UiT

THE ARCTIC
UNIVERSITY
OF NORWAY

Faculty of Science and Technology

Department of Mathematics and Statistics

Mathematics of Viral Infections: A Review of Modeling Approaches and A Case-Study for Dengue Dynamics

—

Name : Yong Chung Han
Supervisor : Martin Rypdal

Master thesis in Applied Mathematics – August 2018

UNIVERSITY OF TROMSØ

Contents

Abstract

In this thesis we use mathematical models to study the mechanisms by which diseases spread. Transmission dynamics is modelled by the class of SIR models, where the abbreviation stands for susceptible (S), infected (I) and recovered (R). These models are also called compartmental models, and they serve as the basic mathematical framework for understanding the complex dynamics of infectious diseases. Theory developed for the SIR framework can be applied the real-world dynamics, for instance to the spread of the dengue virus. We look at how parameters such as the as *basic reproduction number*, R_0, drive epidemics by allowing transitions from a disease-free equilibrium (DFE) when $R_0 < 1$ to an endemic equilibrium (EE) when $R_0 > 1$. A case study was carried out to investigate dengue transmission dynamics in a single-serotype model by using a vector-to-human compartmental model. Here the approach is to explore the underlying dynamical structures, as well as looking at the projected impact of possible interventions such as vaccines and vector-control measures.

Acknowledgements

First and foremost, I would like to thank my advisor, Martin Rypdal, for his mentorship, guidance, patience, time, and input throughout my graduate studies at UiT – The Arctic University of Norway. I would also like to thank Kristoffer Rypdal for his assistance and constructive criticism.

Lastly I would like to thank my parents. My parents gave up many things to ensure that I would have the opportunities I have in life. Their sacrifices have molded me into the man I am, and there aren't enough worlds to explain how thankful I am for them. I love you mom and dad, thanks for always getting better grades and being smarter than me.

Last but not the least, you know who you are, thank you for putting up with me going nerd and not having time to be social. Thanks for the memories!

Chapter 1

Introduction

Infectious diseases burden communities and societies throughout the world. As the incidence of an infectious disease starts to increase in any population, people start to look for methods that are most effective in combating the outbreak, or at least control the number of infections. Scientists have made tremendous progress in fight against diseases. Yet, infectious diseases remain a major cause of mortality. In epidemiology, one aims to investigate the progress of well-being and diseases in a specific population in order to control related-health problems, and in this thesis we use mathematics to describe complex disease dynamics using simplifications and hypotheses about the relevant mechanisms.

1.1 Background

Mathematical models have become an important tools for breaking down and analyzing the spread of infectious diseases. They help our *understanding* and facilitate *predictions*. Models are also used to test the plausibility of epidemiological explanations. Another application is foreseeing the possible effects of changes system dynamics, and to predict structural changes through early warning signals. Thereby making it possible to control an emerging disease outbreak.

Mathematical epidemiology has a long history. The first epidermal model was formulated by Daniel Bernoulli [1] in the 18th century. Bernoulli was trained as physician and was a member of a famous family of mathematicians. Based on a theoretical approach to the effects of a disease, his first published model demonstrated increased life expectancy for individuals vaccinated against smallpox. About one hundred years later, the Russian physician EnKo [2, 3] used a binomial probabilistic model for describing the epidemic of measles in discrete time. Since then, several simple models have been used to describe disease propagation on a population level. Hamer [4] hypothesized that the rate of transmission $\lambda = \beta I$ depends on the numbers of susceptible and infected. This is referred to as the *mass-action transmission* rate. Sir Ronald Ross [5, 6] formulated a continuous-time mathematical model for the transmission dynamics of malaria. Additionally, he also explained how the effectiveness of various intervention strategies for malaria. Up until that point, most work had been of an empirical nature, but that changed with the works of Kermack and Mckendrick [7], two students of Ross in 1927. They used a system of ordinary differential equations (ODEs) to formulate the threshold theorem, which states that the initial number (critical fraction) of susceptibles must be exceeded in order for an epidemic outbreak to occur. In 1969, important generalizations were made Severo [8, 9]. Shortly after, further generalizations were made by Anderson and May [10]. Instead of a direct product between the numbers of susceptible and infectious individuals, he considered that the probability of a new infection might be modeled as the product between the number of susceptible individuals raised to some power $1 - b$, and the number of infected to the power l. The parameters l and b are called the *"infection power"* and the *"safety-in-numbers power"*, respectively. In 1987, Liu [11] presented stability conditions similar to those found by Severo, but now generalized to the models of Anderson and May. Over the last two decades several new models have been proposed. Most of these models have a deterministic character, are highly simplified, and are omitting many filter details.

1.2 Epidemic Models

Mathematical models of epidemics are created under the assumption that the observed population can be divided into multiple subsets, called compartments. The simplest compartmental model was described by Kermack and McKendrick [7] in 1927. In its modern formulation, the Kermack-McKendrick Model (hereafter called the KM

Model) [12] is based on relatively simple assumptions on the rates of flow between the different compartments. It models the spread of a communicable disease using a latency period and a general mode of transmission. Non-linear transmission is described by the *SIR* and *SEIR* models. These models include the effect of immunity against re-infection. This implies that there is a flow of individuals from the susceptible class "S" to the exposed "E", to the infected "I". After an infection individuals enter the class "R", indicating that they are removed from the population of interest, either through death or through immunity. There are many hypotheses underlying this model. For instance, the population is assumed to be large and closed. Also, natural births and deaths during the outbreaks are disregarded. Other simplifications are the lack of a latency period (individuals become infectious as soon as they become infected), lifetime immunity after recovery, and homogeneous mixing [13].

1.3 Stochastic versus Deterministic Models

Both stochastic and deterministic models are applicable and useful in the study infectious diseases, and for identifying strategies for their prevention at the population scale.

Stochastic models have been successfully used in the framework of very complicated systems in many fields of science [14]. They rely on chance variation in risk of exposure, and this gives better insights into an individual-level modelling. Their individual roles potentially incorporate a large amount of heterogeneity and complexity, which give much more insightful monitoring. That being the case, they can be difficult set up, the results could potentially be meaningless.

By contrast, deterministic models are natural first models when faced with a new problem [15]. In epidemiology, deterministic models are generally better in explaining what happens with respect to spatial dynamics when dealing large populations, since the larger the population, the better the is assumptions of homogeneity (i.e each person in a given class is equivalent to the others). The disadvantage is that their imposed structure of generality removes the possibility of embedding more realistic infection profiles.

Most of the models that have been used to describe characteristic behaviour patterns in infectious diseases are deterministic. They are hence less dependent on high-quality data, easier to set up, and compatible computer software are widely available and user friendly. The most fundamental model is the SIR model. And most other deterministic models, including (SIRS and SEIR, etc) are regarded as extensions of the SIR model. The major challenges are to understand the limitations of these models, and to understand the relationship between the deterministic models and their corresponding stochastic approximations.

Chapter 2

Mathematical modelling of outbreaks

2.1 Models and Notation

To understand the mechanism of disease transmission a well established technique is need in order to do inference for epidermic models. The most basic assumptions is to divide the population into three sub-groups. These sub-groups being defined by health status, exposure to the pathogen, demographic or epidemiological features. The second aim is predicting the past and future temporal course.

This simplest model consists of three different compartments, and the ratio of *Susceptible (S)*, *Infected (I)* and *Recovered or Resistant (R)* in a large population. In deterministic models, all these variables are functions in discrete time $t = 0, 1, 2, \ldots$ or differentiable functions in continuous time $t \geq$. The key variable are:

- S*usceptibles* (ratio in population is denoted S): Individuals in the population who have not been infected. They are healthy but at risk of becoming infected. Once they have contracted the infection, they move into the infected sub-group.

- I*nfected* (ratio in population is denoted I): Infected individuals who are contagious or are carriers. They can infect susceptible individuals.

- R*ecovered* or R*emoved* (ratio in population is denoted R): Individuals who have recovered or died from the disease. Unfortunately, the SIR model does not describe a difference between immunity, non-immunity, or even innate immunity.

2.2 Basic Assumptions

The SIR models all share several core assumptions [16, 17]:

1. The total size of host population remains constant ($S + I + R = N$).
2. The population must mix homogeneously.
3. It will not allow any host demographic turnover (either birth or death) in the period of the epidemic, and all infections are assumed to end with recovery or removal from compartments.
4. - A person can leave or discharge from the susceptible compartment only by becoming infected.
   - A person can leave or discharge from the infected compartment only by recovering from the disease.
5. The probability of being infected does not depend on factors such as age, gender or social status.
6. Infected are not subject to quarantine procedures.
7. During epidemics, susceptible isolate themselves from infected, or take other protective measures.
8. The recovery rate is constant in time.
9. The dynamical equations are of first order:

$$\begin{aligned}\frac{dS}{dt} &= -\text{ rate of new infections}\\ \frac{dI}{dt} &= \text{rate of new infections} - \text{rate of recovery}\\ \frac{dR}{dt} &= \text{rate of recovery}\end{aligned}$$

2.3 The SIR model

The model is based on a few simple assumptions [17]:

New infections occur through contact between infected and susceptible hosts, and the rate of change is proportional to the number of interactions. This is the product S(t), I(t) with a constant α-parameter. The number of susceptibles decrease as individuals come into contact with the infected:

$$\frac{d}{dt}S(t) = -\beta\frac{S(t)I(t)}{N} \tag{2.1}$$

When susceptibles become infected, members leave the susceptible compartment and join the infected compartment with rate $\alpha S(t)I(t)$. Thus, the total population of infected hosts increase. Vice versa, the hosts leave the infected compartment and join the recovered group. Since β is assumed constant, this implies that the rate of change is dependent with time as the size of the infected group varies:

$$\frac{d}{dt}I(t) = \beta\frac{S(t)I(t)}{N} - \gamma I(t) \tag{2.2}$$

Since infected carriers can only leave their compartment by joining the new R-compartment, it only changes through addition of those recovered from infection. The recovery rate is given by the constant parameter β:

$$\frac{d}{dt}R(t) = \gamma I(t) \tag{2.3}$$

The diagram below illustrates the dynamics of the classic SIR model:

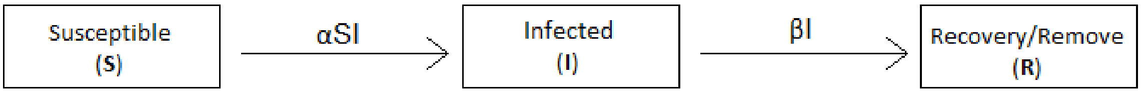

Figure 2.1: A flow diagram demonstrating the relationships between Susceptible (S), infected (I) and recovered (R)

.

2.4 Modified Differential Equations

As mentioned above, a modified version of the system of SIR differential equations will be more appropriate and realistic. A more general system is the SEIR model. As is evident from its name, the SEIR model contains one more compartment compared to the SIR model. The letter E represents the set of exposed individuals in the exposed incubation phase, during which one is infected, but not yet infectious.

In the next several chapters of this thesis, we will discuss a particular vector-borne infectious disease called dengue fever. With the SEIR model, we can narrow our focus to model dengue fever by developing a model for the coupled dynamics of disease prevalence in humans and in mosquitoes (vectors), and investigate certain measures for controlling dengue fever.

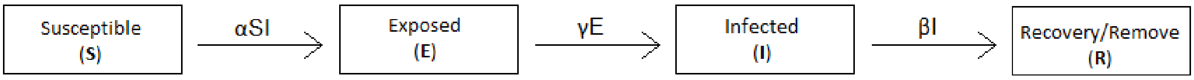

Figure 2.2: Flow diagram of the SEIR model illustrates the transitions among (S), Exposed (E), Infective (I) and Recovered (R).

In many situations, individuals can-not infect susceptibles immediately after they get infected. It is only after sufficient colonization by the pathogen that transmission can occur, i.e. there is a threshold on the pathogen abundance, which gives rise to a new compartment E. The equations become [18]:

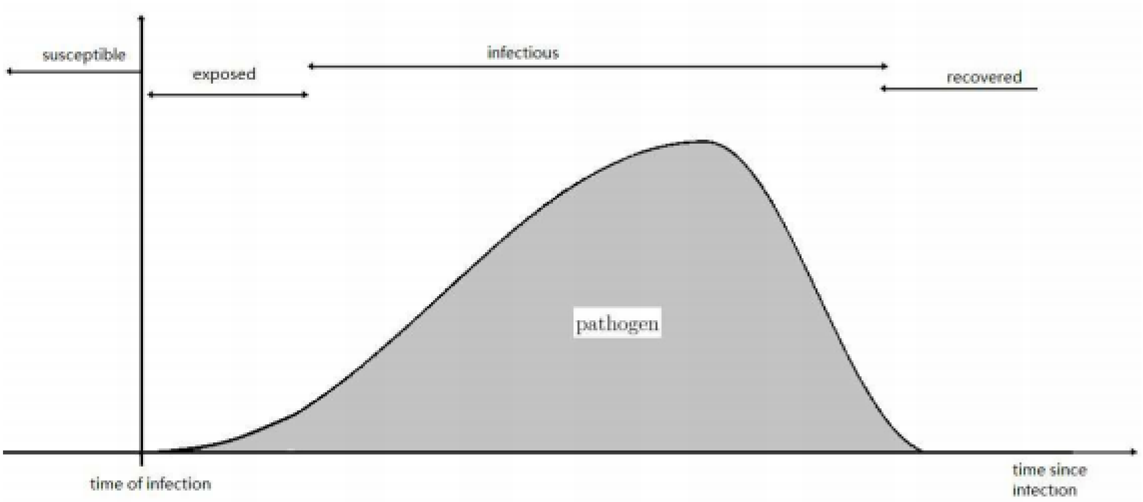

Figure 2.3: The time-line of the infections showing the four analytical dynamics of the pathogen and the infection classes: Susceptible (S), Exposed (E), Infected (I) and Recovered (R).

$$
\begin{aligned}
\frac{d}{dt}S(t) &= -\beta\frac{S(t)I(t)}{N} \\
\frac{d}{dt}E(t) &= \beta\frac{S(t)I(t)}{N} \ - \ \kappa E(t) \\
\frac{d}{dt}I(t) &= \kappa E(t) \ - \ \gamma I(t) \\
\frac{d}{dt}I(t) &= \gamma I(t)
\end{aligned}
$$

2.5 The Basic Reproductive Number, R_0

An important parameter in disease modelling is the *Basic Reproductive Ratio*, denoted R_0. This parameter tells us if a population is at risk of an epidemic. The reproductive rate is the number of secondary infections produced by the primary infection into the total susceptible population [19], and it can be used to predict who will not become infected as $t \to \infty$. R_0 as a dimensionless number that determines the threshold condition for the disease-free equilibrium **DFE**. It can be expressed as a product of three quantities [20]:

$$
R_0 \propto \begin{pmatrix}\text{Probability of}\\ \text{transmission}\\ \text{per contact}\end{pmatrix} \cdot \begin{pmatrix}\text{Number of}\\ \text{contacts}\\ \text{per unit time}\end{pmatrix} \cdot \begin{pmatrix}\text{Duration of}\\ \text{infection}\end{pmatrix}
$$

$$
R_0 \propto \left(\frac{\text{infection}}{\text{contact}}\right) \cdot \left(\frac{\text{contact}}{\text{time}}\right) \cdot \left(\frac{\text{time}}{\text{infection}}\right)
$$

More specifically:

$$R_0 = pcd$$

where

- p is the transmission probability
- c is the contact rate
- d is the duration of the infectious periods

If $R_0 > 1$ then the disease invades. It increases to reach its maximum and then decreases to zero. The **DFE** is unstable. If $R_0 < 1$ the disease dies out. It decreases monotonically to zero. So the **DFE** is stable [21].

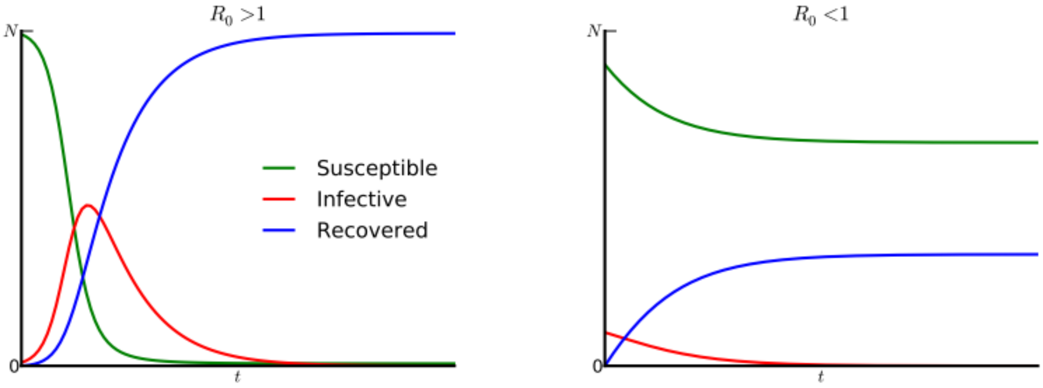

Figure 2.4: The time evolution of all classes showing the epidemic curves in the SIR model.

