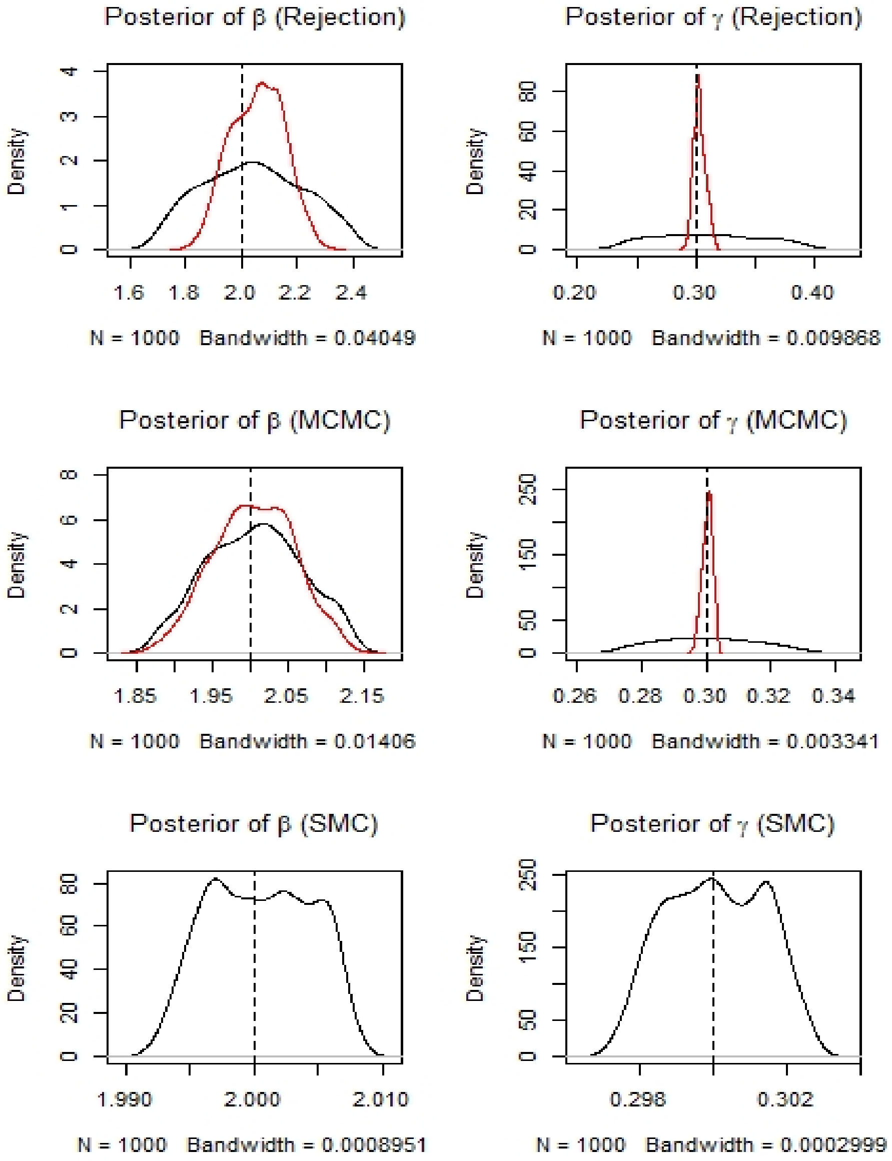


Figure 11: ABC posterior of each algorithm. The dotted vertical line describes the true value of the parameters. The black line represents the ABC posterior while the red line represents the ABC posterior after performing local-linear regression.

wider range of accepted parameters. The ABC-MCMC algorithm implemented here is only possible because the range of the prior is quite small. For example, if a vague prior is used such as Normal$(0, 100^2)$, it is very probable that the initial calibration step will set ϵ to be very large and so the posterior distribution will have a large range.

	Posterior mean	Computation time (s)	Times model ran
ABC Rej	2.041 (β) 0.311 (γ)	14.5	10000
ABC-MCMC	2.005 (β) 0.300 (γ)	4.3	2650
ABC-SMC	2.000 (β) 0.300 (γ)	11.7	5000

Table 1: Summaries of the posterior mean, computation time in seconds, and number of times the the SIR model was ran by each algorithm. The computation time is recorded when 1000 samples from the ABC posterior is obtained. Note that posterior means for ABC Rej and ABC-MCMC are calculated after performing regression adjustment.

4 Comparison of deterministic and stochastic SIR models

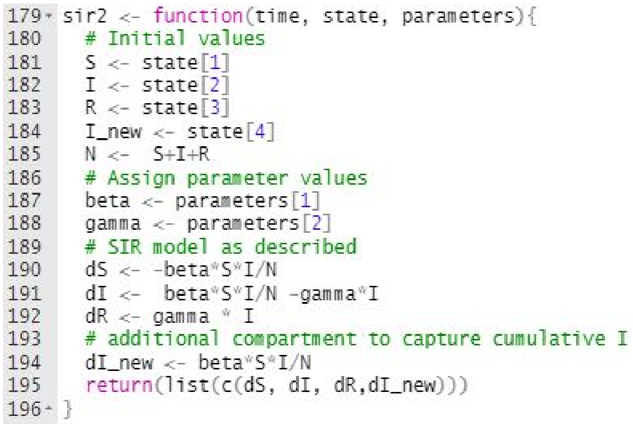
4.1 Modelling Covid-19 data

The goal of this section is to investigate the relationship between a deterministic and a stochastic SIR model by implementing them on the UK Covid-19 dataset. We agree that the SIR model may not be the most appropriate to model the pandemic but the main purpose of this study is to compare the differences between a stochastic and deterministic epidemic model. Firstly, we implement both SIR compartmental models on Covid-19 data in the United Kingdom and make some comparisons between these two models such as assessing their simulation variance, and prediction performance by cross-validating with unseen data. In particular, Covid-19 data from England and Bristol will be used. The data is available at [42]. Note that the dataset only contains cumulative number of infected individuals daily with no data on the number of recovered individuals. Due to the nature of the data, we introduce another compartment to the model, I_{new}, where

$$\frac{dI_{\text{new}}}{dt} = \frac{\beta S(t)I(t)}{N}$$

which captures the cumulative number of people infected by the disease.

Before going into the implementation of SIR models in R, it is important to note that epidemic data (e.g. cumulative Covid-19 cases reported) are typically



```
sir2 <- function(time, state, parameters){
  # Initial values
  S <- state[1]
  I <- state[2]
  R <- state[3]
  I_new <- state[4]
  N <-  S+I+R
  # Assign parameter values
  beta <- parameters[1]
  gamma <- parameters[2]
  # SIR model as described
  dS <- -beta*S*I/N
  dI <-  beta*S*I/N -gamma*I
  dR <- gamma * I
  # additional compartment to capture cumulative I
  dI_new <- beta*S*I/N
  return(list(c(dS, dI, dR,dI_new)))
}
```

Figure 12: An example of how a deterministic SIR function can look like.