2.5.1 Epidemic SIR model

To determine if there is an epidemic, we look at the stability of the disease free equilibrium. We only need to consider the variable $I(t)$. The condition for an epidemic to occur is $dI/dt > 0$. We have:

$$\beta \frac{SI}{N} - \gamma I > 0$$

$$\Updownarrow$$

$$\frac{\beta}{\gamma}\frac{S}{N} > 1$$

At the beginning of an epidemic, almost everyone is susceptible, i.e. implies $S \approx 1$. For $S = 1$ we obtain the condition

$$\frac{\beta}{\gamma} = R_0 > 1 \tag{2.4}$$

The phase plane (SI-plane)

The variable $R(t)$ can be disregarded when studying the dynamics of the SIR model. We can derive a useful analytic result by dividing the equations for $\dot{I}$ by $\dot{S}$, and making use of the Chain Rule [22]:

$$\begin{aligned}
\frac{dI}{dS} &= \frac{\left(\frac{dI}{dt}\right)}{\left(\frac{dS}{dt}\right)} \\
&= \frac{\beta SI/N - \gamma I}{-\beta SI/N} \\
&= -1 \ + \ \frac{\gamma I}{\beta SI/N} \\
&= \frac{\gamma N}{\beta S} \ - \ 1
\end{aligned}$$

Multiplying both sides by dS gives

$$\begin{aligned}
dI &= \left(\frac{\gamma N}{\beta S} \ - \ 1\right) dS \\
\int_0^t 1 \ dI &= \int_0^t \left(\frac{\gamma N}{\beta S} \ - \ 1\right) dS \\
\big[I\big]_0^t &= \left[\frac{\gamma N}{\beta}\ln(S) \ - \ S\right]_0^t \\
I(t) \ - \ I_0 &= \left[\frac{\gamma N}{\beta}\ln(S) \ - \ S\right]_t \ - \ \left[\frac{\gamma N}{\beta}\ln(S) \ - \ S\right]_0 \\
I(t) \ - \ I_0 &= \frac{\gamma N}{\beta}\ln S(t) \ - \ S(t) \ - \ \frac{\gamma N}{\beta}\ln S_0 \ + \ S_0 \\
\underbrace{I_0 \ + \ S_0 \ - \ \frac{\gamma N}{B}\ln S_0}_{constant} &= \underbrace{I(t) \ + \ S(t) \ - \ \frac{\gamma N}{B}\ln S(t)}_{constant}
\end{aligned}$$

Hence,

$$I(t) \ - \ I_0 = \frac{\gamma N}{\beta}\ln\left(\frac{S(t)}{S_0}\right) \ - \ S(t) + \ S_0 \tag{2.5}$$

and

$$\begin{aligned}
I(t) \ - \ I_0 &= \frac{\gamma N}{\beta}\ln\left(\frac{S(t)}{S_0}\right) \ - \ S(t) + \ S_0 \\
I(t) &= I_0 \ + \ S_0 \ - \ S(t) \ + \ \frac{\gamma N}{\beta}\ln\left(\frac{S(t)}{S_0}\right) \\
&= I_0 \ + \ S_0 \ - \ S(t) \ + \ \frac{N}{R_0}\ln\left(\frac{S(t)}{S_0}\right) \qquad (2.6)
\end{aligned}$$

We can infer that γ/β is the inverse of R_0. We denote $\rho = \gamma/\beta$. This is the ***relative removal rate*** [19]. Note that γ has units of 1/time and gives the removal rate from the infected group.

2.6 The Threshold Phenomenon

To understand what factors will determine whether or not an epidemic will occur, or if the infection will fail to invade, we consider the initial stage after the infection is introduced into a population consisting of $S(0)$ susceptibles. We start by rewriting the equation for dI/dt on the form

$$\begin{aligned}\frac{dI}{dt} &= \beta S(0)I(0) \ - \ \gamma I(0)\\ &= S(0)I(0) \ - \ \frac{\gamma}{\beta}I(0)\\ &= S(0)I(0) \ - \ \rho I(0)\end{aligned}$$

It is easily to show that the disease always dies out if the initial ratio $S(0) \times I(0)$ is less than $\rho = \gamma/\beta$, since then $\frac{dI}{dt} < 0$.

We can determine the maximum point on the curve in SI-plane as follows. With the normalization $N = 1$ we have

$$\begin{aligned}\frac{dI}{dS} &= \frac{\rho N}{S} \ - \ 1\\ &= \frac{\rho}{S} \ - \ 1\end{aligned} \tag{2.7}$$

and the condition for a critical point becomes:

$$\frac{dI}{dS} = 0$$

which implies that

$$\begin{aligned}&0 = \frac{\rho}{S} \ - \ 1\\ &\Updownarrow\\ &\rho = S\end{aligned} \tag{2.8}$$

The second derivative gives

$$\frac{d^2I}{dS^2} = -\frac{\rho}{S^2} \ < \ 0$$

Equating the value of the of the conserved quantity at $t = 0$ and at asymptotically (t $= \infty$), we get [17]

$$\begin{aligned}
I_{max} &= I(\infty) \\
&= I_0 \ + \ S_0 \ - \ S(\infty) \ + \ \frac{N}{R_0} \ln\left(\frac{S(\infty)}{S_0}\right) \\
&= \underbrace{I_0 \ + \ S_0}_{N_0} \ - S(\infty) \ + \ \underbrace{\frac{1}{R_0}}_{\rho} \ N \ln\left(\frac{S(\infty)}{S_0}\right) \\
&= N_0 \ - \ \underbrace{S(\infty)}_{S_\infty} \ + \rho N \ln\left(\frac{\overbrace{\boxed{S(\infty)}}^{S_\infty}}{S_0}\right) \\
&= N_0 \ - \ S_\infty \ + \ \rho N \ln\left(\frac{S_\infty}{S_0}\right) \qquad (2.9)
\end{aligned}$$

where the total population is actually constant, $N(t) = S(t) + I(t) + R(t)$ and equal to $N_0 = S_0 + I_0$.

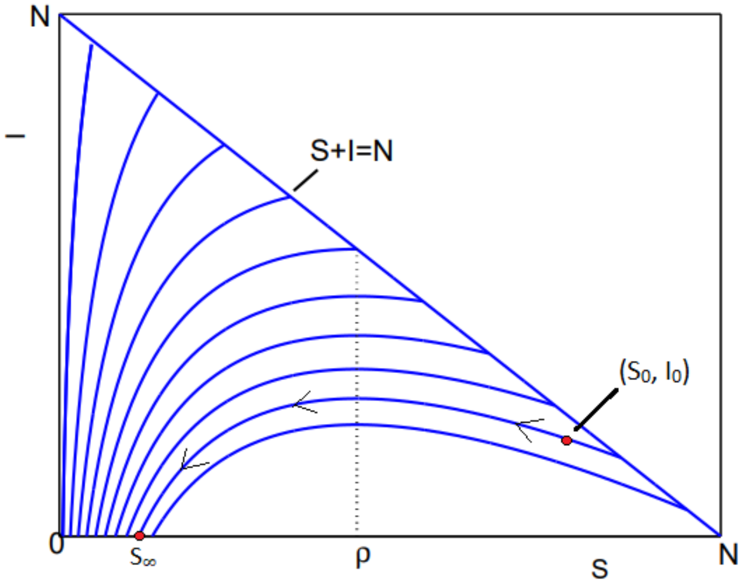

Figure 2.5: The SI phase plane trajectory system for SIR model; curves are (S), Infective (I).

The result can be expressed as [17]

$$\rho = \frac{N - S_\infty}{\ln\left(\frac{S_0}{S_\infty}\right)} \qquad (2.10)$$

As we can see from the parametric plots of $I(t)$ verses $S(t)$ in figure (2.5) on page 23, that the trajectories start with $S > \rho$ and *I(t)* increases from I_0, for any initial values. However, if $S_0 < \rho$ then I decreases from I_0 and no epidemic occurs. Recall that equation (2.7) shows that the solution $-\rho/S-1$ is positive for $S < \rho$, and negative for $S > \rho$.

2.6.1 The limiting number of susceptible individuals

Let S_∞ be the number of individuals not infected throughout the epidemic as $t \to \infty$. To evaluate the number of susceptibles at any time t, let us first divide equation (2.1) by (equation 2.3):

$$\begin{aligned}
\frac{dS}{dR} &= \frac{\left(\frac{dS}{dt}\right)}{\left(\frac{dR}{dt}\right)} \\
&= -\ \frac{\beta SI/N}{\gamma I} \\
&= -\ \frac{\beta S\cancel{I}}{\gamma \cancel{I} N} \\
&= -\ \underbrace{\frac{\beta}{\gamma}}_{1/\rho} \cdot \underbrace{\frac{S}{\boxed{N}}}_{1} \\
&= -\ \frac{1}{\rho} S(t) \qquad (2.11)
\end{aligned}$$

We integrate equation (2.11) [23, 24]:

$$
\begin{aligned}
\frac{dS}{dR} &= -\ \frac{1}{\rho}S(t) \\
\frac{1}{S}\ dS &= -\ \frac{1}{\rho}\ dR \\
\int_0^t \frac{1}{S}\ dS &= -\ \int_0^t \frac{1}{\rho}\ dR \\
\Big[\ln S\Big]_0^t &= -\ \frac{1}{\rho}\Big[R\Big]_0^t \\
\ln S(R) - \ln S_0 &= -\ \frac{1}{\rho}\Big(R(t) - R_0\Big) \\
\ln S(R) - \ln S_0 &= \ \frac{1}{\rho}R_0\ \ -\ \ \frac{1}{\rho}R(t) \\
\ln S(R) &= \ln S_0\ \ +\ \ \frac{1}{\rho}R_0\ \ -\ \ \frac{1}{\rho}R(t) \\
S[R(t)] &= e^{\left[\ln S_0\ +\ \frac{R_0 - R(t)}{\rho}\right]} \\
S(t) &= e^{\left[\ln S_0\right]}\cdot e^{\left[\frac{R_0 - R(t)}{\rho}\right]} \\
&= S_0 e^{\left[\frac{R_0 - R(t)}{\rho}\right]} \\
&= S_0 e^{-\left[\frac{R(t) - R_0}{\rho}\right]} \\
&= S_0 e^{-\left[\frac{R(t)}{\rho} - \cancelto{0}{\frac{R(0)}{\rho}}\right]} \\
&= S_0 e^{-\frac{R(t)}{\rho}} \quad \text{or} \quad S_0 e^{-R(t)R_0} \qquad (2.12)
\end{aligned}
$$

The number of susceptible individuals decreases as a function of the number of recovered individuals. The fact that the rate R/ρ also increases, means that any epidemic ends with there always being a portion of the population which will not be infected throughout the epidemic outbreak.

If we substitute R_0 into equation (2.12) [17]:

$$
S(R(t)) = S_0 e^{-\frac{R(t)}{\rho}} \qquad (2.13)
$$

It follows that

$$
\begin{aligned}
S_\infty &= S_0 e^{-\frac{R_\infty}{\rho}} \\
&= S_0 e^{-\frac{N - S_\infty}{\rho}} \qquad (2.14)
\end{aligned}
$$

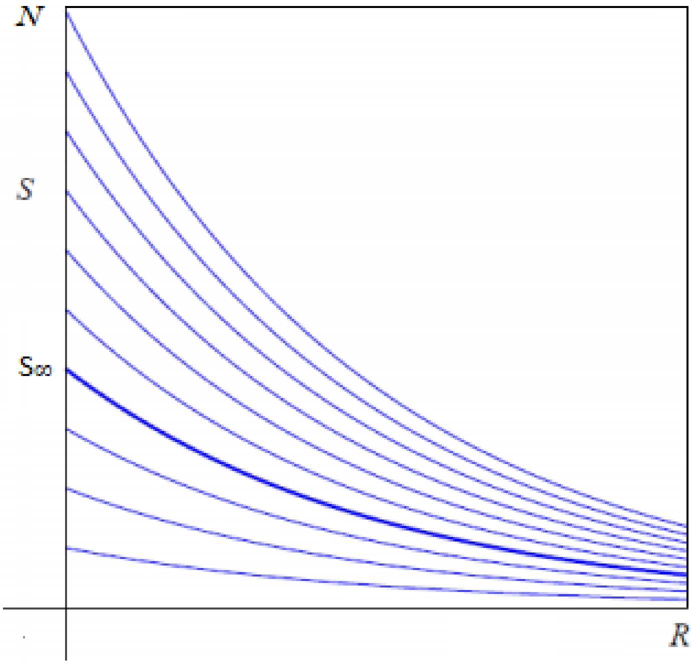

Figure 2.6: The RS phase plane trajectory system for the SIR model.

Alternatively, we can use this relation to compute the value of R_∞ for the final proportion of recovered hosts. We can rewrite the asymptotic behaviour of equation (2.14) as follows [23]:

$$\begin{aligned} S_\infty &= S_0 e^{-\frac{R_\infty}{\rho}} \\ S_0 e^{R_\infty R_0} &= 1 \;-\; R_\infty \\ 1 \;-\; R_\infty \;-\; S_0 e^{R_\infty R_0} &= 0 \end{aligned} \tag{2.15}$$

This transcendental equations relates the overall size of the epidemic to the basic reproductive number. Figure (2.7) shows that no epidemic occurs if $R_0 < 1$. However, there is a positive solution if $R_0 > 1$. This represents an outbreak of the infection. This provided the size population is well-mixed.

2.7 Results and Discussion

We examine the dynamics for various choices of *basic reproductive number*: $R_0 = 0.82, 1.005$, and 1.4. We performed all simulations and made graphics with the software ***R***.

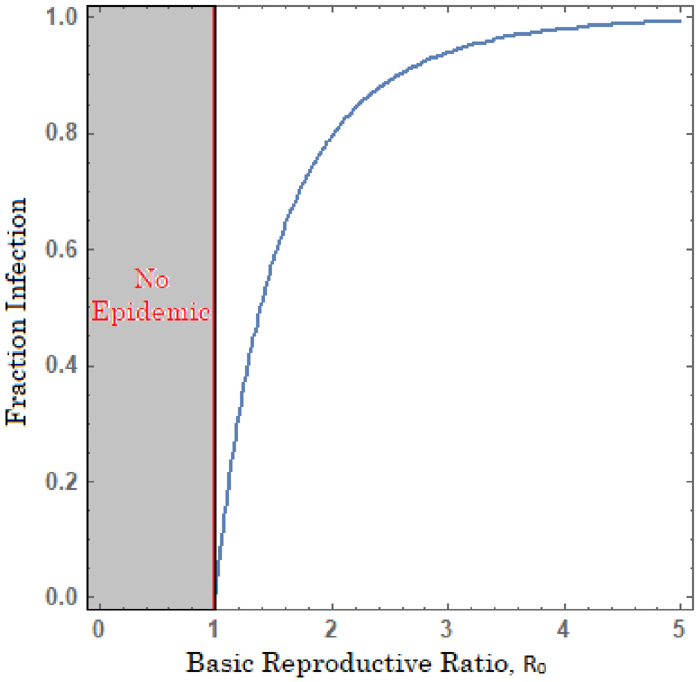

Figure 2.7: The total fraction of the population as a function of disease R_0.

From equation (2.14) we simplify further [23]:

$$\begin{aligned} S_\infty &= S_0 e^{-\frac{N-S_\infty}{\rho}} \\ &= S_0 e^{-\left(\frac{N}{\rho}-\frac{S_\infty}{\rho}\right)} \\ &= S_0 e^{-\left(\frac{1}{\rho}-\frac{S_\infty}{\rho}\right)} \\ &= S_0 e^{-\left(R_0-R_0S_\infty\right)} \\ &= S_0 e^{-R_0\left(1-S_\infty\right)} \end{aligned} \tag{2.16}$$

and so S_∞ is the positive root of the transcendental equation:

$$S_0 e^{-R_0\left(1-z\right)} = z \tag{2.17}$$

The *basic reproductive ratio*, from equation (2.8), can also be defined as

$$S = \frac{1}{R_0} \tag{2.18}$$

It tells us when the epidemic will peak [17]. Changes to the *basic reproductive number*, R_0 impact the transmission rate and the duration of epidemics.

2.8 Model fit and parameter sensitivity

In order to study the effect of key parameters on R_0, we performed a sensitivity analysis on R_0's dependency to β and γ. The fraction of susceptibles left in the population at the end pandemic, S_∞ is gauged by the numerical model simulation.

2.8.1 Basic reproductive number, $R_0 < 1$

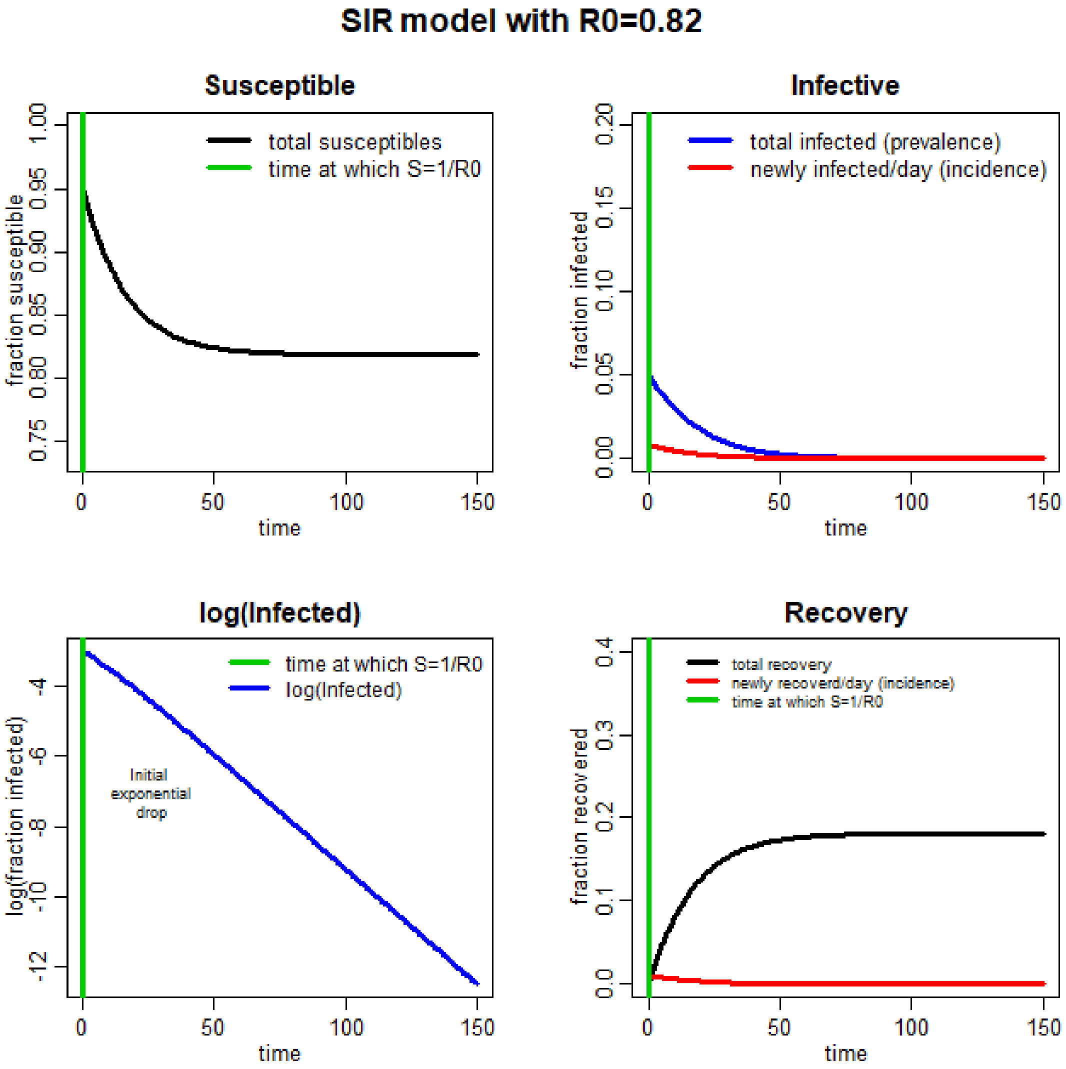

Figure 2.8: Trajectories of the SIR model. Parameters are $R_0 = 0.82$ and $\gamma = 0.2$.