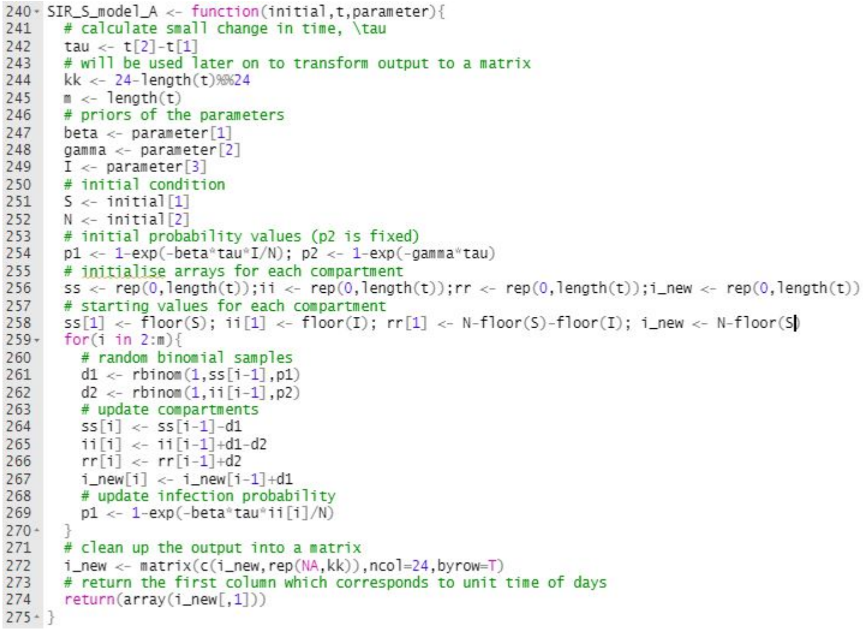
under-reported or complicated by data incompleteness. To account for this observation error, a data model is often used. For example, a Poisson distribution or negative binomial distribution could be a candidate as the data model, e.g.

$$Y_t \sim \text{Poisson}(\phi X_t)$$

where X_t are outputs from the process model (SIR model in this case), Y_t are the observations from the data and ϕ a parameter. In general, to model an epidemic through Bayesian inference, one would need three components: (i) the process model, (ii) the data model, and (iii) the prior (parameter model). (Refer [43] for an example). These type of models are termed Partially observed Markov Process (POMP) models [44]. However, our interest is to compare the deterministic against the stochastic SIR model. Therefore, the POMP modelling framework will not be used. Instead, only the process model is used to fit the data. Next, we will implement both deterministic and stochastic SIR models in R.

Deterministic SIR

The deterministic SIR model, which is governed by a set of ordinary differential equations (ODE) can be easily implemented using the function `ode` in the R package `deSolve` [45]. First, one needs to create a function that describes the SIR model where one feeds in arguments such as unit time to run the ODEs, initial values, and respective parameters. A code snippet of such function is displayed in Figure 12. With the function constructed, it is straightforward to run the SIR model using the `ode` function. The outputs in the I_{new} compartment is then our simulated statistics that we would want to compare with the observed cumulative infected people.



```r
SIR_S_model_A <- function(initial,t,parameter){
  # calculate small change in time, \tau
  tau <- t[2]-t[1]
  # will be used later on to transform output to a matrix
  kk <- 24-length(t)%%24
  m <- length(t)
  # priors of the parameters
  beta <- parameter[1]
  gamma <- parameter[2]
  I <- parameter[3]
  # initial condition
  S <- initial[1]
  N <- initial[2]
  # initial probability values (p2 is fixed)
  p1 <- 1-exp(-beta*tau*I/N); p2 <- 1-exp(-gamma*tau)
  # initialise arrays for each compartment
  ss <- rep(0,length(t));ii <- rep(0,length(t));rr <- rep(0,length(t));i_new <- rep(0,length(t))
  # starting values for each compartment
  ss[1] <- floor(S); ii[1] <- floor(I); rr[1] <- N-floor(S)-floor(I); i_new <- N-floor(S)
  for(i in 2:m){
    # random binomial samples
    d1 <- rbinom(1,ss[i-1],p1)
    d2 <- rbinom(1,ii[i-1],p2)
    # update compartments
    ss[i] <- ss[i-1]-d1
    ii[i] <- ii[i-1]+d1-d2
    rr[i] <- rr[i-1]+d2
    i_new[i] <- i_new[i-1]+d1
    # update infection probability
    p1 <- 1-exp(-beta*tau*ii[i]/N)
  }
  # clean up the output into a matrix
  i_new <- matrix(c(i_new,rep(NA,kk)),ncol=24,byrow=T)
  # return the first column which corresponds to unit time of days
  return(array(i_new[,1]))
}
```

Figure 13: An example of the implementation of a SIR stochastic SIR model in R that outputs the cumulative number of infected throughout the epidemic.

Stochastic SIR

For the stochastic SIR model, we can directly implement a function that runs equation 3 in Section 2.2. Additionally, we will use the the approximation procedure of τ-leaps method with $\tau = 1/24$ to decrease the computation time without compromising the accuracy of the simulated data significantly. The code snippet of such function is described in Figure 13.

About the Covid-19 dataset, due to the implementation of various non-pharmaceutical interventions (NPIs) by the UK government and the increasing awareness of the public to practice social distancing, it is not reasonable to assume that the parameters β and γ that controls the models to be constant throughout the pandemic. However, we can argue that the parameters remain somewhat constant during a set period of days of the pandemic. To that end, we will use data from a 30-day period to fit our data. Specifically, data from 17th April 2020 to 16th May 2020 will be used for England whereas data from the 1st May 2020 to 30th May 2020 will be used for Bristol. A (weak) justification for the use of these dates is that the transmission of disease would have stabilised (and so the model parameters are relatively constant) some time after the first lockdown in UK [46] on 23rd March 2020.

Data	Population	Parameters	Priors
England	5.6×10^7	β	Lognormal(log(0.12),0.3)
		γ	Lognormal(log(0.15),0.3)
		I_0	Normal($4 \times 10^4, 5 \times 10^3$)
Bristol	4.634×10^5	β	Uniform(0.0,0.1)
		γ	Uniform(0.0,0.1)
		I_0	Uniform(300,812)

Table 2: This table summarises the input information for running the ABC algorithm. Both deterministic and stochastic models use the same priors for the parameters.

Data	Parameter	Posterior	Posterior	2.5%	97.5%
		Mean	Median	Quantile	Quantile
England (D)	β	0.1097	0.1084	0.0843	0.1423
	γ	0.1267	0.1274	0.0775	0.1749
	I_0	38527	38527	28814	48636
England (S)	β	0.1095	0.1081	0.0847	0.1432
	γ	0.1269	0.1273	0.0795	0.1743
	I_0	38650	38715	29011	48419
Bristol (D)	β	0.0313	0.0298	0.0174	0.0510
	γ	0.0510	0.0505	0.00397	0.0966
	I_0	522	499	314	788
Bristol (S)	β	0.0321	0.0303	0.0173	0.0550
	γ	0.0528	0.0533	0.00403	0.0973
	I_0	520	505	312	793