For $R_0 \approx 0.82 < 1$, we see in figure (2.8) that we have a stable disease-free equilibrium in the infected compartment. The initial values are $S_0 = 9.50$ and $I_0 = 500$, where the infected hosts are 5% of the population. The infection will in this case die out in the long run without being able to replace themselves by new infections.

2.8.2 Basic reproductive number, $R_0 \approx 1$

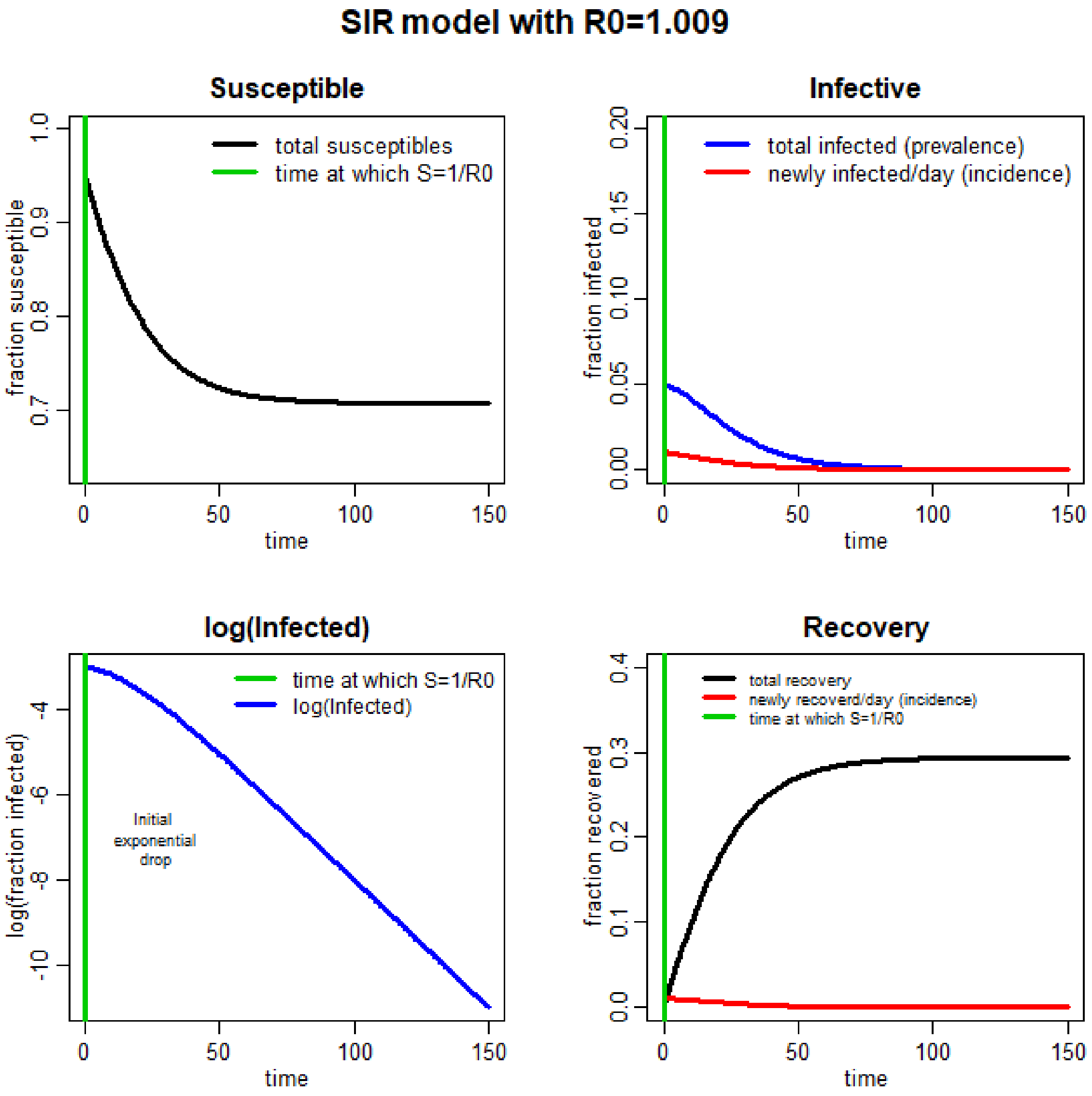

Figure 2.9: Trajectories of SIR model. Parameter values are $R_0 = 1.009$ and $\gamma = 0.2093$.

For the parameter values $\gamma = 0.2$ $R_0 = 1.009 \approx 1$ we see in figure (2.9) that deterministic system (2.1-2.3) is near a critical value of the parameter R_0, close to where the system undergoes a bifurcation from a stable disease-free equilibrium with no endemic equilibrium to an unstable disease-free equilibrium with a stable endemic equilibrium. As before, the initial values are $S_0 = 9.50$ and $I_0 = 500$, and the where the infected hosts are 5% of the population.

2.8.3 Basic reproductive number, $R_0 > 1$

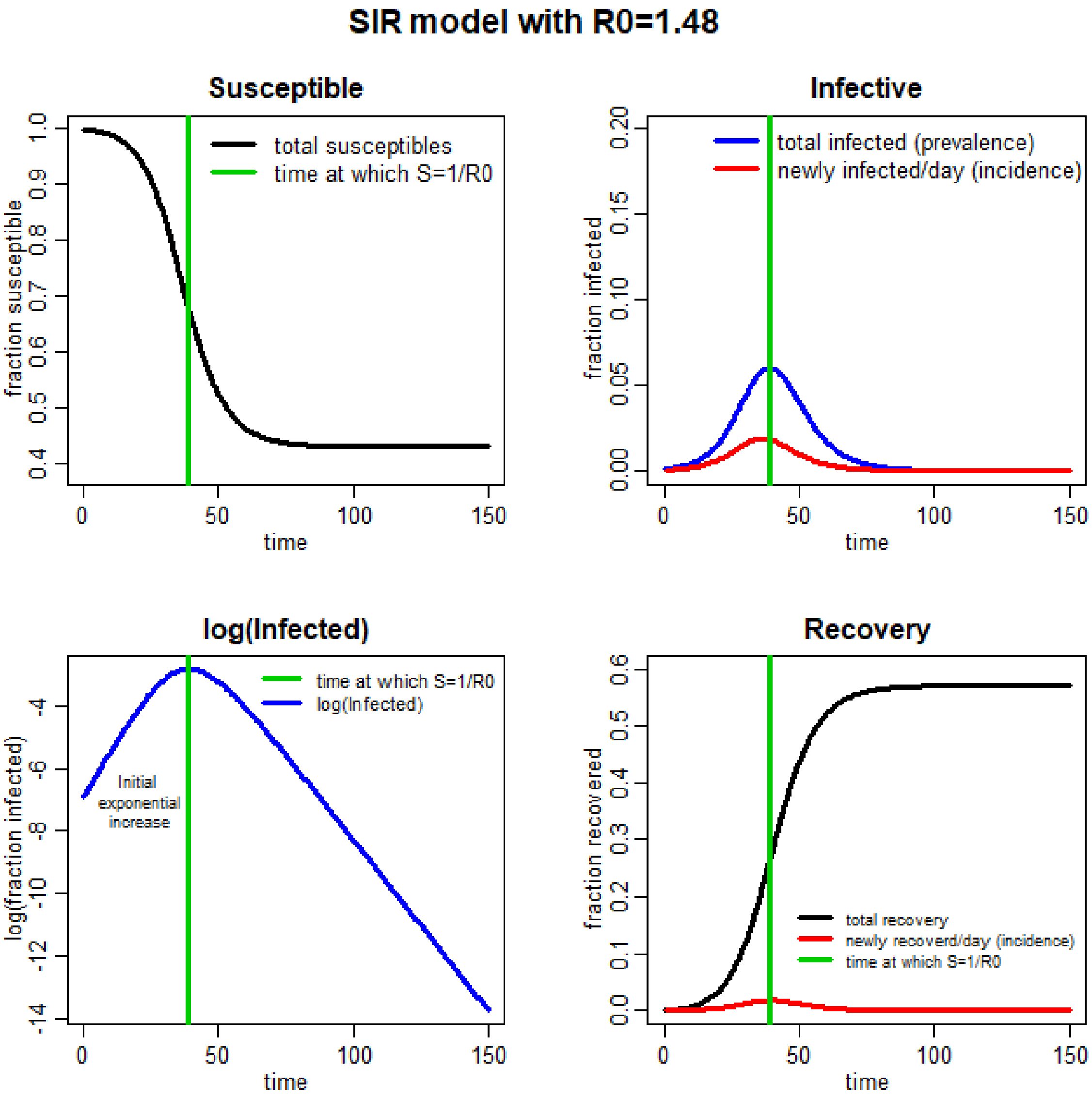

Figure 2.10: Trajectories of SIR model. Parameters are $R_0 = 1.48$ and $\gamma = 0.3$.

For $\beta = 1.48$ and $R_0 \approx 1.48 > 1$ we have a stable endemic equilibrium. See figure (2.10). The initial values are $S_0 = 0.999$ and $I_0 = 0.001$, and the infected hosts are 0.1% of the population. We can see that incidence reaches a maximum and then decreases. In fact, $I(t)$ increases as long as $S > \gamma/\beta$.

Chapter 3

Modified of SIR - Vital with Demographic

In the last chapter we presented the basic framework for the SIR model given the assumptions of a closed population without demographics (no births, deaths or migrations). This scenario is rather naive and unrealistic. Clearly demographic processes will be important. The easiest, and most common, way of introducing the influx of susceptibles is through birth, and to assume that there is a natural host life-span, $1/\mu$.

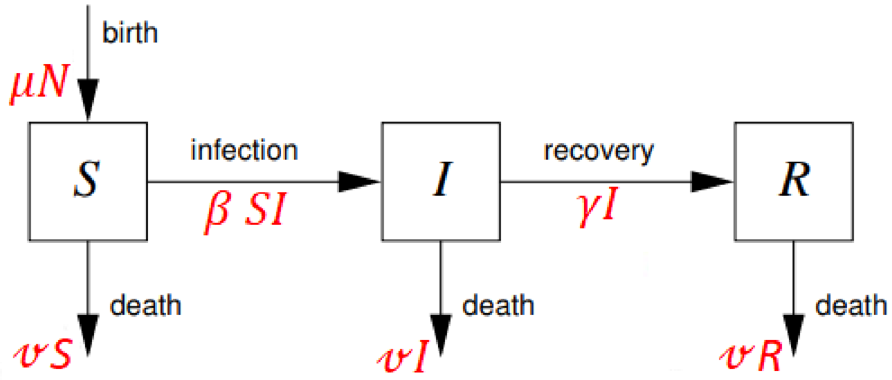

Figure 3.1: Flowchart showing transition rates between subsets in the SIR model with demographics.

Our equations become [25]:

$$
\begin{aligned}
\frac{dS}{dt} &= -\frac{\beta SI}{N} + \mu(N-S) \\
\frac{dI}{dt} &= \frac{\beta SI}{N} - \gamma I - \mu I \\
\frac{dR}{dt} &= \gamma I - \mu R
\end{aligned}
$$

This is a model where total fraction population size is bounded, $S + I + R = N$. We could then follow the approach proposed by [26] in which the total faction population size conserved $N = 1$. This requires that birth and death rates are equal ($\nu = \mu$). This gives the following model:

$$\frac{dS}{dt} = \underbrace{B}_{birth} - \underbrace{\beta SI}_{infection} - \underbrace{\nu S}_{death} \tag{3.1}$$

$$\frac{dI}{dt} = \underbrace{\beta SI}_{infection} - \underbrace{\gamma I}_{recovery} - \underbrace{\nu I}_{death} \tag{3.2}$$

$$\frac{dR}{dt} = \underbrace{\gamma I}_{recovery} - \underbrace{\nu R}_{death} \tag{3.3}$$

The parameters have the following meaning:

- B - birth rate.
- β - per capita infection rate
- ν - death rate.
- γ - recovery rate.
- N - total population size.

and are subject to initial conditions $S(0) = S_0 > 0, I(0) = I_0 \geq 0$ and $R(0) = R_0 \geq 0$.

We wish to explore whether the demographic dynamics may allow a disease to die out or persist in a population in the long term. For this specific reason, we look at the stability of system at the *disease free equilibrium* (**DFE**) point. The condition is that the point $(S, I, R) = (N,0,0)$ satisfies the following equations:

$$\begin{cases} \frac{dS}{dt} & = 0 \\ \frac{dI}{dt} & = 0 \\ \frac{dR}{dt} & = 0 \end{cases} \tag{3.4}$$

It is also of interest of look at the existence and stability of an *Endemic Equilibrium* (**EE**), for which $I > 0$.

3.0.1 The Disease free equilibrium

With some algebraic manipulations, the Jacobian of system is given by [23]:

$$J = \begin{pmatrix} -\nu - \frac{\beta I}{N} & -\frac{\beta S}{N} & 0 \\ \frac{\beta I}{N} & \frac{\beta S}{N} - \gamma - \nu & 0 \\ 0 & \gamma & -\nu \end{pmatrix} \tag{3.5}$$

If we substitute (1,0,0) into the **DFE** we obtain:

$$J = \begin{pmatrix} -\nu & \beta & 0 \\ 0 & \beta - \gamma - \nu & 0 \\ 0 & \gamma & -\nu \end{pmatrix} \tag{3.6}$$

We can find the eigenvalues of this matrix:

$$\lambda_1 = -\nu \tag{3.7}$$

$$\lambda_2 = -\nu \tag{3.8}$$

$$\lambda_3 = \beta - \nu - \gamma \tag{3.9}$$

The stability analysis gives:

1. $\lambda_1 < 0$ an $\lambda_2 < 0$ are always negative.

2. λ_3 is determined by

$$\lambda_3 = (\gamma + \nu)(R_0 - 1) \tag{3.10}$$

   where $R_0 = \frac{\beta}{\gamma+\nu}$

- if $R_0 < 1$, then **DFE** is stable, so this is an epidemic *deceases* and implies $\lim_{x\to\infty} I(t) = 0$
- if $R_0 > 1$, then **DFE** is unstable, i.e. an epidemic occurs.

3.0.2 The endemic equilibrium

For the endemic equilibrium we have [27]:

$$\begin{aligned}
\lim_{x\to\infty} S(t) &= \frac{\gamma+\nu}{\beta} = \frac{1}{R_0} \tag{3.11}\\
\lim_{x\to\infty} I(t) &= \frac{\nu(N-S)}{\beta S} \\
&= \frac{\nu\cancel{S}(\frac{N}{S}-1)}{\beta\cancel{S}} \\
&= \frac{\nu}{\beta}\left(\frac{1}{S}-1\right) \\
&= \frac{\nu}{\beta}\big(R_0-1\big) \tag{3.12}\\
\lim_{x\to\infty} R(t) &= 1-\frac{1}{R_0}-\frac{\nu}{\beta}\big(R_0-1\big) \tag{3.13}
\end{aligned}$$

Proof:

We are only going to consider small deviation the equilibrium point. We write the system as a vector differential equation as follows:

$$\frac{d}{dt}\begin{pmatrix} S \\ I \end{pmatrix} = \begin{pmatrix} \nu-\beta SI-\nu S \\ \beta SI-\gamma I-\nu I \end{pmatrix} = \mathbf{f}(S,I)$$

By the Taylor's theorem:

$$\mathbf{f}(S,I) = \mathbf{f}(S_0,I_0) + \mathbf{J}(S_0,I_0)\left[\begin{pmatrix} S \\ I \end{pmatrix} - \begin{pmatrix} S_0 \\ I_0 \end{pmatrix}\right] + \ldots$$

At the equilibrium point, $\mathbf{f}(S_0, I_0) = 0$, therefore the dynamics close to (S_0, I_0) are to first order determined by the derivative of vector function **f**(S,I), i.e by the *Jacobian matrix*, $\mathbf{J}(S_0, I_0)$.