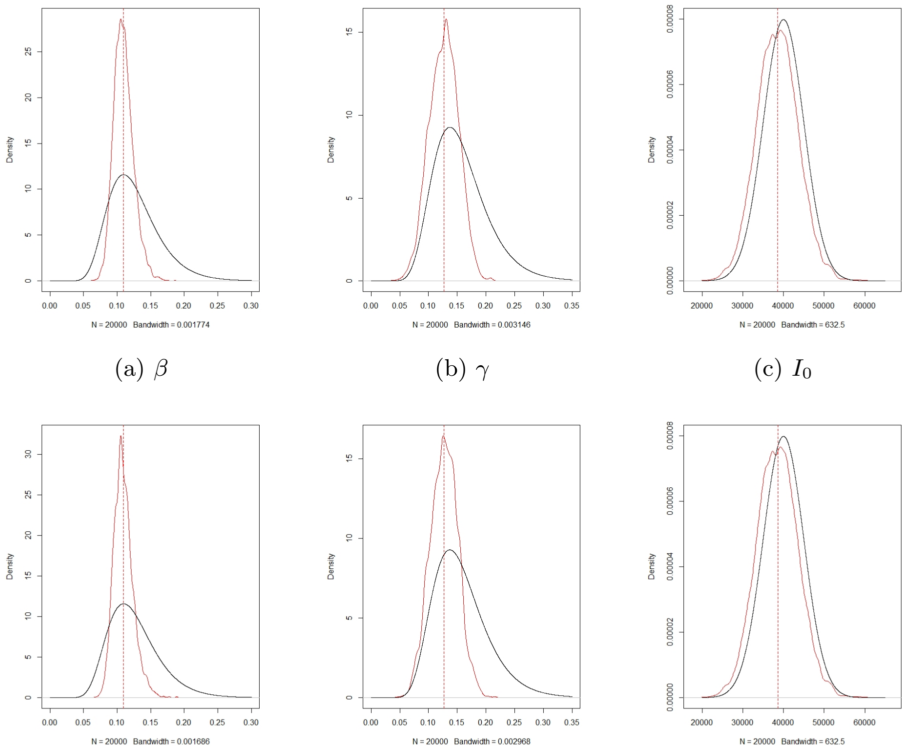
Table 3: Table summarises the parameter estimations for the models. D denotes deterministic model whereas S denotes stochastic models.

4.1.1 Parameter estimation

Parameter estimation of the models are done using the ABC-SMC algorithm by using the `ABC_sequential` function in the R package `EasyABC`. The information about the priors of the parameters and population of each area is described in Table 2. Population of England and Bristol is obtained from [47] and [48] respectively. Figure 14 and 15 shows the posterior distribution as compared to the prior and the posterior mean. We can observe that the posterior means of the same parameters for the deterministic model and stochastic models in both scenarios are similar. The posterior means, posterior median, 2.5% and 97.5% quantiles of each parameters are summarised in Table 3.

4.1.2 Simulation variance

In this section we will discuss the simulation variance of the model when we use the ABC posterior mean as our Bayes estimator. Note that for determin-



(a) β

(b) γ

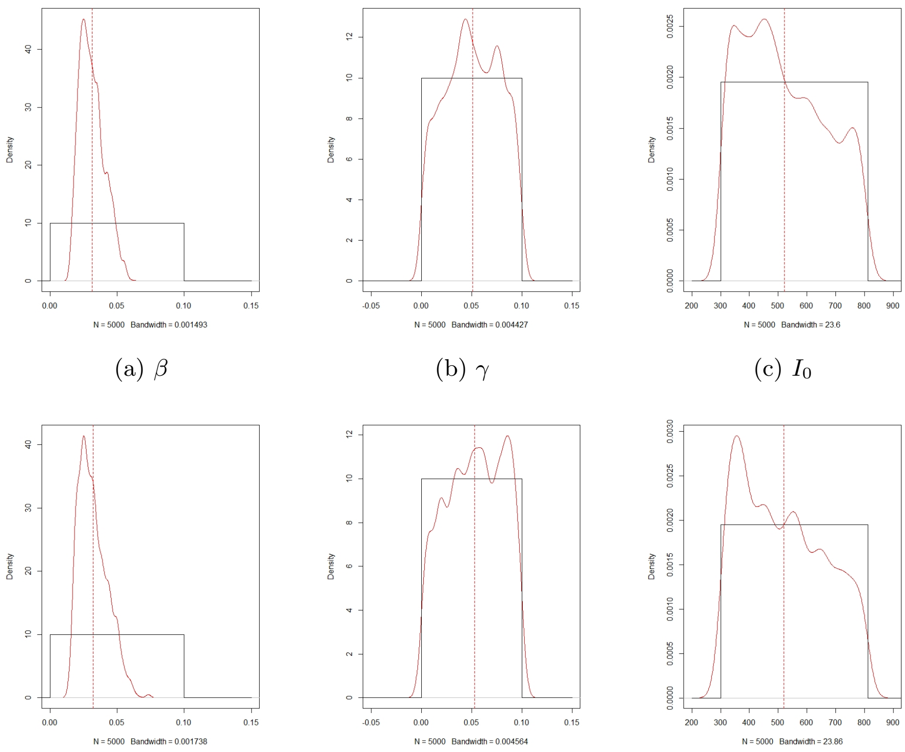
(c) I_0

(d) β

(e) γ

(f) I_0

Figure 14: The set of figures describe the posterior distribution compared to the prior distribution for Covid-19 data in England. The first row denotes the deterministic model and the second row denotes the stochastic model. The black line represents the prior distribution whereas the red line represents the ABC posterior. The red dotted line corresponds to the posterior mean.



(a) β

(b) γ

(c) I_0

(d) β

(e) γ

(f) I_0

Figure 15: The set of figures describe the posterior distribution compared to the prior distribution for Covid-19 data in Bristol. The first row denotes the deterministic model and the second row denotes the stochastic model. The black line represents the prior distribution whereas the red line represents the ABC posterior. The red dotted line corresponds to the posterior mean.

istic models, each run of the model with the same parameter inputs will result in exactly the same outputs every time. Figure 16 illustrates the fit of the deterministic model and the stochastic model respectively.

From Figure 16, the deterministic models are able to fit the general trend of the data quite effectively. We can observe that for the England data, the model was not able to capture the spike in the number of cumulative cases at around day 15. Similarly, for the Bristol data, the model was only able to fit a best fit curve. On the other hand, the stochastic model introduces uncertainty into the cumulative cases due to the random nature of the model as we can see from the multiple grey lines that were plotted. We can observe that for the England data, the stochastic model also could not capture the sudden spike at around day 15. For the Bristol data, the trajectory of the cumulative cases curve is entirely within the range of estimation of the stochastic model.

Now, if we compare the two stochastic models (England (S) and Bristol (S) in Figure 16), we can observe that the relative variance of the Bristol model is much larger than the England model. To quantify this, we can compare the coefficient of variation [49] of the cumulative cases on each day. The coefficient of variation is defined as

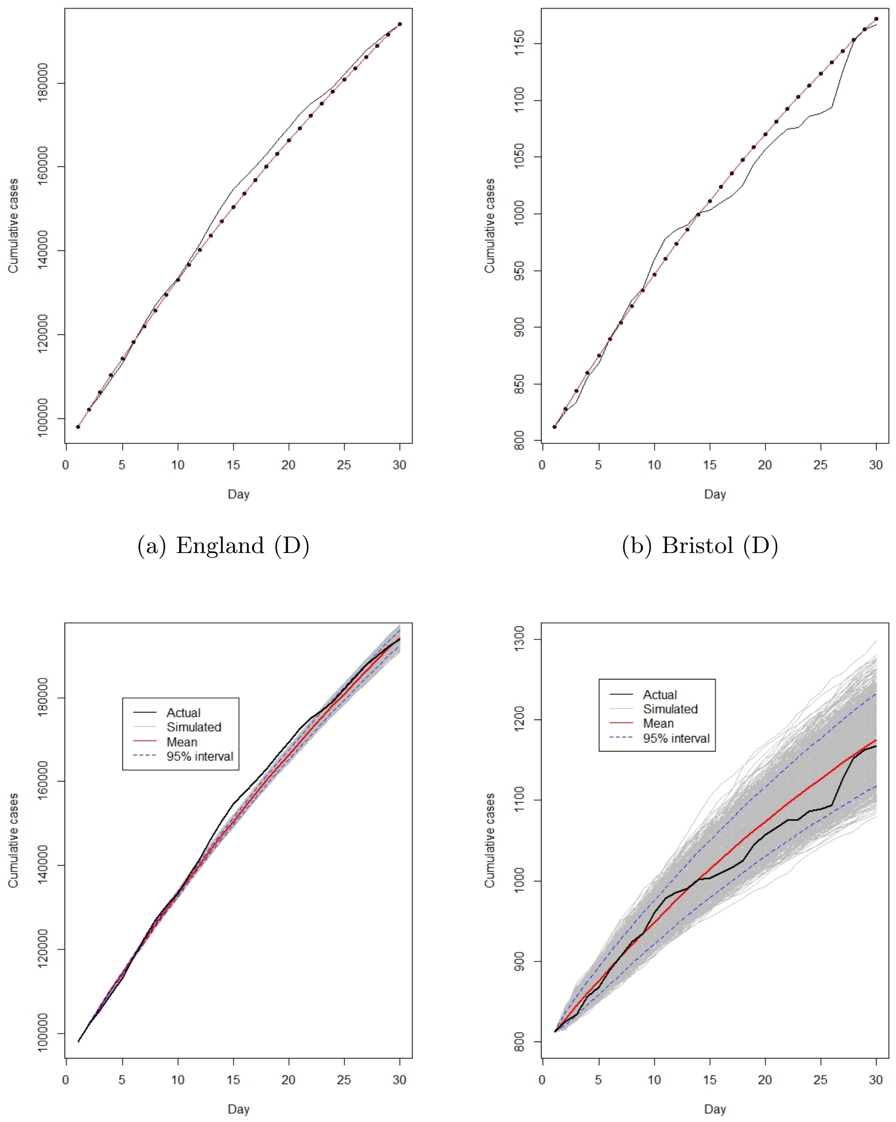
$$\mathrm{cv}_i = \frac{\sigma_i}{\mu_i},$$

where σ_i represents the standard deviation of cumulative cases of day i and μ_i represents the mean cumulative cases of day i. Figure 17 displays the results. We can observe that coefficient of variation of the Bristol data is larger than the England data for all days so we can conclude that the relative variance in Bristol data is larger.

With that, we can conclude that the larger population of England weakens the effect of randomness and hence the relatively smaller variance whereas the smaller population in Bristol amplifies the effect of randomness and so has a relatively larger variance. In the next section, we will cross-validate our models by testing their performance on predicting unseen data.

4.1.3 Prediction performance

To investigate the predictive performance of the model, we can cross-validate our model by predicting the cumulative cases 10 days into the future. Note that model parameters may change over time, and so the assumption that model parameters maintain relatively constant may not hold but we will ignore this. Figure 18 displays how the models perform. We can discover that the deterministic models overestimate the number of cumulative cases across all days. The stochastic model on the other hand, was capable of capturing the trajectory of the curve by introducing uncertainties. To compare their predictive performance quantitatively, we can calculate the root mean squared error for both models. Table 4 summarises the information. We can see that the stochastic models have larger RMSE as compared to the deterministic model. However, the deterministic models have no way to quantify their uncertainties about the estimations. On the other hand, the stochastic model is capable of quantifying



(a) England (D)

(b) Bristol (D)

(c) England (S)

(d) Bristol (S)

Figure 16: Plot of cumulative cases of actual data compared to the models. First row shows the deterministic model whereas the second row the stochastic model. Black line represents the actual data of the number of cumulative cases. Red and grey lines shows the simulated data from the posterior mean of the parameters. Red line represents the mean of the simulations whereas the dotted blue lines are the 95% interval. 5×10^3 samples were simulated for the stochastic models.

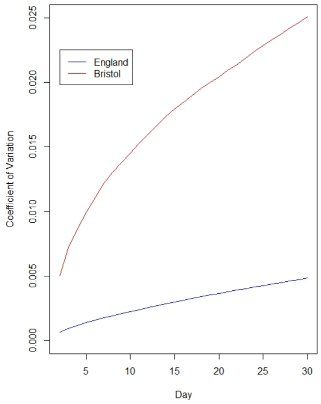
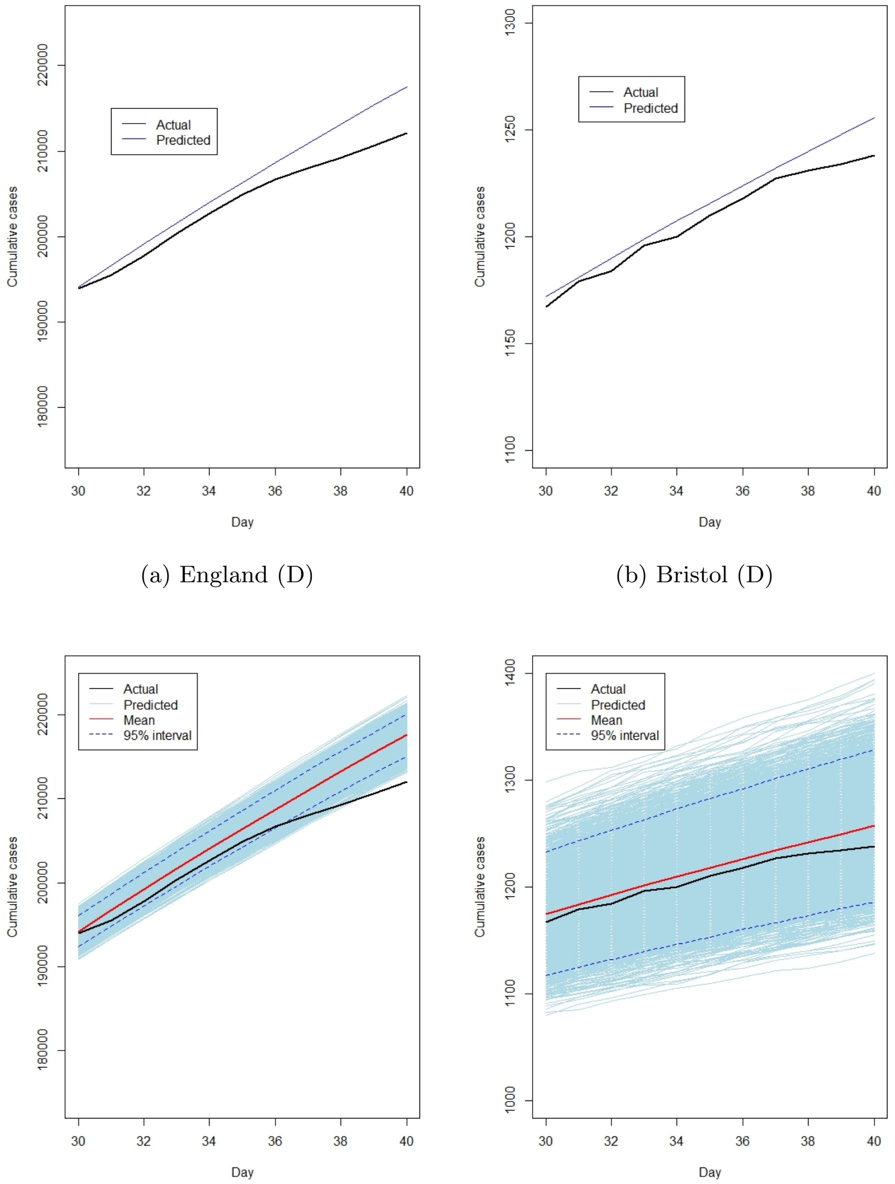