The Jacobian matrix of $\mathbf{f}(S_0, I_0)$ is:

$$\begin{aligned}\mathbf{f}(S,I) &= \begin{pmatrix} \frac{\delta f_1}{\delta S} & \frac{\delta f_1}{\delta I} \\ \frac{\delta f_2}{\delta S} & \frac{\delta f_2}{\delta I} \end{pmatrix} \\ &= \begin{pmatrix} -\beta I - \nu & \beta S \\ \beta I & \beta S - \gamma - \nu \end{pmatrix} \end{aligned} \tag{3.14}$$

1. **The equilibrium at** $(S, I) = (1, 0)$

   We substitute (S, I)=(1,0) for *disease-free equilibrium* point [28, 29]:

   $$\mathbf{J}(1,0) = \begin{pmatrix} \nu & \beta \\ 0 & \beta - \gamma - \nu \end{pmatrix} \tag{3.15}$$

   Hence, we have the eigenvalues

   $$\lambda_1 = -\nu \tag{3.16}$$

   $$\lambda_2 = \beta - \gamma - \nu \tag{3.17}$$

   where:

   (a) $\lambda_1 = -\nu < 0$ is always negatives.

   (b) $\lambda_2 = \beta - \gamma - \nu$ only if

   - $$\begin{aligned}\beta - \gamma - \nu &< 0 \\ \frac{\beta}{\gamma + \nu} &< 1\end{aligned} \tag{3.18}$$

   - $$\begin{aligned}\beta - \gamma - \nu &> 0 \\ \frac{\beta}{\gamma + \nu} &> 1\end{aligned} \tag{3.19}$$

This indicates the rôle of the *basic reproductive number*:

$$R_0 = \frac{\beta}{\gamma + \nu}$$

2. **The equilibrium at** $(S, I) = \left(\frac{\gamma+\nu}{\beta}, \frac{\nu(\beta-\gamma-\nu)}{\beta(\gamma+\nu)}\right)$

Again, to check the stability of the *endemic equilibrium* at [28, 29]

$$\mathbf{J}\left(\frac{\gamma+\nu}{\beta}, \frac{\nu(\beta-\gamma-\nu)}{\beta(\gamma+\nu)}\right) = \begin{pmatrix} \frac{\beta\nu}{\gamma+\nu} & -\gamma-\nu \\ \frac{\nu(\beta-\gamma-\nu)}{\gamma+\nu} & 0 \end{pmatrix}$$

After few algebraic steps, we have the eigenvalues:

$$\begin{aligned} \lambda_{1/2} &= -\frac{\beta\nu}{\gamma+\nu} \pm \sqrt{\frac{\beta^2\nu^2}{(\gamma+\nu)^2} - 4\nu(\beta-\gamma-\nu)} \\ &= -\underbrace{\nu R_0}_{a} \pm \underbrace{\sqrt{\nu^2 R_0^2 - 4\nu(\beta-\gamma-\nu)}}_{b} && (3.20) \\ &= -a \pm b && (3.21) \end{aligned}$$

We observe that

(a)

$$R_0 = \frac{\beta}{\gamma+\nu} > 1 \qquad (3.22)$$

if $a > 0$, $b < 0$ and $\nu^2 R_0^2 > 4\nu(\beta - \gamma - \nu)$, then the fixed point is *stable.*

(b)

$$R_0 = \frac{\beta}{\gamma+\nu} < 1 \qquad (3.23)$$

if $a > 0$, $b > 0$ and $\nu^2 R_0^2 < 4\nu(\beta - \gamma - \nu)$ implies that the fixed point is *unstable.*

To summerize, the system has the following equilibria of steady states:

$$E_1 = (S, I, R) \;=\; (1, 0, 0) \qquad (3.24)$$

$$\begin{aligned} E_2 &= (S,I,R) = \left(\frac{\gamma+\nu}{\beta}, \frac{\nu(\beta-\gamma-\nu)}{\beta(\gamma+\nu)}, \frac{\gamma(\beta-\gamma-\nu)}{\beta(\gamma+\nu)}\right) \\ &= \left(\frac{1}{R_0}, \frac{\nu}{\beta}(R_0-1), 1-\frac{1}{R_0}-\frac{\nu}{\beta}(R_0-1)\right) \end{aligned} \tag{3.25}$$

which has the threshold properties that are independent of initial values in the set

$$\big(S(0), I(0), R(0)\big) \in \big\{(S,I,R) \in [0,N]^3 : S \geq 0, I \geq 0, R \geq 0, S+I+R = N\big\}$$

where that $R_0 = \frac{\beta}{\gamma+\nu}$. One can show that

1. **DFE**:

$$R_0 < 1 \Longrightarrow \lim_{t\to+\infty} \big(S(0), I(0), R(0)\big) = \big(N, 0, 0\big)$$

2. **EE**:

$$R_0 > 1, I(0) > 0 \Longrightarrow \lim_{t\to+\infty} \big(S(t), I(t), R(t)\big) = \left(\frac{1}{R_0}, \frac{\nu}{\beta}(R_0-1), 1-\frac{1}{R_0}-\frac{\nu}{\beta}(R_0-1)\right)$$

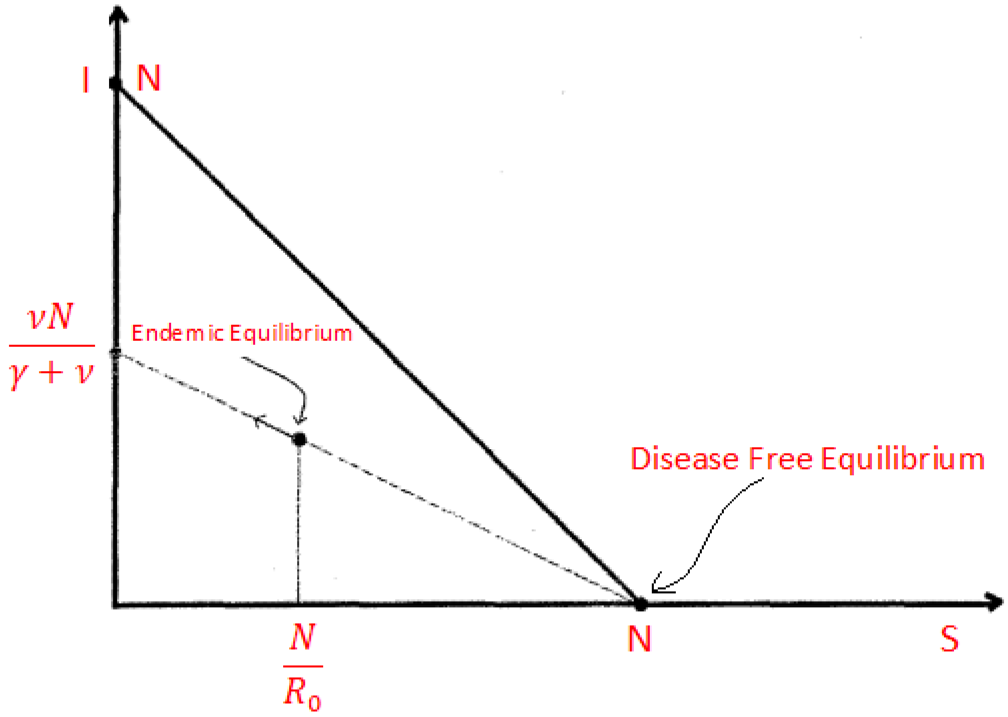

Figure 3.2: Steady state for an endemic equilibrium

With heuristic arguments, one may show that R_0 (see Figure 3.2) corresponds as the average number of infectious caused by a single infectious host subject in an wholly susceptible population. The above relationship means that if this number is $R_0 < 1$, then the disease get extinct. If $R_0 > 1$, then the disease will remain permanently endemic in the population.

3.1 The Epidemic Curve

We can also calculate the average rate of *recovery* from equation (2.3). By definition $S + I + R = 1$, so we can re-arrange the equation (2.3) as

$$\begin{aligned} \frac{d}{dt}R(t) &= \gamma\big(1 - \underbrace{S} - R\big) \\ &= \gamma\big(1 - S_0 e^{-R_0 R(t)} - R(t)\big) \end{aligned} \tag{3.26}$$

where S is substituted from equation (2.12). This equation does not have any explicit solution for R in terms of t. We can use Taylor expansion of $e^{-R_0 R(t)}$ according to the formula:

$$e^{-x} = 1 - x - \frac{1}{2}x^2 + \mathcal{O}(x^3) \tag{3.27}$$

and omit the last term as we assume that $R_0 R$ is small. This gives

$$e^{-R_0 R(t)} = 1 - R_0 R(t) + \frac{R_0 R^2(t)}{2} \tag{3.28}$$

Then substituting into equation (3.26) gives [30],

$$\begin{aligned} \frac{dR}{dt} &= \gamma\left(1 - R(t) - S_0\left[1 - R_0 R(t) + \frac{R_0^2 R^2(t)}{2}\right]\right) \\ &= \gamma\left(1 - R(t) - S_0 + S_0 R_0 R(t) - \frac{S_0 R_0^2 R^2(t)}{2}\right) \\ &= \gamma\left(1 - S(t) - S_0\Big[1 - R_0\Big]R(t) - \frac{S_0 R_0^2}{2}R^2(t)\right) \\ &= \gamma\left(1 - S(t) + S_0\Big[R_0 - 1\Big]R(t) - \frac{S_0 R_0^2}{2}R^2(t)\right) \end{aligned} \tag{3.29}$$

We obtain the solution as:

$$R(t) = \frac{1}{R_0^2 S_0}\left(S_0 R_0 - 1 + \alpha \tanh\left[\frac{1}{2}\alpha\gamma t - \phi\right]\right) \tag{3.30}$$

Letting

$$\alpha = \sqrt{\left(S_0 R_0 - 1\right)^2 + 2 S_0 I_0 R_0^2} \tag{3.31}$$

and

$$\phi = \tanh^{-1}\left[\frac{1}{\alpha}\left(S_0 R_0 - 1\right)\right] \tag{3.32}$$

To derive the epidemic curve as a function of time, we would differentiate equation (3.30) with respect to time [30]:

$$\text{reported cases} \sim \frac{dR}{dt} = \frac{\gamma\alpha^2}{2 S_0 R_0^2}\operatorname{sech}^2\left(\frac{\alpha\gamma}{2}t - \phi\right) \tag{3.33}$$

This is a classical epidemic curve of the disease that shown in Figure (3.3). Epidemiologist are highly interested in this curve because it is used to compare the forecasts of models with the data. For instance, the curve indicates that there is a greater force of infection at early stages.

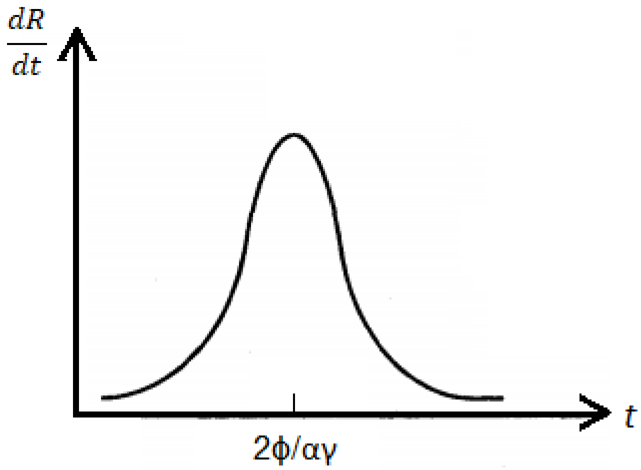

Figure 3.3: Epidemic curve

In the limit $t \to \infty$, everyone in the population becomes infected. Hence we can approximate

$$R_\infty \approx \frac{1}{R_0^2 S_0}\big(R_0 S_0 - 1 + \alpha\big) \tag{3.34}$$

or

$$R_\infty \approx \frac{\rho^2}{S_0}\left(\frac{S_0}{\rho} - 1 + \alpha\right) \tag{3.35}$$

Taking into consideration (3.31) and $N - S_0 = I_0 \approx 0$ as the initial number of infectious [30]:

$$\begin{aligned}
R_\infty &\approx \frac{\rho^2}{S_0}\left(\frac{S_0}{\rho} - 1 + \alpha\right) \\
&= \rho \; - \; \frac{\rho^2}{S_0} \; + \; \frac{\rho^2}{S_0}\alpha \\
&= \rho \; - \; \frac{\rho^2}{S_0} \; + \; \frac{\rho^2}{S_0}\left[\left(S_0 R_0 - 1\right)^2 + 2 S_0 I_0 R_0^2\right]^{\frac{1}{2}} \\
&= \rho \; - \; \frac{\rho^2}{S_0} \; + \; \frac{\rho^2}{S_0}\left(\frac{S_0}{\rho} - 1\right) \; + \; \frac{\rho^2}{S_0}\left(\frac{2 S_0 I_0}{\rho^2}\right)^{\frac{1}{2}} \\
&= \rho \; - \; \frac{\rho^2}{S_0} \; + \; \rho - \; \frac{\rho^2}{S_0} \; + \; \left(\frac{2\rho}{S_0}\right)^{\frac{1}{2}} \underbrace{I_0}_{\sim\, 0} \\
&= 2\rho\left(1 - \frac{\rho}{S_0}\right)
\end{aligned} \tag{3.36}$$

3.1.1 SIR Model (with birth and death rates)

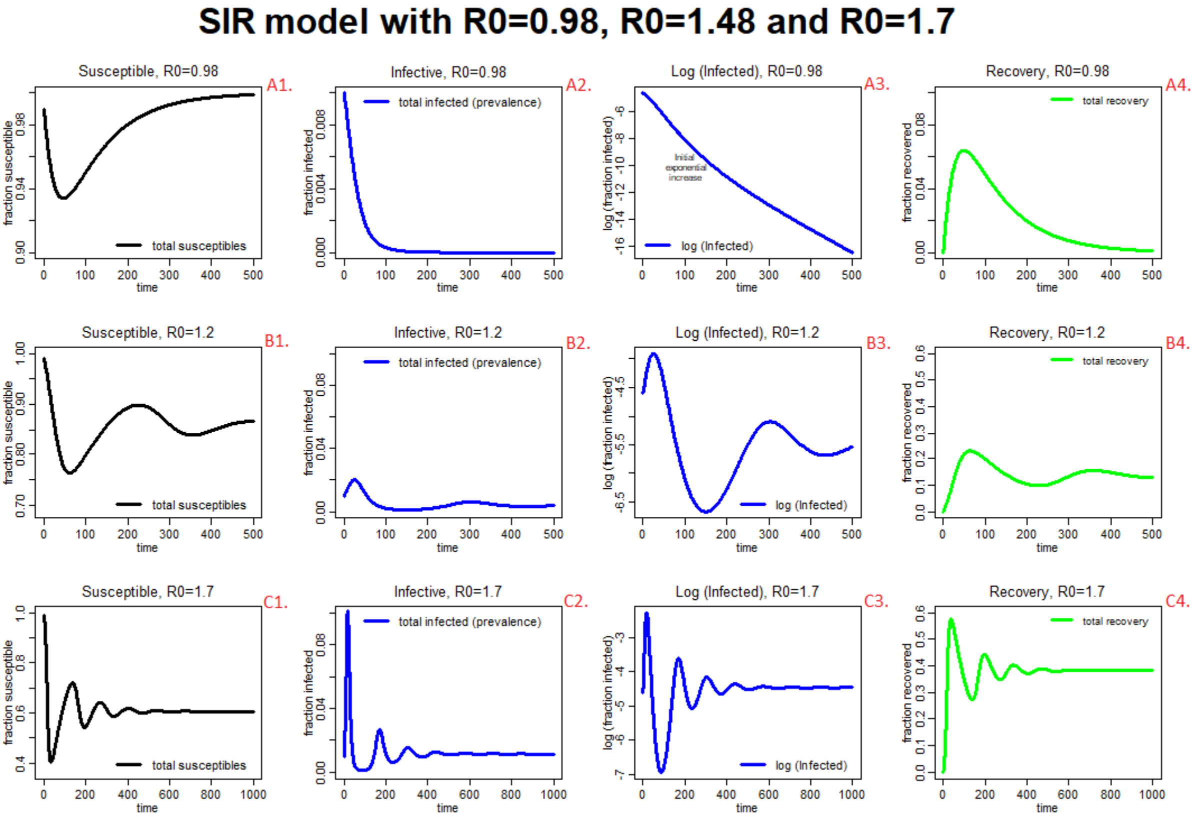

Figure 3.4: SIR model with vital dynamics. Population $N = 1000$ and $I_0 = 10$, birth and death rate are constants ($\gamma = \nu = 0.01, R_0 = 0.98, 1.48$ and $R_0 = 1.70$)

As expected, enabling vital dynamics can sustain an epidemic or allow new introductions to spread because new births provide more susceptible individuals. We can see (Figure 3.4, B2 and C2) that there are strong peaks in the epidemic as the *basic reproductive rate* is increased (if $R_0 > 1$). The peaks are followed by decaying oscillations to the final endemic stage. Any small perturbation will give rise to a *damped oscillation*, with frequency and damping rate determined by the set of parameters. On the other hand, the plots in A1 to A4, show that as $R_0 < 1$ decreases, the infection dies out rapidly ($R_0 = 0.98, \gamma = 0.33, \mu = \nu = 0.01$).

3.2 Seasonal Forcing

Many infectious diseases fluctuate over time, and they frequently show seasonal patterns in their incidence. The cause of seasonal fluctuations may be periodic contact rates, periodic patterns in temperature climate, and periodic patterns in vaccination programs. Hence, the transmission parameter $\beta = \beta(t)$ can be though of as periodically varying. With time-varying parameters we generally have to rely on computer simulations. Most often one uses sinusoidal paramters in the ODE system (2.1-2.3), with a period of 1 year.