Figure 17: Coefficient of variation plot of each day. Day 1 is excluded because both stochastic model start with the same initial condition, i.e. the initial cumulative cases of day 1 is fixed.

Data	Model	Root mean square error (RMSE)
England	Deterministic	9364.50
	Stochastic	10255.20
Bristol	Deterministic	27.65
	Stochastic	110.52

Table 4: Table summarises the root mean square error for the models. The deterministic model has a fixed root mean square error for a given set of parameters. For the stochastic model, a random sample of 5×10^3 samples are used to calculate the RMSE.



(a) England (D)

(b) Bristol (D)

(c) England (S)

(d) Bristol (S)

Figure 18: First row corresponds to the deterministic models whereas the second row corresponds to the stochastic models. Black line represents the data 10 days into the future (Day 31 to 40). Blue line represents the prediction of the model whereby the posterior means of the respective parameters are used.

such uncertainties. As we can see from Figure 18, the stochastic models do a much better job at capturing the trajectory of the curves than the deterministic models as they consistently overestimate the curve. Therefore, despite having a higher RMSE, the stochastic models are preferable over deterministic models.

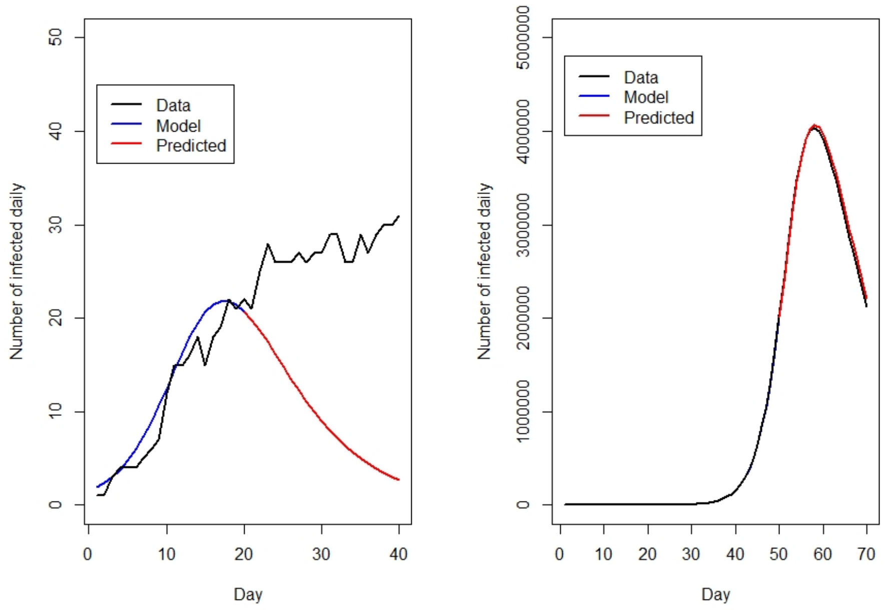
To conclude our study on the Covid-19 data in England and Bristol, population size seems to be a significant factor in how randomness affects the cumulative number of disease infection cases. When the population size is small, randomness can cause significant fluctuations in the number of cases but the effect is largely diminished when population size is large. We have also concluded that the stochastic model is almost always preferable over the deterministic model due to its capability of quantifying uncertainties. On the other hand, deterministic model may be biased as it only produces the same curve when predicting unseen data. Additionally, the transmission of infectious disease is stochastic in nature and so it is natural to simulate the epidemic process with a stochastic model. To continue the study on the comparison of deterministic and stochastic models further, the next section will be focusing on simulating deterministic and stochastic model under different scenarios.

4.2 Comparison using simulations

In the previous section, we have shown how population size will affect randomness in the disease transmission process using the Covid-19 dataset. We have also argued that using stochastic models to model an epidemic is almost always better because it allows us to quantify our uncertainties about our predictions. As we have discussed in Section 2.2, epidemic processes are stochastic in nature. Therefore, we would like to investigate how good the deterministic model is when we assume that the epidemic is a stochastic process. Besides that, the behaviour of the stochastic model is often not reflected in the deterministic model. So, we are also interested in seeing how varying reproduction number R_0, initial number of infected I_0, and recovery time (in days) affects the trajectory of a stochastic model. We will summarise the outcome of the epidemics in a few categories, namely the number of infected at the peak (Peak infected), the total infected throughout the epidemic (Final size), duration of the epidemic (Epidemic duration) and the number of days until the peak is reached (Peak day).

4.2.1 Stochastic model assumption

In this section, we will assume the epidemic itself is a random process. Hence, we will simulate an epidemic with the stochastic SIR model and observe how the deterministic model fits the data. Additionally, we will observe this on two different population sizes, $N = 10^2$ and $N = 10^7$. Both epidemics are simulated with the same parameters such that $\beta = 0.4$, $\gamma = 0.1$, $I_0 = 1$. For the epidemic with the smaller population size, the summary statistics are taken to be the number of infected daily from day 1 to day 20 whereas for the epidemic with the larger population size, the summary statistics is taken to be the number of



(a) $N = 10^2$

(b) $N = 10^7$

Figure 19: Comparison between the fit of a deterministic model (model parameters inferred from the posterior mean of the ABC posterior) on a stochastic epidemic model simulation with small population vs large population.

infected daily from day 35 to 55. The deterministic models are fitted using the ABC-SMC algorithm with priors $\beta \sim \text{Uniform}(0, 1)$, $\gamma \sim \text{Uniform}(0, 0.5)$, and $I_0 \sim \text{lognormal}(\log(1), 3)$. The results of the deterministic model fitted with the ABC posterior mean are plotted in Figure 19.

As we can see from Figure 19, the deterministic model is able to fit the data much better when the population size is large as compared to a smaller population size. When the population size is small, the deterministic model is not capable of predicting the peak number of infected, it significantly underestimates the peak and the time the peak occurs. As the peak is essential information for the government and responsible bodies to make important decisions, the deterministic model is not suitable in providing sufficient information when the population size is small. Therefore, one should be cautious when trying to fit a deterministic epidemic model when the population size is small.