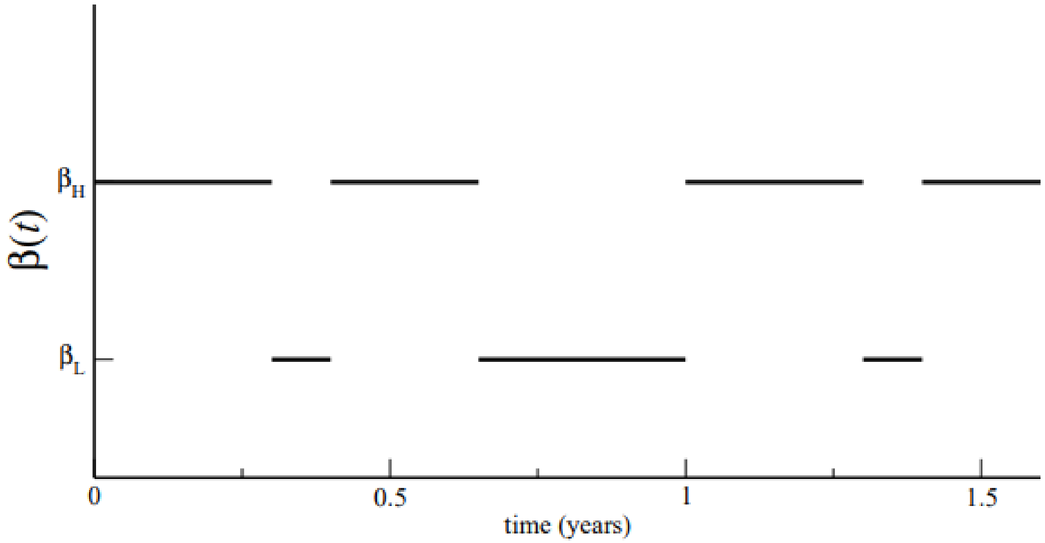

Figure 3.5: Seasonal variations in transmission: discrete level representation, modeling a climate effect between summer and winter. And a yearly vacation effect. [31]

As so, it can be modelled as having a seasonal behaviour of increased infection due the fact that the transmission rate has annual periodicity as shown in figure (3.5). One way of modeling seasonality could be [32]

$$\beta(t) = \beta_0 \left[1 + \beta_1 \cos\left(\frac{2\pi(t-\phi)}{365} \right) \right] \tag{3.37}$$

where β_0 is the baseline transmission rate, β_1 is the relative seasonal forcing (is the amplitude of the seasonal variation strength) and ϕ corresponds for the time (measured in days) of the year when transmission rate is maximal.

3.2.1 Seasonal transmission and seasonality in birth

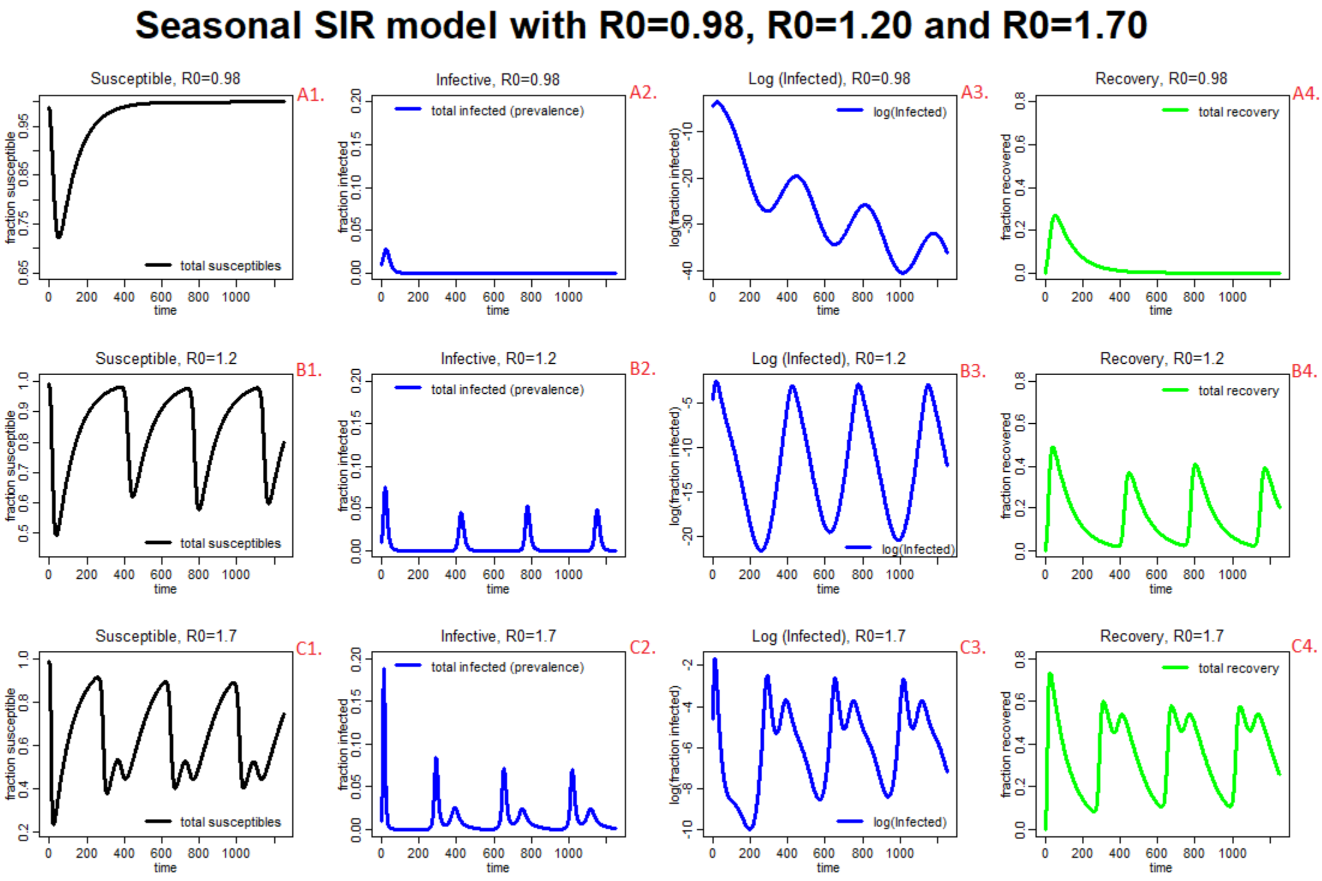

Figure 3.6: We model the effect of birth seasonality with a simple cosine function so that mean duration of the birth and death rates, as well as infectious period are constant, but R_0 varies; ($B = \gamma = \nu = 0.01, \beta_1 = 0.3, R_0 = 0.98, 1.2$ and 1.70)

We can also extend our analysis from the previous SIR model with an embedded seasonal forcing from equation (3.37). Here we only focus on the effect of changes in the transmission amplitude (β_0), which will have the greatest impact and induce resonance and lead to complex dynamics. We use a total population of $N = 1000$, and an initial value of $I_0 = 10$. The transmission rate β_0 can be obtained from equation (2.4) on page 20. Equation (3.37) gives

$$B_0 \propto R_0\gamma \tag{3.38}$$

Seasonality in either transmission rate, birth rate or death rate can yield complex dynamics in the SIR model. See Figure (3.6). Figures (A1-A4) show dynamics for $R_0 = 0.98$. Compare these plots to (B1-B4), where we observe that increasing *basic reproductive ratio* increases the contact rate. The trend (blue) provides a strong seasonal cycle in R_0 by affecting host recovery or per-contact transmission probability.

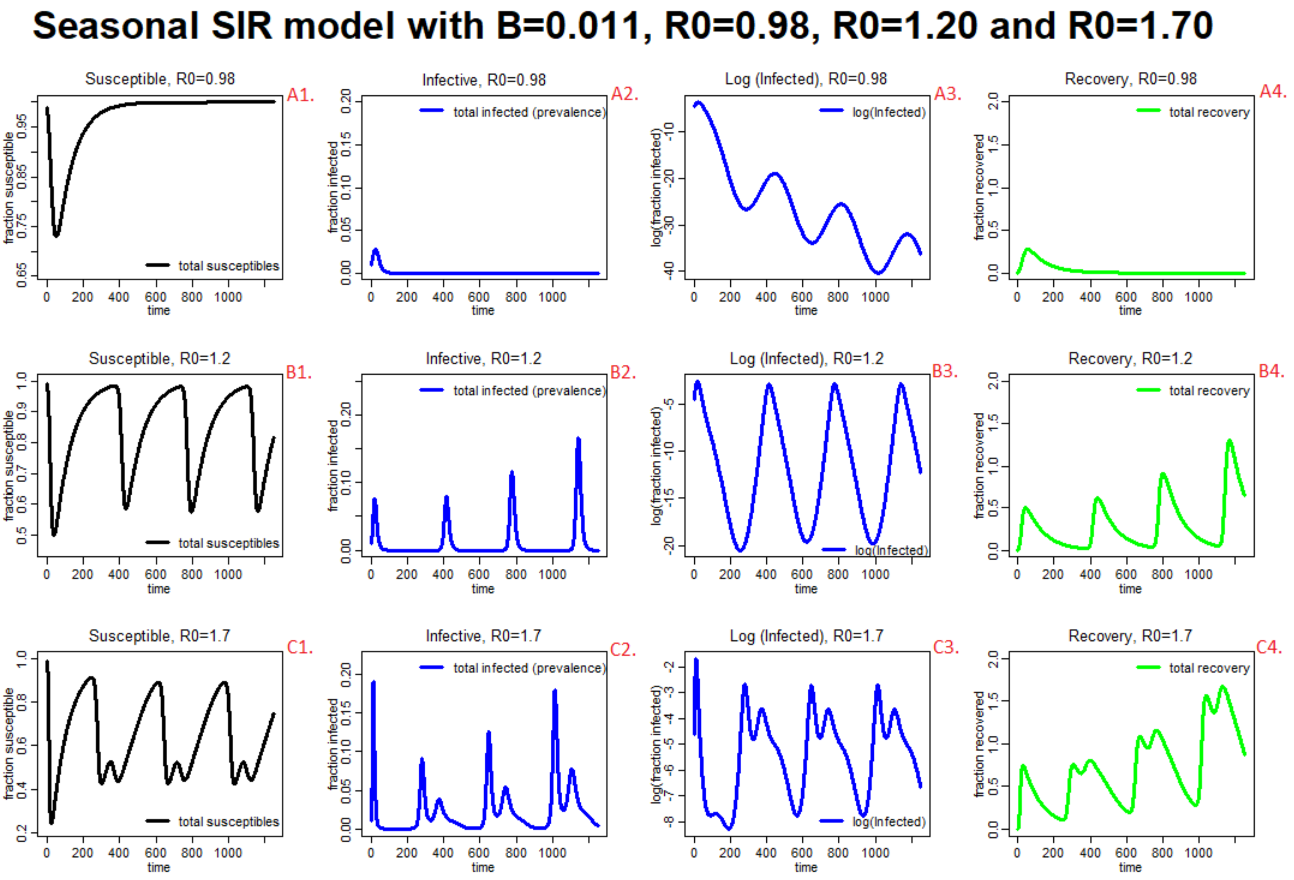

Figure 3.7: We model the effect of birth seasonality with a simple cosine function. ($\gamma = \nu = 0.01, B = 0.011, \beta_1 = 0.3, R_0 = 0.98, 1.2$ and $R_0 = 1.70$)

As demonstrated in Figure (3.7, B2), we can have multi-daily oscillations when transmission rates vary seasonally. The intuition behind this is that the transmission rate consists of both the contact rate and transmission probability. Increasing the birth rate is likely to increase the contact rate, which also induce a resonance in amplitude.

At a higher *basic reproductive ratio*, $R_0 = 1.70$, with birth rate constant, we obtain results similar to those shown in previous plots C2 and C4. As we see that increasing birth amplitude exacerbates the tendency for chaotic dynamics and also shifts the timing of the epidemic peaks more varied and shorter cyclic duration. The general pattern remains similar, however the number of infected host can periodically hike above 1 for more long term occur, even for this rather smaller system, N. Eventually this causes an irregular increase chaotically therefore resulting an *undamped oscillations*, for instance, the amplitude corresponds the number of infected people increase with time ($t \to \infty$) and since it is $R_0 > 1$. To be specified, the response C2 showed increasing *frequency* and increasing undamped ratio as the amplitude increased.

Chapter 4

Extension to stochastic SIR

The objective in this chapter is to explore the dynamical long term behaviour of a stochastic SIR model. We establish the a stochastic SIR model, which is conceptually more complex than the deterministic one. However it is not much more difficult to simulate numerically, but a large number of realizations is needed to determine the expected behaviour.

$$\frac{d}{dt}S(t) = -\beta\frac{S(t)I(t)}{N}, \qquad \frac{d}{dt}I(t) = \beta\frac{S(t)I(t)}{N} \; - \; \gamma I(t), \qquad \frac{d}{dt}R(t) = \gamma I(t) \qquad (4.1)$$

4.1 From Deterministic to Stochastic Models

There are few different stochastic modelling frameworks. For instance, the *discrete time Markov Chain* (DTMC) model, the *continuous time Markov Chain* (CTMC) model, and the class of stochastic differential equation (SDE) models. These stochastic processes differ in their underlying assumptions in terms of the time and the state variables. In a DTMC model, the time and state variables are discrete. In a CTMC model, time is continuous whereas the state variable is discrete. Lastly, the SDE models are based on diffusion process where both the time at time scales, $t \in [0, \infty)$ and state variables; $S(t), I(t), R(t) \in (0, 1, 2, \ldots, N)$, are continuous.

4.2 Markov Chain methods

In this section, we review basic Markov Chain Monte Carlo (MCMC) methods [33].

4.2.1 Basic notions

One hundred years ago, the Russian mathematician, named Andreyevich Markov developed the theory of Markov chains [34]. Random samples are drawn sequentially, with the distribution of the sampled draws depending on the last value drawn:

$$\Pr(X_t = j_t \mid X_{t-1} = i_{t-1}, X_{t-2} = i_{t-2}, \dots) = \Pr(X_t = j \mid X_{t-1} = i) \tag{4.2}$$

In more general terms, let $p_{i,j}$ be probability of a transition from state i to state j. In n time steps

$$p_{i,j}^{(n)} = Pr(X_t = n \mid X_0 = i_0) \tag{4.3}$$

Note that $p_{i,j}$ is the probability of single-step transition:

$$p_{i,j} = Pr(X_t \mid X_t = i_{t-i}) \tag{4.4}$$

The *transition probability matrix* can be define equation (4.3) as $\Pr = [p_{i,j}]$ and $\sum_j p_{i,j} = 1$ for the all state space are discrete at index, i.

4.2.2 Transmission probabilities of state

For our SIR-type models we can define events. For example, an infection event decreases the number of susceptible by one, while increasing the number of infected by one.

Event	Transition	Rate at which event occurs	Probability of transition in time interval $[t,\ t{+}dt]$
Infection	$S \to S-1,\ I \to I+1$	$\beta SI/N$	$(\beta SI/N)\ dt$
Recovery	$I \to I-1,\ R \to R+1$	γI	$\gamma I\ dt$

Table 4.1: Possible events in a *standard stochastic* SIR model, rates and probabilities of occurrence at a time interval

In the time interval $[t, t + dt]$ shown in Table (5.1), the probability of an infection with β is the simultaneous transitions $S \to S - 1$ to $I \to I + 1$ is $\beta \frac{SI}{N}\ dt + o(dt)$ where susceptible individuals moving from class S to I. Change in the state transitions $I \to I - 1$ to $R \to R + 1$ has probability $\gamma I\ dt + o(dt)$. We have:

$$\Pr((S_{t+dt}, I_{t+dt}, R_{t+dt}) = j | (S_t, I_t, R_t) = i) = (-1, 1, 0)) = \beta \frac{S_t I_t}{N}\ dt + o(dt) \tag{4.5}$$

$$\Pr((S_{t+dt}, I_{t+dt}, R_{t+dt}) = j | (S_t, I_t, R_t) = i) = (0, -1, 1)) = \gamma I_t\ dt + o(dt) \tag{4.6}$$

with the complementary probability:

$$\Pr((S_{t+dt}, I_{t+dt}, R_{t+dt}) = j | (S_t, I_t, R_t) = i) = (0, 0, 0)) = 1 - \left(\beta \frac{S_t I_t}{N} + \gamma I_t \right) dt + o(dt)$$

4.3 Simulation of a Monte Carlo steps

A pseudo-code for simulating a Monte Carlo method of SIR model has following steps:

1. Setup the model parameters, and some initial condition at time, $t = 0$ in all compartments.

2. Determine all possible changes of +1 or -1 that can occur in the number of individuals in the compartments.

3. Based on the current state of the system, determine the time step, dt needed for just one individual to change compartments in the entire system, **averagely**

4. Determine the average number of times, based on the current state of the system, that each of the possible transitions will take place in time dt.

5. Sample Poisson distributed random numbers based on these probabilities.

6. Repeat steps 2 to 5 for as many time steps as desired, or some conditions are reached (for example, no transitions are possible due to the state of the system).

4.3.1 Time step implementing for dt

Notice that the flow out of the Susceptible (S) compartment at time (t) is $\beta SI/N$ in SIR system (2.1). The flow out of the Infected (I) compartment (2.2) is γI, and there is no flow out of the Recovered(R) compartment (2.3). The units of $\beta SI/N$ and γI are based on the chosen unit time. A good estimate of the optimal time step in the model is

$$dt = \frac{1}{\beta SI/N \ + \ \gamma I} \tag{4.7}$$

4.4 Results

4.4.1 Single Chain with N=1000

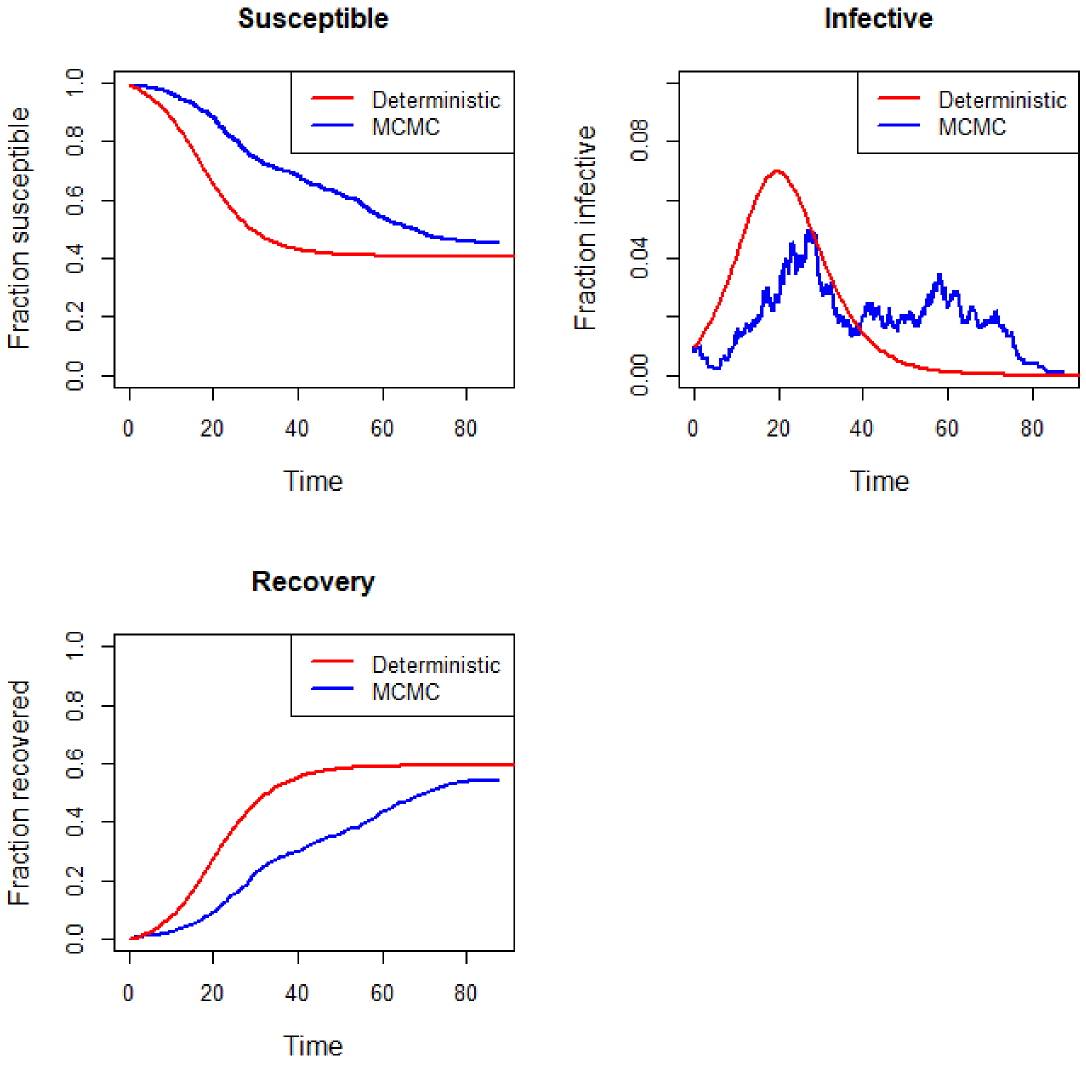

Figure 4.1: Trajectories of SIR model with random effects, curves are (S), (I), Recovery (R) verses time, corresponds to $R_0 = 1.48$, $\gamma = 0.3$ and $I_0 = 10$ with initial population $N = 1000$ for 1 iteration.

From the plots shown above, the single simulation method indicate that random walking effects are not recommended at all, due to confidence valid. Using the MCMC algorithms starting with a singles guess and generates a single chain of samples from that guess which not given enough to burn-in and converge to the target distribution. In this case just one realization is useless to infer anything from the stochastic model.

4.4.2 Single Chain with N=10,000

The following plots illustrate (4.2).

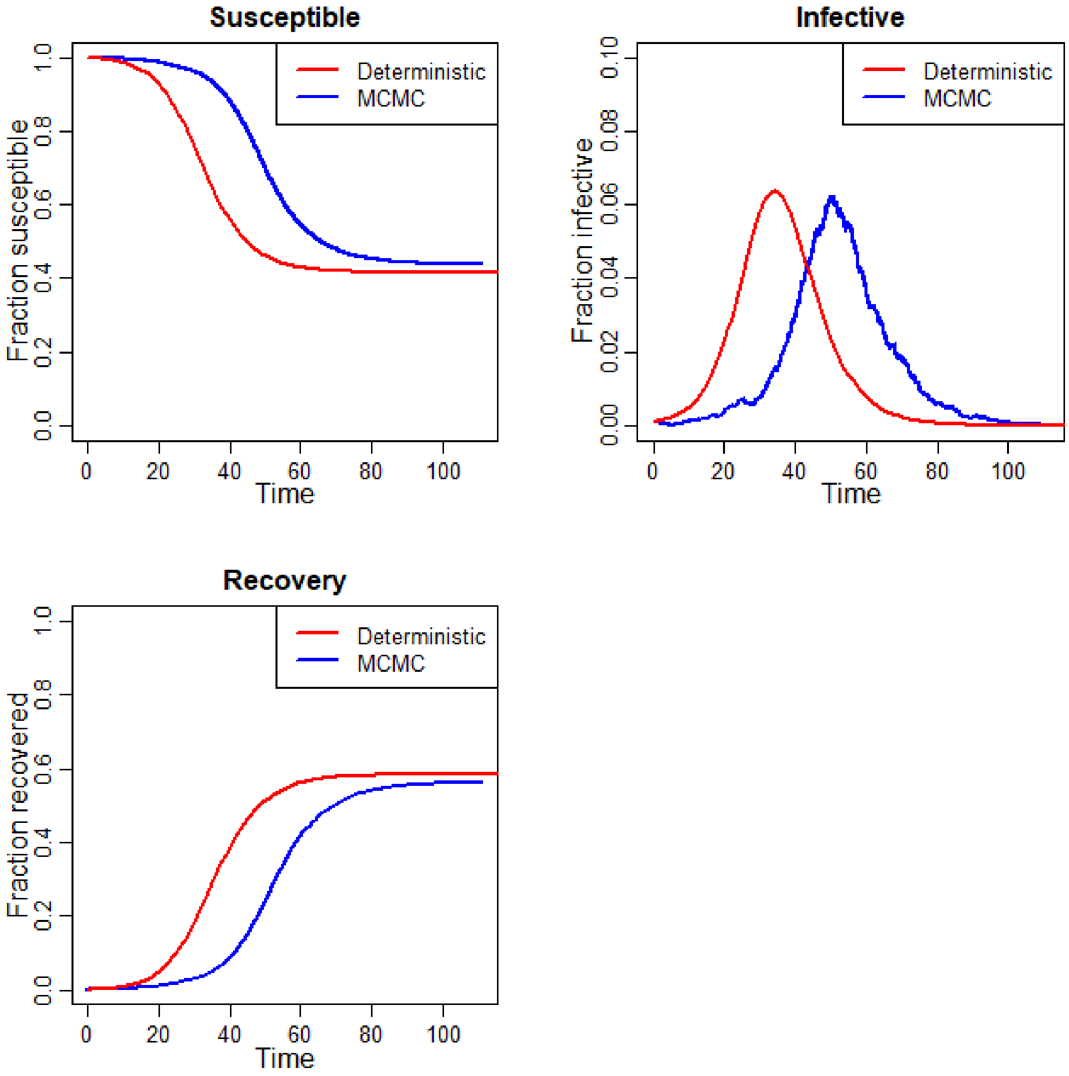

Figure 4.2: Trajectories of SIR model with random effects. Parameters are $R_0 = 1.48$ and $\gamma = 0.3$ and $I_0 = 10$ with initial population $N = 10000$ for 1 iteration.

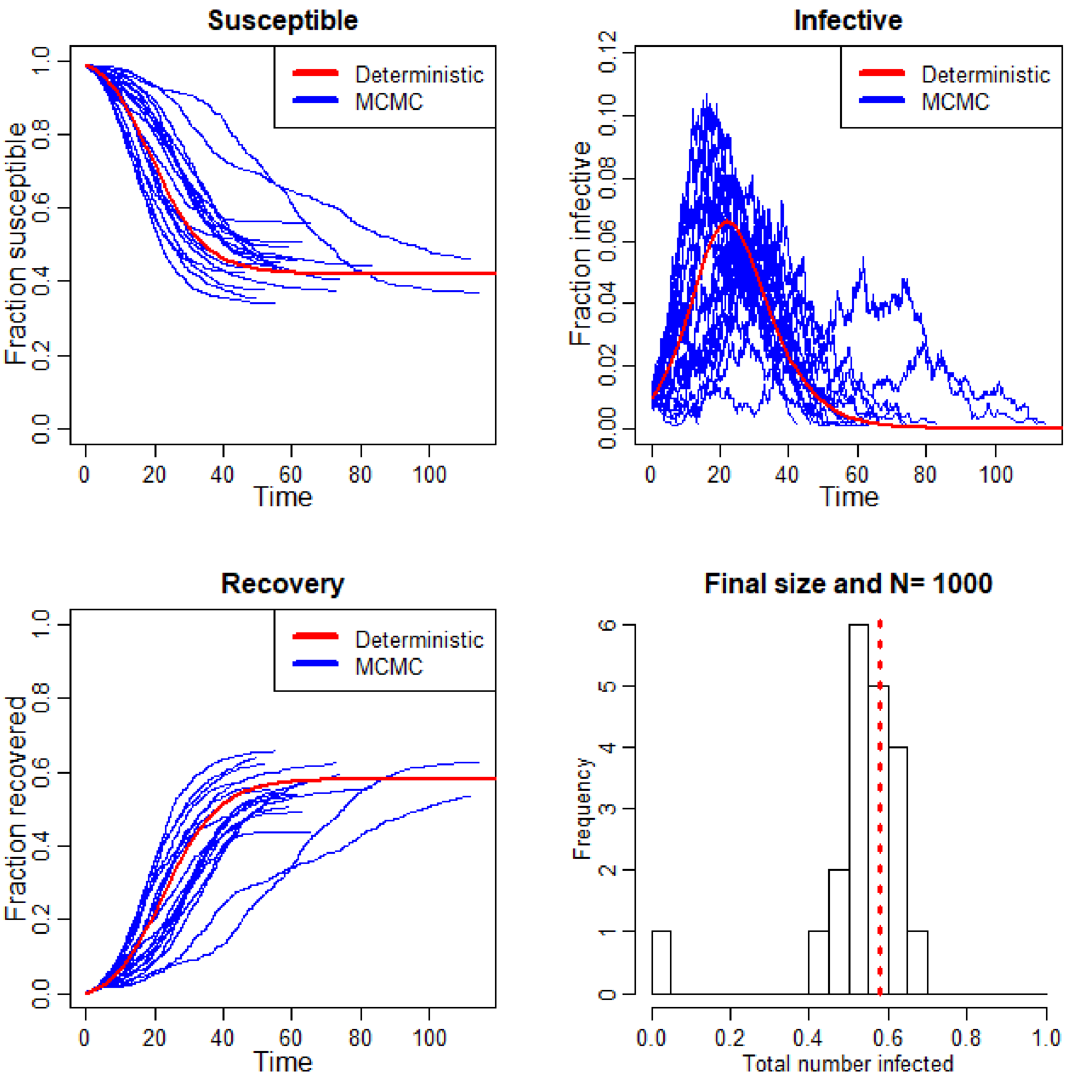

Figure 4.3: Trajectories of SIR model with random effects. Paramters are $R_0 = 1.48$, $\gamma = 0.3$ and initial population $N = 1000$ for 20 iterations.

4.4.3 Multiple iterations, 20 with sample size N=10,000

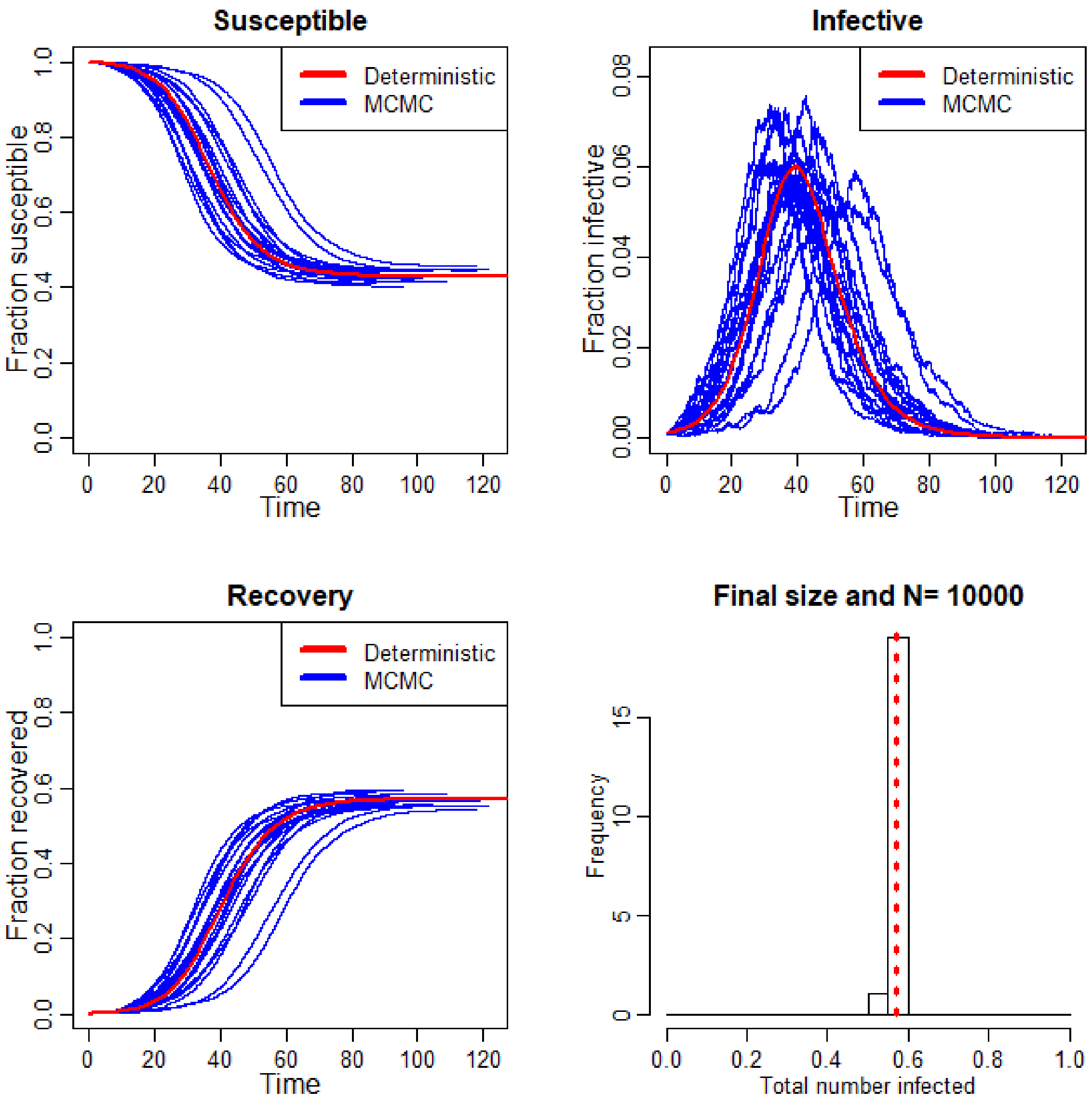

Figure 4.4: Trajectories of SIR model with random effects. Paramters are $R_0 = 1.48$, $\gamma = 0.3$ and initial population $N = 10,000$ for 20 iterations.

The plots (4.4) shows a big number of iterations. We note that 20 samples are required to obtain convergence to a desired tolerance level. As expected, the effect of sample size increased. The Markov Chain method has captured the essence of the true population by using deterministic and stochastic process as a benchmark for performance measurement [35]. The last plot shows that the final size of total number of infected represents proportion of peak incidence occur in the population.

4.4.4 Comparisons: Using MCMC for parameter estimation

In this section we use prevalence counts to compare our recursion method with the Metropolis-Hastings algorithm implemented in the R function `MCMCmetrop1R` (MCMCpack package) https://github.com/cran/MCMCpack, and birth-death method in the R function `dbd_prob` (MultiBD package) https://github.com/msuchard/MultiBD. This Metropolis-Hastings helps to solve to a real problem where the application consider uncertainty in measurements and uncertainty in model parameters to perform inverse problem from the Bayesian inference approach.