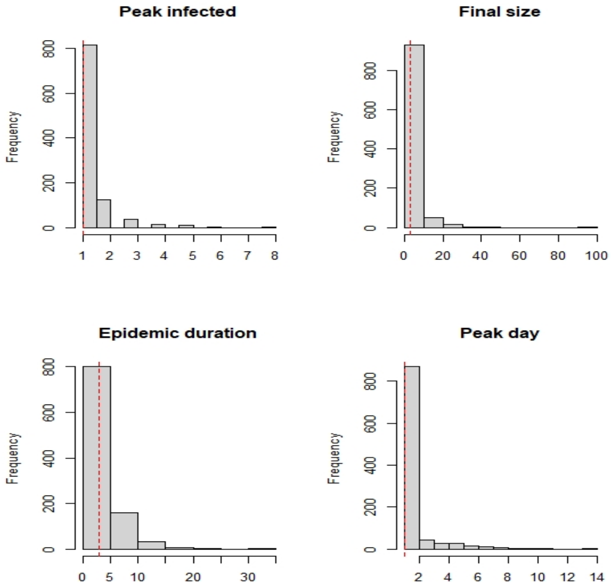


Figure 20: Histogram representing distributions of interest when $R_0 < 1$.

4.2.2 Reproduction number

For a deterministic epidemic model, it is common knowledge in epidemiology that when $R_0 > 1$ an epidemic will happen, and when $R_0 < 1$, an epidemic will not happen so the disease will eventually die out. However, it may not be the case for stochastic models. So, we would like to investigate how R_0 affects the dynamics of a stochastic model. We will compare differing R_0 values with the same recovery rate $\gamma = 0.5$, initial infected $I_0 = 1$, population size $N = 10^3$. We investigate the peak number of infected, final epidemic size, epidemic duration, and the day at which the peak happened for three different R_0 values, $R_0 = 0.4$, $R_0 = 1$ and $R_0 = 3$.

From the three figures, it is apparent that all stochastic models, no matter the value of R_0, has finite probability of an epidemic happening or not happening. From Figure 20, when $R_0 < 1$, the key features of the epidemic for both models are quite similar but the stochastic model has a small probability for a minor outbreak to happen. When $R_0 = 1$, the disease persists within the population for a longer period of time for the deterministic model whereas the epidemic duration for the stochastic model is significantly shorter. When $R_0 > 1$, the key features of the stochastic model are similar to the deterministic model if we ignore the times where the stochastic model results in an epidemic not happening. In the next section, we will investigate how the number of initially infected individuals affect the dynamics of the outcome.

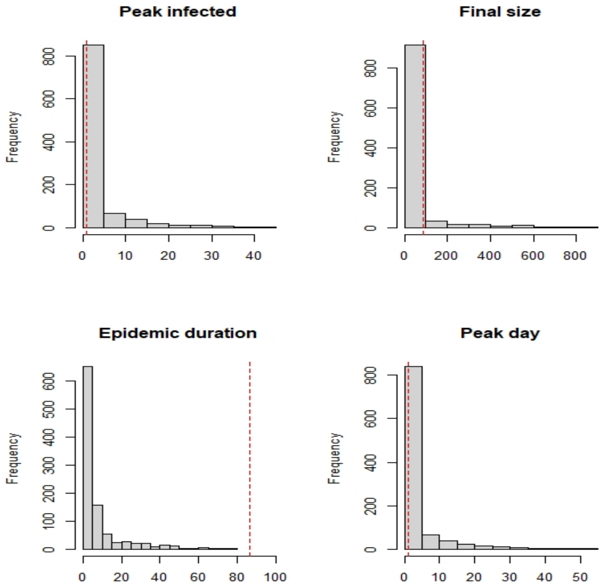


Figure 21: Histogram representing distributions of interest when $R_0 = 1$.

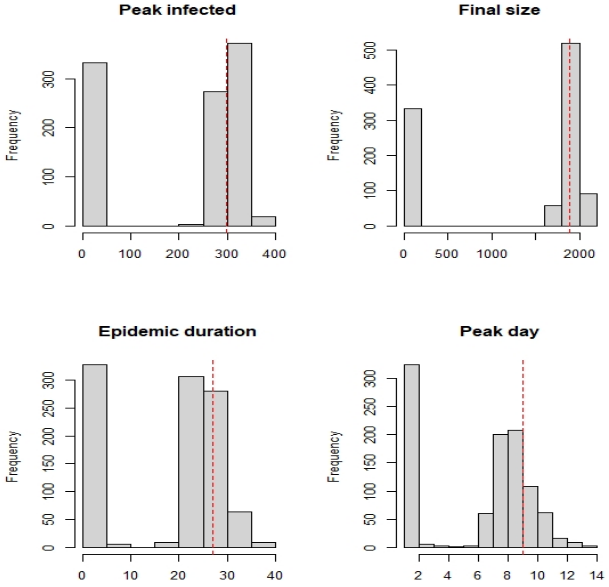


Figure 22: Histogram representing distributions of interest when $R_0 > 1$.

4.2.3 Initial number of infected

In this section, the stochastic model was ran 1000 times and the mean of the peak infected, final size, epidemic duration and peak day are compared with the deterministic model over $I_0 = 1, ..., 15$. Initial conditions that are used for this section are $\beta = 1.5$, $\alpha = 0.5$ and $N = 1000$. The key properties of each model are compared and shown in Figure 23. The results show that the mean dynamics of the stochastic model becomes closer to the deterministic model as I_0 increases. It is also interesting to note that when $I_0 = 1$, the epidemic has an approximate probability of 0.7 to happen whereas when $I_0 > 1$, the epidemic happens almost surely with probability 1.

4.2.4 Recovery time

In this section, we will analyse how recovery time of each infected individual affects the epidemic curve. As we have discussed in earlier sections, γ is the rate of recovery of each infected individuals. So, each infected individuals recover in $1/\gamma$ days. We will compare how the recovery time affects the key properties of the epidemic as it is in the last section. Initial conditions used are $I_0 = 1$, $\beta = 1.5$, and $N = 1000$. The results are plotted in Figure 24. Similar to the last section, we compare the mean of 1000 simulations from the stochastic model with the deterministic model. We can observe that the mean dynamics of the stochastic model is very similar to the deterministic model. Figure 25 shows the plot of how increasing the number of days for an infected individual to recover affects the probability of an epidemic happening. Unsurprisingly, the longer the time required for an average infected individual to recover, the higher the probability of an epidemic.

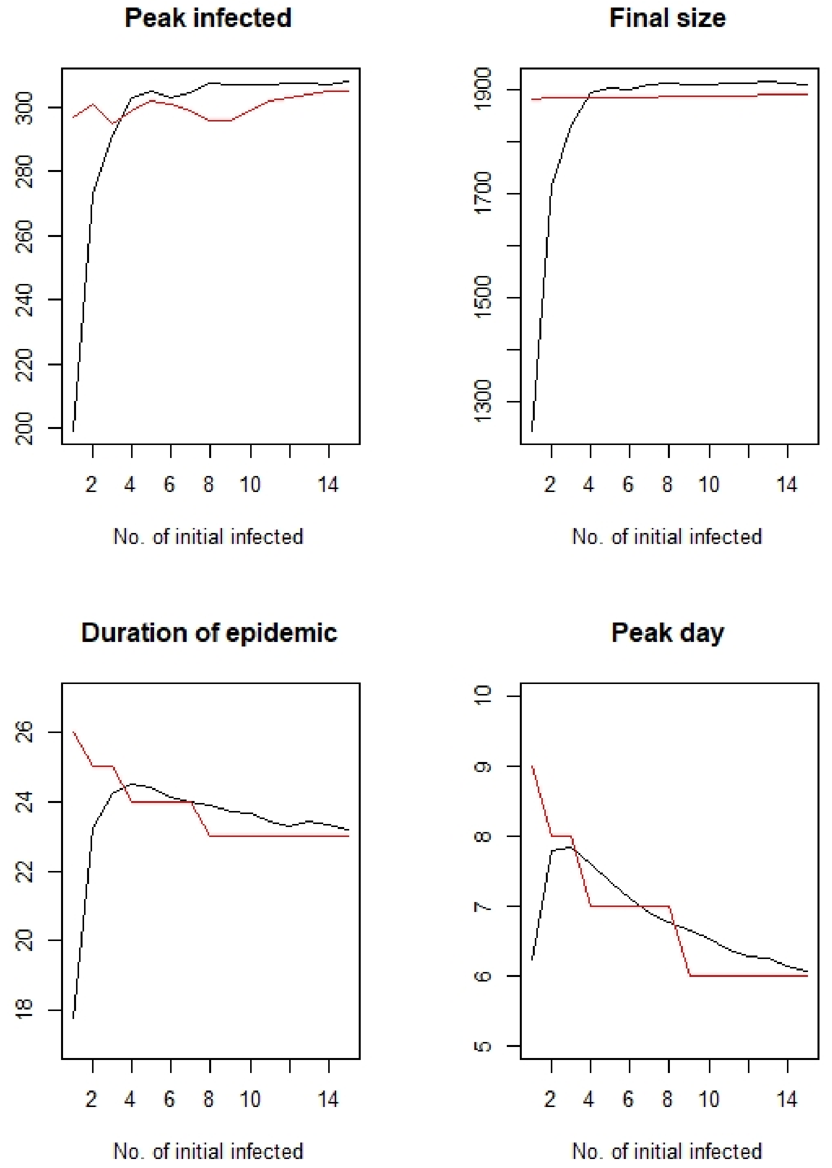


Figure 23: Figure shows the effect of I_0 on the four key properties of an epidemic. Black line represents the mean of 1000 simulations of the stochastic model whereas the red line represents the deterministic model.

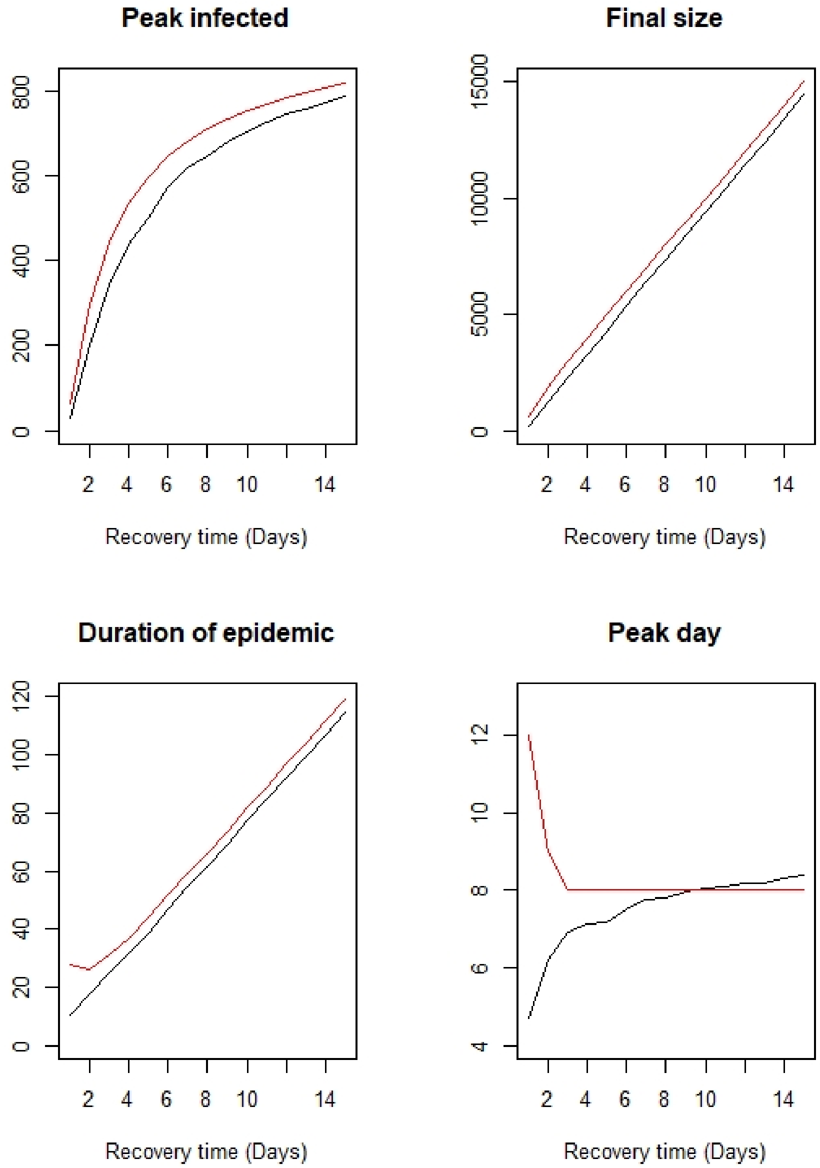


Figure 24: Figure shows the effect of γ on the four key properties of an epidemic. Black line represents the mean of 1000 simulations of the stochastic model whereas the red line represents the deterministic model.

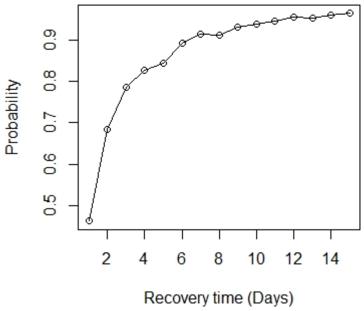


Figure 25: Probability of an epidemic happening as the number of days for an individual to recover from the disease increases.

5 Conclusion

To conclude, the purpose of this research project is to provide an introduction to compartmental epidemic model, specifically SIR model. Due to the complexity of such model, particularly stochastic epidemic models, the likelihood function of such model is intractable. Therefore, we introduced ABC as our method of inference, which is through repeated sampling from the model. We have also introduced different computation algorithms and compared them based on computation time and accuracy of estimates. Moreover, we made some comparisons between the deterministic and stochastic SIR models through simulations and implementing them on Covid-19 data in the UK, namely England and Bristol. Through our study, we were able to determine that stochastic models are especially important in epidemic modelling when population size is small. Moreover, stochastic models give us an opportunity to better quantify our uncertainties about the parameter estimates.