The SIR epidemic model is a bi-variate process because there are two independent random variables, $S(t)$ and $I(t)$. To write down ordinary differential equations (ODE) for the flow counting processes. These are [36],

$$\frac{dN_{SI}}{dt} = \beta_{SI}(t)S(t) \qquad \text{and} \qquad \frac{dN_{IR}}{dt} = \gamma_{IR}(t)I(t) \tag{4.8}$$

We see that $\{S(t), I(t)\}$ [36] consider as a birth-death process without loss of generality $S(t) + I(t) + R(t) = N$ and propose an algorithm to trace all possible transitions of $S(t)$ and $I(t)$ during a small time $(t, t+dt)$ occur with probabilities (4.5) and (4.6). Denote [37]

$$P_m = \Pr\left\{ \begin{array}{c|c} S(t_{m+1}) = s_{m+1} & S(t_m) = s_m \\ I(t_{m+1}) = i_{m+1} & I(t_m) = i_m \end{array} \right\} \tag{4.9}$$

Bi-variate process with n observations $\{(s_m, i_m)\}_{m=1}^{n}$ at time $\{t_m\}_{m=1}^{n}$, the log of the likelihood function [37] can be written as

$$\log\, l(\beta, \gamma | (s_m, i_m)_{m=1}^{n}) = \sum_{m=1}^{n-1} \log\, P_m \tag{4.10}$$

To satisfy positive constraints, we opt to use $\log \beta$ and $\log \gamma$ as our model parameters, since β and γ are non-negative. We assume *a priori* that $\log \beta \sim (\mu = 0, \sigma = 100)$ and $\log \gamma \sim (\mu = 0, \sigma = 100)$.

In this model, let β is the unknown infection rate of the disease and γ is the unknown recovery rate of infective people. We propose $\hat{\beta} = \beta$ and $\hat{\gamma} = \gamma$ to the unknown parameters and explore the posterior distribution of $(\log \beta, \log \gamma)$ by using a random-walk Metropolis algorithm implemented in the R function `MCMCmetrop1R` from package `MCMCpack` [38]. To be desired realizations of the model parameters in a stochastic SIR model, we have considered $N = 100$ and $N = 1000$ individuals from 0 to T (30 days). To initialize this process for evaluation of epidemic growth over time, we choose initial values of transitions rates are $\hat{\beta} = 0.4$ and $\hat{\gamma} = 0.35$ substitute into $\big((log(0.4), log(0.35)\big)$. We discard the first 200 iterations and summarize the posterior distribution of $(\hat{\beta}, \hat{\gamma})$ using the remaining iterations.

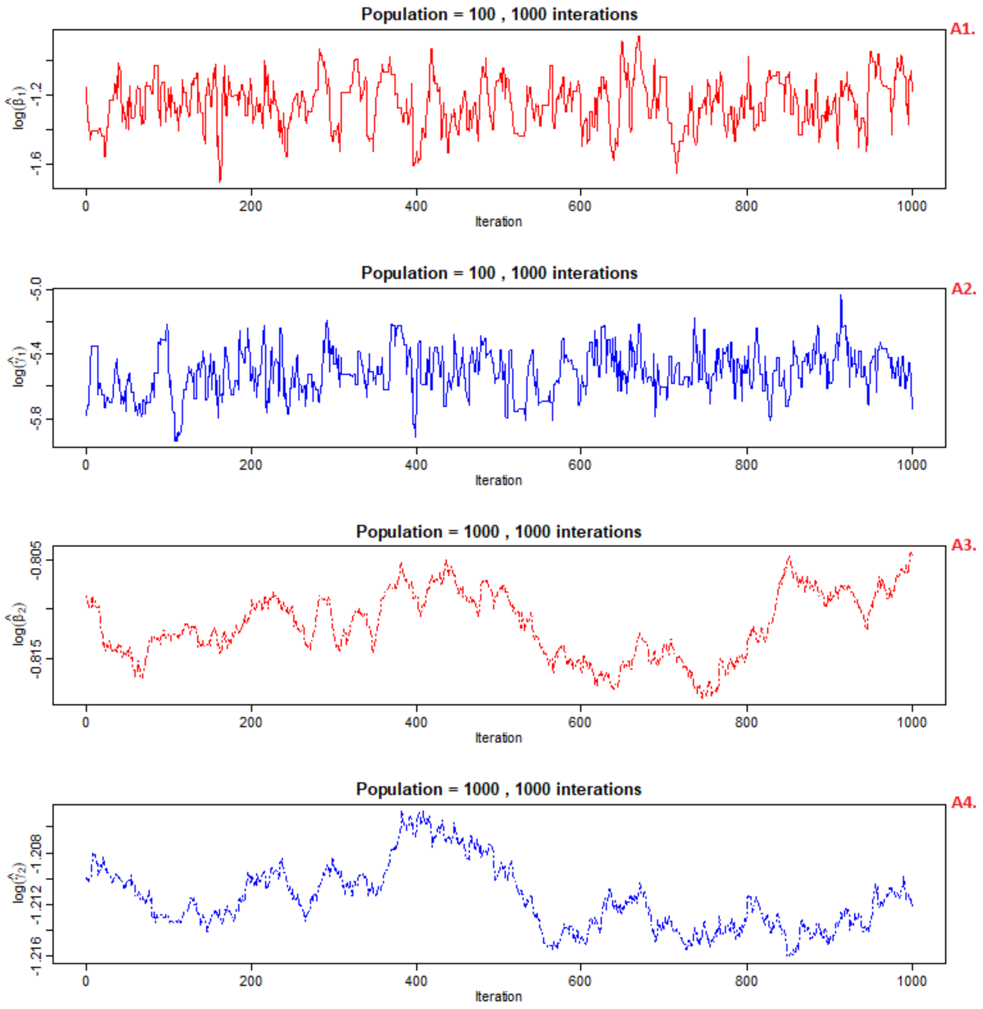

Figure 4.5: Trace plots summarize the posterior distribution for output (A1.) realizations model parameter which corresponds $\hat{\beta}_1$, (A2.) as for $\hat{\gamma}_1$, (A3.) as for $\hat{\beta}_2$, and (A4.) as for $\hat{\gamma}_2$. The first and second figures explain a good mixing where as the third and forth show bad mixing behavior.

Since convergence of the chain will occur regardless of the starting point, it is recommended to pick any feasible starting point [39]. The time and chain will take time to converge as vary depending on the starting point. To mitigate the effect of the starting distribution, which in our case we would discard 200 number of the first draws. For instance, let say t then run the Markov chain from n steps discarding away all the data without output [40]. After we specify the total number of iterations, 1000 shown for the Figure 4.5. The output was recorded to constitute samples from the density posterior distribution and the convergence can be visually assessed through trace plots. Trace plots provide an useful method for detecting problems with

Metropolis-Hastings convergence and mixing. We can notice that our chain gets lag in some areas of the parameter space at Figure 4.5 (A3) and (A4), which indicate bad mixing, indicates a high dependence between successive iterations, which implies a slow mixing or convergence rate [41]. Whereas Figure 4.5 (A1) and (A2) express first two plots are a good mixing of chain.

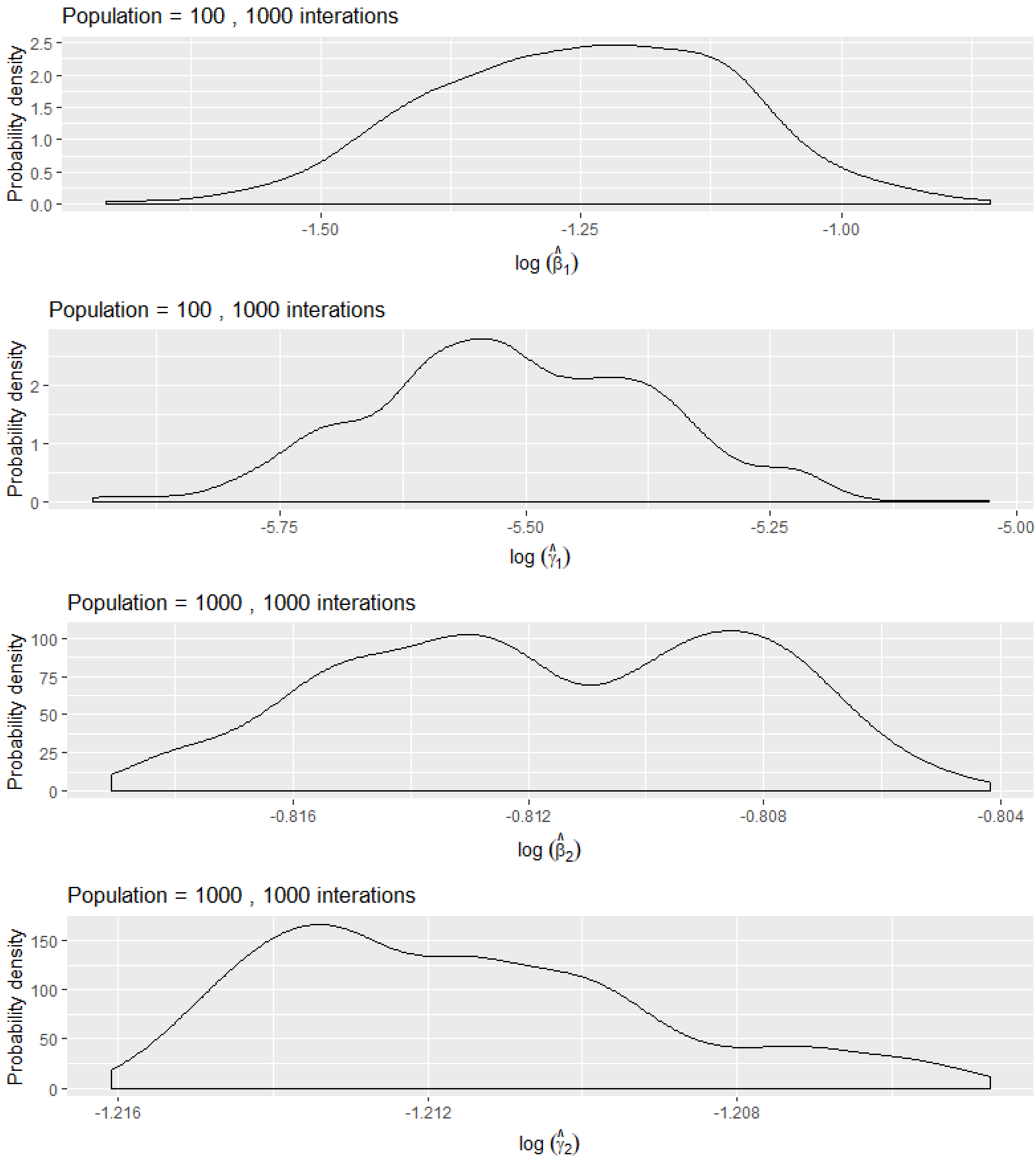

Figure 4.6: Presented above are the first and second posterior density plots have good mixing, whereas the third and fourth show bad mixing behaviour of the histogram.

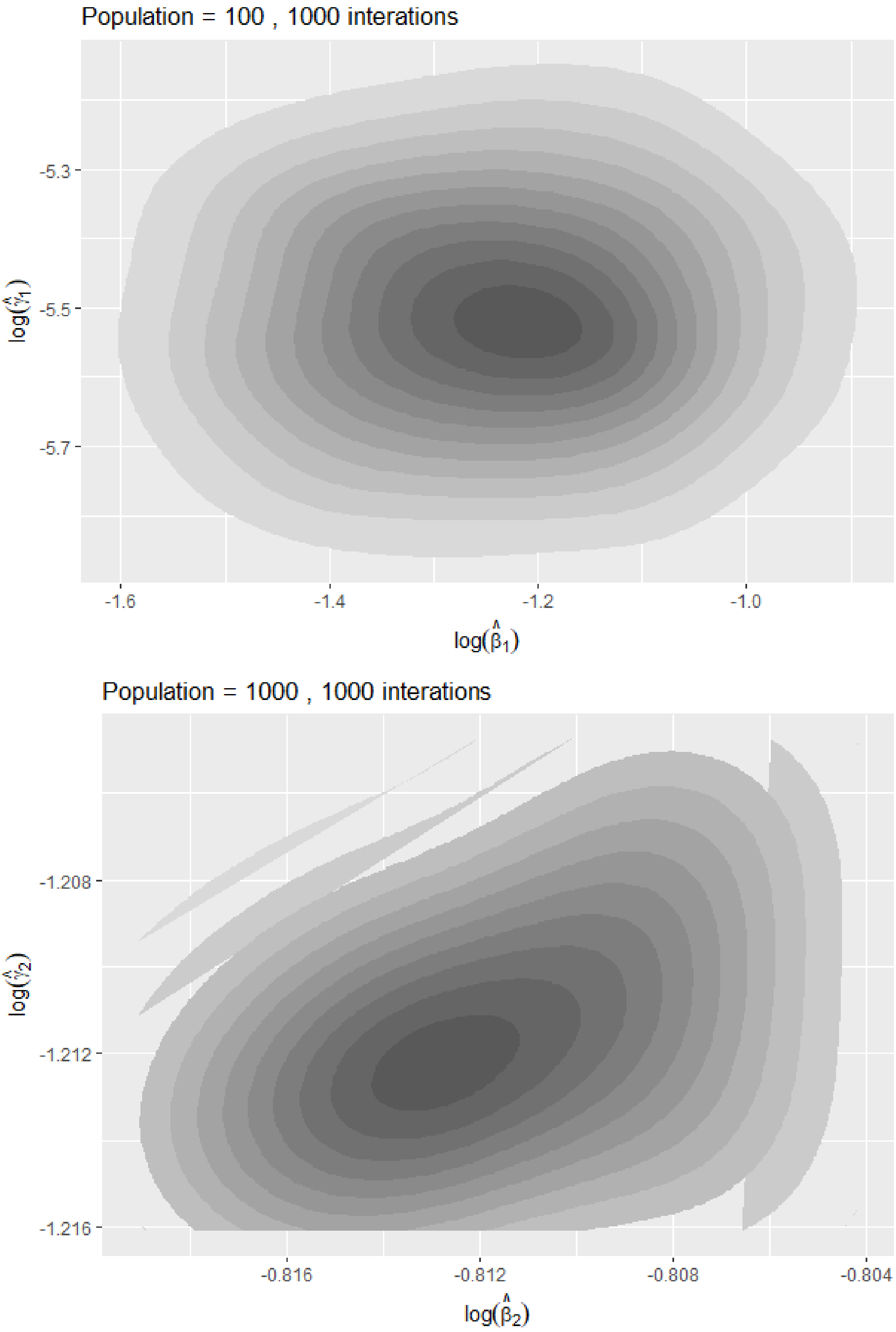

Figure 4.7: Posterior distributions (log scale) of the infection rate $\hat{\beta}_{1/2}$ and the recovery $\hat{\gamma}_{1/2}$ estimated over 1000 iterations for SIR model.

Parameter	Population: N = 100		Population: N = 1000	
	1000 iterations		1000 iterations	
	$\hat{\beta}_1$	$\hat{\gamma}_1$	$\hat{\beta}_2$	$\hat{\gamma}_2$
True value	0.444	0.299	0.444	0.299
Mean	0.427	0.321	0.470	0.312
Median	0.453	0.291	0.435	0.291
Standard deviation	0.039	0.011	0.031	0.017
Bayesian 95% C.I.	(0.285, 0.643)	(0.216, 0.394)	(0.365, 0.555)	(0.264, 0.343)
M-H acceptance rate	0.566		0.9042	

Table 4.2: Posterior parameter summaries from MCMC algorithm with initial parameter $\hat{\beta} = 0.4$ and $\hat{\gamma} = 0.35$ with population of $N = 100$ and $N = 1000$ respectively in fixed 1000 iterations.

Parameter	Population: N = 100		Population: N = 100	
	100 iterations		2000 iterations	
	$\hat{\beta}_1$	$\hat{\gamma}_1$	$\hat{\beta}_2$	$\hat{\gamma}_2$
True value	0.444	0.299	0.444	0.299
Mean	0.423	0.314	0.458	0.286
Median	0.456	0.281	0.423	0.294
Standard deviation	0.046	0.032	0.037	0.029
Bayesian 95% C.I.	(0.312, 0.594)	(0.243, 0.341)	(0.371, 0.499)	(0.272, 0.321)
M-H acceptance rate	0.513		0.551	

Table 4.3: Posterior parameter summaries from MCMC algorithm with initial parameter $\hat{\beta} = 0.4$ and $\hat{\gamma} = 0.35$ with iteration of 100 and 2000 respectively in fixed number of population.

The results from Table 4.2 shows that, the initial parameters of $\beta = 0.4$ and $\gamma = 0.35$ are used to carry out the stochastic model which are fall within Bayesian credible interval for $\hat{\beta}_1 = (0.285,\ 0.643)$, $\hat{\beta}_2 = (0.365,\ 0.555)$, $\hat{\gamma}_1 = (0.216,\ 0.394)$

and $\hat{\gamma}_2 = (0.264,\ 0.343)$ with mean estimates of $(\hat{\beta}_1,\ \hat{\beta}_2) = (0.427,\ 0.321)$ and $(\hat{\gamma}_1,\ \hat{\gamma}_2) = (0.470,\ 0.312)$ from Metropolis-Hasting simulation. However, $\hat{\beta}_1$ and $\hat{\gamma}_1$ show the random data-sets also lie with the confidence interval with only 50% acceptance rate, it tells the proposal function is too wide compared to the target distribution we sample from. For the number populations, $N = 100$ in Table 4.3, the Bayesian of 95% credible interval has slightly narrower between $\hat{\beta}_2 = (0.371,\ 0.499)$ $\hat{\gamma}_2 = (0.272,\ 0.321)$ and coverage for 2000 iterations compared to previous 1000 iterations with 1000 population. Having said that the basic reproduction number R_0 is also an important quantity in the SIR model which directly influence the analysis of transmission disease between compartments based on the formula, $R_0 = \beta/\gamma = 0.4/0.35 < 1$. Hence, it is not only parameter driving the dynamic of the epidemic.