An important issue which this research project did not mention is parameter identifiability. However, this should not be a problem with the models we constructed. This paper [50] has shown that the parameters of SIR and SEIR type deterministic models are structurally identifiable when the measured output is cumulative number of infected people. Based on this paper from [51], "parameter identifiability issues are more likely to arise with more complex models (based on number of states and parameters)". Therefore, it is important to perform parameter identifiability tests before reporting results from the model. Note that one should also be cautious when interpreting results from the model when only data early in the pandemic is available which is shown in [52].

This research project has provided some elementary knowledge to compartmental epidemic modelling using ABC. For me, some interesting avenues that I would be interested to pursue in the future are modelling epidemic using time-varying parameters, implementing complex SEIR-type models to model the

Covid-19 pandemic more closely, and comparing ABC with state space method such as Particle Markov chain Monte Carlo [53].

References

[1] MG Garner, SA Hamilton, et al. Principles of epidemiological modelling. *Revue Scientifique et Technique-OIE*, 30(2):407, 2011.

[2] History.com Editors. Pandemics that changed history. https://www.history.com/topics/middle-ages/pandemics-timeline.

[3] History.com Editors. Black death. https://www.history.com/topics/middle-ages/black-death, Sep 2010.

[4] History.com Editors. Spanish flue. https://www.history.com/topics/world-war-i/1918-flu-pandemic, Oct 2010.

[5] Worldometer.info. Coronavirus cases. https://www.worldometers.info/coronavirus/?utm_campaign=homeAdvegas1%3F%22, Apr 2021.

[6] Valeria Saladino, Davide Algeri, and Vincenzo Auriemma. The psychological and social impact of covid-19: new perspectives of well-being. *Frontiers in psychology*, 11:2550, 2020.

[7] David Brown Lora Jones, Daniele Palumbo. Coronavirus: How the pandemic has changed the world economy. https://www.bbc.co.uk/news/business-51706225, Jan 2021.

[8] Daniel Bernoulli and Sally Blower. An attempt at a new analysis of the mortality caused by smallpox and of the advantages of inoculation to prevent it. *Reviews in medical virology*, 14(5):275, 2004.

[9] Nicolas Bacaër. Ross and malaria (1911). In *A short history of mathematical population dynamics*, pages 65–69. Springer, 2011.

[10] 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.

[11] Tapiwa Ganyani, Christel Faes, and Niel Hens. Simulation and analysis methods for stochastic compartmental epidemic models. *Annual Review of Statistics and Its Application*, 8, 2021.

[12] William Ogilvy Kermack and Anderson G McKendrick. Contributions to the mathematical theory of epidemics. ii.—the problem of endemicity. *Proceedings of the Royal Society of London. Series A, containing papers of a mathematical and physical character*, 138(834):55–83, 1932.

[13] Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25):2340–2361, 1977.

[14] George EP Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976.

[15] Peter J Diggle and Richard J Gratton. Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2):193–212, 1984.

[16] Richard J Rossi. *Mathematical statistics: an introduction to likelihood based inference*. John Wiley & Sons, 2018.

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

[18] Jonathan K Pritchard, Mark T Seielstad, Anna Perez-Lezaun, and Marcus W Feldman. Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798, 1999.

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

[20] Mark A Beaumont. Approximate bayesian computation. *Annual review of statistics and its application*, 6:379–403, 2019.

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

[22] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.

[23] VA Epanechnikov. Nonparametric estimates of a multivariate probability density. theor. *Probab. Appl*, 14:153–158, 1969.

[24] Brian Amberg. File:kernels.svg, Dec 2008.

[25] Wentao Li and Paul Fearnhead. Convergence of regression-adjusted approximate bayesian computation. *Biometrika*, 105(2):301–318, 2018.

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

[27] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[28] Marco Taboga. Generalized least squares. https://www.statlect.com/fundamentals-of-statistics/generalized-least-squares, 2017. Lectures on probability theory and mathematical statistics, Third edition. Kindle Direct Publishing. Online appendix.

[29] Jarno Lintusaari, Michael U Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Fundamentals and recent developments in approximate bayesian computation. *Systematic biology*, 66(1):e66–e82, 2017.

[30] David T Frazier, Christian P Robert, and Judith Rousseau. Model misspecification in abc: consequences and diagnostics. *arXiv preprint arXiv:1708.01974*, 2017.

[31] 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.

[32] 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.

[33] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

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

[35] Luke Bornn, Natesh S Pillai, Aaron Smith, and Dawn Woodard. The use of a single pseudo-sample in approximate bayesian computation. *Statistics and Computing*, 27(3):583–590, 2017.

[36] 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.

[37] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[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] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.

[40] Daniel Wegmann, Christoph Leuenberger, and Laurent Excoffier. Efficient approximate bayesian computation coupled with markov chain monte carlo without likelihood. *Genetics*, 182(4):1207–1218, 2009.

[41] Franck Jabot, Thierry Faure, Nicolas Dumoulin, and Carlo Albert. *EasyABC: Efficient Approximate Bayesian Computation Sampling Schemes*, 2015. R package version 1.5.

[42] UK Government. Official uk coronavirus dashboard. https://coronavirus.data.gov.uk, 2021.

[43] Libo Sun, Chihoon Lee, and Jennifer A Hoeting. Parameter inference and model selection in deterministic and stochastic dynamical models via approximate bayesian computation: modeling a wildlife epidemic. *Environmetrics*, 26(7):451–462, 2015.

[44] Aaron A. King. Introduction to partially observed markov processes. https://kingaa.github.io/short-course/intro/intro.html.

[45] Karline Soetaert, Thomas Petzoldt, and R. Woodrow Setzer. Solving differential equations in R: Package deSolve. *Journal of Statistical Software*, 33(9):1–25, 2010.

[46] Garry Blight, Frank Hulley-Jones, Pablo Gutiérrez, Lydia McMullan, and Pamela Duncan. Covid chaos: how the uk handled the coronavirus crisis. https://www.theguardian.com/world/ng-interactive/2020/dec/16/covid-chaos-a-timeline-of-the-uks-handling-of-the-coronavirus-crisis, Feb 2021.

[47] Neil Park. Population estimates for the uk, england and wales, scotland and northern ireland: mid-2019. https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationestimates/bulletins/annualmidyearpopulationestimates/mid2019estimates#population-growth-in-england-wales-scotland-and-northern-ireland, Jun 2020.

[48] Bristol City Council. The population of bristol. https://www.bristol.gov.uk/statistics-census-information/the-population-of-bristol, Mar 2020.

[49] Adam Hayes. Coefficient of variation (cv). https://www.investopedia.com/terms/c/coefficientofvariation.asp, Apr 2021.

[50] Gergely Horváth and Gábor Szederkényi. Identifiability analysis and parameter estimation of a seir epidemic model. In *Pannonian Conference on Advances in Information Technology (PCIT 2020)*, page 25, 2020.

[51] Kimberlyn Roosa and Gerardo Chowell. Assessing parameter identifiability in compartmental dynamic models using a computational approach: application to infectious disease transmission models. *Theoretical Biology and Medical Modelling*, 16(1):1–15, 2019.

[52] Timothy Sauer, Tyrus Berry, Donald Ebeigbe, Michael M Norton, Andrew Whalen, and Steven J Schiff. Identifiability of infection model parameters early in an epidemic. *medRxiv*, 2020.

[53] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.