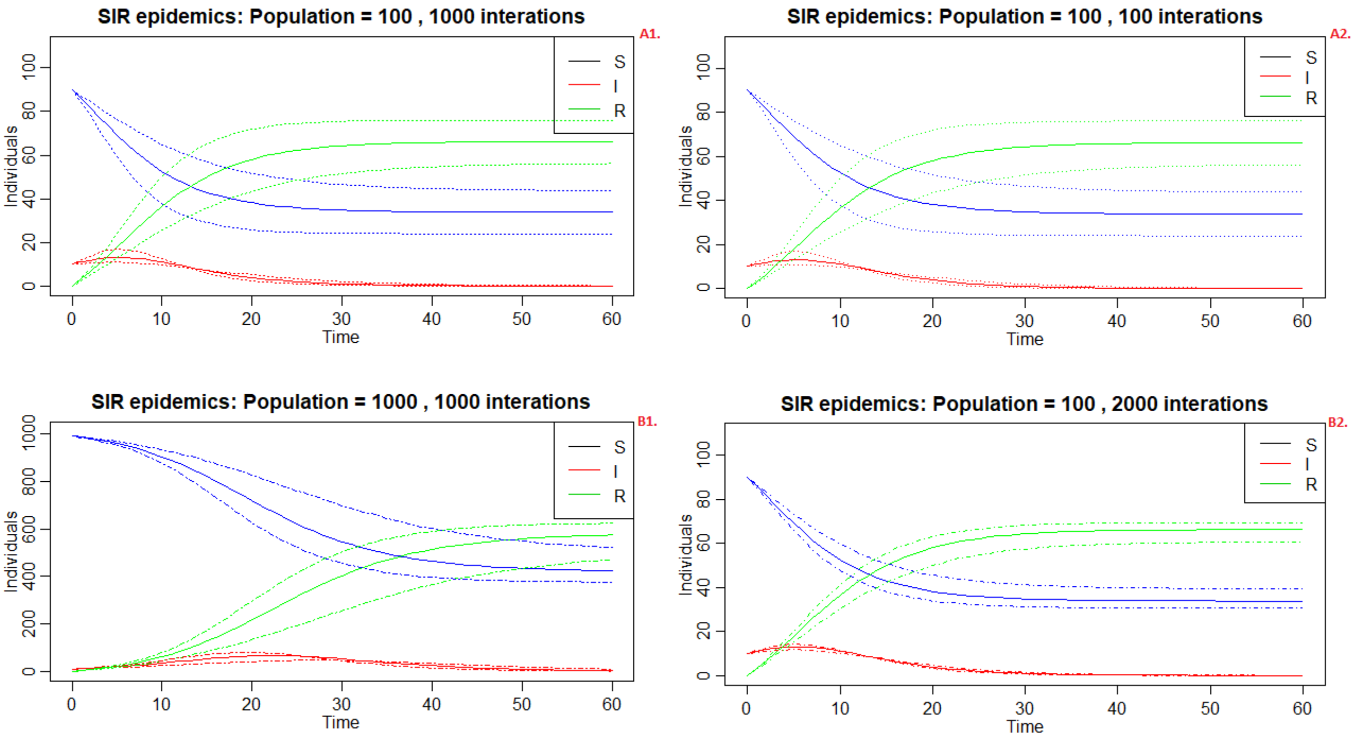

Figure 4.8: SIR epidemic plots for number of Susceptible, Infected and Recovery individuals with 5th and 95th quantiles are shown.

We have simulated the population trajectories for the above model, by performing numerous of Monte Carlo simulation, which gives us the stochastic SIR model with mean trajectories for each compartment in the experimental population. Figure 4.8 shows the true epidemic paths and parameters values fell well between 5^{th} and 95^{th}

Bayesian credible intervals in all simulations of the susceptible, infected and recovered individuals in the population as the epidemic progresses. The acceptance rates for subject-path proposals were roughly 50% for three simulated SIR models on Figure Figure 4.8: (A1., B1., B2.) and 90% shown on Figure 4.8: (A2.). Our posterior estimates of the model parameters $\hat{\beta}_1$ $\hat{\gamma}_1$, $\hat{\beta}_2$ and $\hat{\gamma}_2$ are also closely match estimates since the true value of parameters $(\beta, \gamma) = (0.444, 0.3)$ were generated a sequence of random samples from a probability distribution where obtained by using Monte Carlo simulation from page 51.

Chapter 5

Case study: Dengue of Vector(SI)-Host(SIR)

5.1 Background

Dengue is a mosquito-borne viral infection that is usually found in tropical and subtropical regions around the world. Warmer weather and rain bring remarkably good conditions for reproduction for vectors that are carriers and transmit the disease. If a susceptible vector bites an infected human during the viremic period, it may become infected and subsequently transmit the virus to other healthy humans. Humans are the source of nutrients for mosquitoes, that grow and reproduce on stagnant water.

Today, dengue fever is the mosquito-borne infection which is regarded as the major international public health concern, threatening about 2.5 billion people all over the world, especially the tropical countries. Worldwide, there is an annual estimated 50-230 millions new cases, 500,000 hospitalizations and 25,000 fatal cases, mostly among children that are suffering from *hemorrhagic* fever. Dengue is particularly common in Southeast Asia [42].

5.2 Application to Dengue Fever

Dengue is caused by four antigenically distinct virus serotypes, denoted as dengue virus 1 (DENv1), dengue virus 2 (DENv2), dengue virus 3 (DENv3) and dengue virus

4 (DENv4). The corresponding illnesses are dengue fever (DF) or classic dengue, and the dengue haemorrhagic (DHF) which may evolve toward severe form so called dengue shock syndrome (DSS). Disease symptoms are a mild form of sudden fever (DF) without respiratory infection, accompanied by rash, flu-like and intense headaches (myalgia and arthralgia). The latter gives the nicknames "*breakbone fever*" or "*bonecrusher disease*". It last generally between 3-7 days, but it my also persist in a benign way [43]. Some individuals develop (DSS) syndrome where the severity of the disease is dramatically increased with a significant mortality rate due to low blood pressure caused by fluid leakage. It usually lasts between 2-3 days and can lead to death [43]. In some cases, susceptible infected by one of the four serotypical virus will never be infected again by the same serotype known (heterologous immunity), whereas one looses immunity to the three other serotypes (heterologous immunity) is around 12 weeks and subsequently become less resistant dengue haemorrhagic fever again.

5.3 Model Approaches

For the above stated reasons, it is worth studying the mechanisms that allow the invasion of dengue. Models can provide insights into the transmission dynamics, invasion and persistence of a certain serotype of dengue in a community. A detailed derivation of a model can provide a qualitative assessment from a mathematical simulation with parameter estimation, sensitivity and comparison of conjunctures to predicting dengue outbreaks. In this case study, we are using systems of ordinary differential equations (ODE) with deterministic model approaches.

5.3.1 Deterministic Assumptions - Dengue scenarios

All the models considered in this work satisfy the following assumptions [44, 45]:

1. The model assumes a homogeneous mixing of the human and vector(mosquito) populations, so that each mosquito bite has equal chance of transmitting the virus to susceptible human in the population (or acquiring infection from an infected human).

2. Any recovered susceptible has permanent immunity or least considered accordingly within the time frame of the disease model.

3. The sexual ratio of human is 1:1, male and female are subject to almost the same epidemiological factors [46].

4. The end of the viremic period coincides with the disappearance of symptoms in symptomatically infected individuals [46].

5. The model does not accommodate for 4 strains of dengue serotypical virus, because its complexity of four co-circulating serotypes. We are just focus only single-serotype system.

6. No vaccination is available or applied.

7. The population size is constant for the models.

5.3.2 Model Description - Parameters

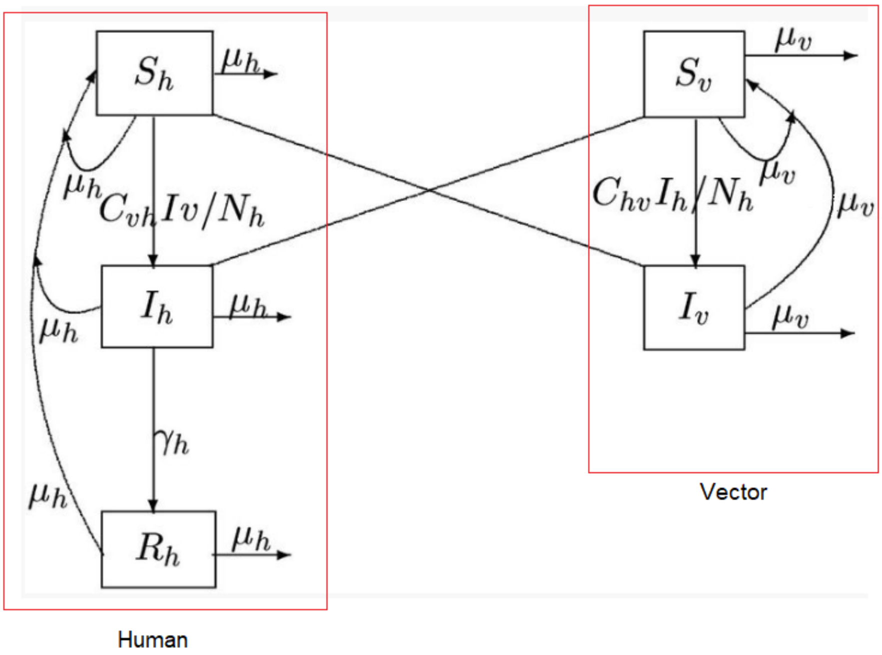

Figure 5.1: Schematic: Human-vector transmission for single-serotype model.

According to the Figure (5.1), we denote a human population (respectively of mosquito population) with size N_h (resp. N_v). As before we have susceptible S_h, infective I_h and recovered R_h (resp. S_v and I_v). The basic parameters used in the model are given table below. [43, 44, 47, 48]:

Interpretation	Notation	Base value	Range
Transmission probability of vector to human	β_{hv}	0.75	0.1-1
Transmission probability of human to vector	β_{vh}	0.75	0.1-1
Bites per susceptible mosquito per day	b_s	0.5	0.3-1
Bites per infectious mosquito per day	b_i	1.0	0.3-1
Effective contact rate, human to vector	C_{hv}	0.375	0.1-1
Effective contact rate, vector to human	C_{vh}	0.75	0.1-1
Average human life span	$\frac{1}{\mu_h}$	25000 days	10950-30000 days
Average vector life span	$\frac{1}{\mu_v}$	4 days	3-14 days
Average recovery rate for human	$\frac{1}{\gamma_h}$	7 days	5-10 days
Average host infection duration	$\frac{1}{\mu_v+\gamma_h}$	3 days	5-10 days

Table 5.1: Description of variables and parameters used in vector-host simulations

5.3.3 Model formulation

Based on the assumptions, the *single-serotype* vector-host transmission model presented by Bailey 1975 [49] provides the basis for coupled SIR host and SI vector models [50]:

Human population

$$\frac{dS_h}{dt} = \mu_h N_h - \frac{\beta_{vh} b_i}{N_h} S_h I_v - \mu_h S_h \tag{5.1}$$

$$\frac{dI_h}{dt} = \frac{\beta_{vh} b_i}{N_h} S_h I_v - (\gamma_h + \mu_h) I_h \tag{5.2}$$

$$\frac{dR_h}{dt} = \gamma_h I_h - \mu_h R_h \tag{5.3}$$

Vector population

$$\frac{dS_v}{dt} = \mu_v N_v - \frac{\beta_{hv} b_s}{N_h} S_v I_h - \mu_v S_v \tag{5.4}$$

$$\frac{dI_v}{dt} = \frac{\beta_{hv} b_s}{N_h} S_v I_h - \mu_v I_v \tag{5.5}$$

The single-serotype system can be simplified to

$$\frac{dS_h}{dt} = \mu_h N_h - \frac{C_{vh}}{N_h} S_h I_v - \mu_h S_h \tag{5.6}$$

$$\frac{dI_h}{dt} = \frac{C_{vh}}{N_h} S_h I_v - (\gamma_h + \mu_h) I_h \tag{5.7}$$

$$\frac{dR_h}{dt} = \gamma_h I_h - \mu_h R_h \tag{5.8}$$

$$\frac{dS_v}{dt} = A - \frac{\beta_{hv} b_s}{N_h} S_v I_h - \mu_v S_v \tag{5.9}$$

$$\frac{dI_v}{dt} = \frac{\beta_{hv} b_s}{N_h} S_v I_h - \mu_v I_v \tag{5.10}$$

where A represents the vector recruitment rate. The human and vector populations remain constant, hence without loss of generality, we can work with the proportions

- β_{vh} (resp. β_{hv}) the average transmission probability of an infectious *vector to human* (resp. *human to vector*).
- I_v (resp. I_h) the number of infectious *vector* (resp. *human*).

This gives

- $C_{hv} = \beta_{hv} b_s$ is the contact rate of *human to vectors.*
- $C_{vh} = \beta_{vh} b_i$ is the contact rate of *vectors to human.*

with the following conditions [51]:

$$S_h + I_h + R_h = N_h \Longrightarrow R_h = N_h - S_h - I_h \tag{5.11}$$

$$S_v + I_v = N_v \Longrightarrow S_v = N_v - I_v \tag{5.12}$$

Hence the model for the human and mosquito

$$\frac{dS_h}{dt} = \mu_h N_h - \frac{C_{vh}}{N_h} S_h I_v - \mu_h S_h \tag{5.13}$$

$$\frac{dI_h}{dt} = \frac{C_{vh}}{N_h} S_h I_v - (\gamma_h + \mu_h) I_h \tag{5.14}$$

$$\frac{dI_v}{dt} = \frac{\beta_{hv} b_s}{N_h} S_v I_h - \mu_v I_v \tag{5.15}$$

is equivalent to the full system (5.6-5.10).

5.4 Equilibrium points

This system (5.13-5.15) is defined by on set Ω given

$$\Omega = \Big\{ (S_h, I_h, I_v) : \ 0 \leq I_v \leq N_v; \ 0 \leq S_h; \ 0 \leq I_h; \ S_h + I_h \leq N_h \Big\} \tag{5.16}$$

We have the equilibrium points:

$$E_0 \ = \ (1, 0, 0) \qquad \textit{and} \qquad E_1 \ = \ (S_h^*, I_h^*, I_v^*) \tag{5.17}$$

where

$$S_h^* \ = \ \frac{\delta + W}{\delta + W R_0}, \quad I_h^* \ = \ \frac{R_0 - 1}{\delta + W R_0}, \quad \text{and} \quad I_v^* \ = \ \frac{\delta (R_0 - 1)}{R_0 (\delta + W)} \tag{5.18}$$

with

$$\alpha = \frac{\delta_{hv}}{\mu_v} \quad \text{and} \quad W = \frac{\mu_h + \gamma_h}{\mu_h} \tag{5.19}$$

Using the next *basic reproductive number* to yield:

$$R_0 = \frac{C_{vh}C_{hv}}{\mu_v(\mu_h + \gamma_h)} \tag{5.20}$$

5.4.1 Stability

1. E_0 is the disease-free equilibrium (DFE). If $R_0 \leq 1$, then $E_0(N_h, 0, 0)$ is globally asymptotically *stable* [43].

2. E_1 is the endemic equilibrium (EE). If $R_0 > 1$, then $E_1(S_h^*, I_h^*, I_h^*)$ is locally asymptotically *unstable* [43].

We can linearize the system of (5.13-5.15) and write in the *Jacobian matrix* form as

$$J(E_1) = \begin{pmatrix} -C_{vh}I_v - \mu_h & 0 & C_{vh}S_h \\ C_{vh}I_v & -(\mu_h + \gamma_h) & C_{vh}S_h \\ 0 & C_{hv} - C_{hv}I_v & -C_{hv}I_h - \mu_v \end{pmatrix} \tag{5.21}$$

5.4.2 Disease-free equilibrium

Equate the equilibrium point at $E_1 = (N_h, 0, 0) = (1, 0, 0)$ and write the Jacobian matrix as

$$\lambda_1 = -\mu_h \tag{5.22}$$

$$\lambda_{2/3} = -(\mu_h + \gamma_h + \mu_v) \pm \frac{1}{2}\sqrt{(\mu_h + \gamma_h + \mu_v)^2 - 4\mu_v(\mu_h + \gamma_h)(1 - R_0)} \tag{5.23}$$

All eigenvalues have negative real part indicates E_1 is locally asymptotically stable for $R_0 < 1$.

5.4.3 Endemic equilibrium

Similarly, we can linearize the system of (5.13-5.15) and write in the *Jacobian matrix* form as [52]:

$$J(E_2) = \begin{pmatrix} -\mu_h\left(\frac{\delta+WR_0}{\delta+W}\right) & 0 & -\mu_h\frac{WR_0}{\delta}\left(\frac{\delta+W}{\delta+WR_0}\right) \\ \mu_h\frac{W(R_0-1)}{\delta+W} & -\mu_h W & \mu_h\frac{WR_0}{\delta}\left(\frac{\delta+W}{\delta+WR_0}\right) \\ 0 & \frac{\mu_v\delta}{R_0}\left(\frac{\delta+WR_0}{\delta+W}\right) & -\mu_v R_0\left(\frac{\delta+W}{\delta+WR_0}\right) \end{pmatrix} \tag{5.24}$$

then the characteristic polynomial of $J(E_2)$ is given by

$$p(\lambda) = \lambda^3 + A\lambda^2 + B\lambda + C \tag{5.25}$$

which implies that:

$$A = -tr(E_2) \tag{5.26}$$

$$B = \begin{vmatrix} J(E_2)_{11} & J(E_2)_{12} \\ J(E_2)_{21} & J(E_2)_{22} \end{vmatrix} + \begin{vmatrix} J(E_2)_{11} & J(E_2)_{13} \\ J(E_2)_{31} & J(E_2)_{33} \end{vmatrix} + \begin{vmatrix} J(E_2)_{22} & J(E_2)_{23} \\ J(E_2)_{32} & J(E_2)_{33} \end{vmatrix} \tag{5.27}$$

$$C = -det\big(J(E_2)\big) \tag{5.28}$$

We obtain

$$A = \mu_h\left(\frac{\delta+WR_0}{\delta+W}\right) + \mu_h W + \mu_v R_0\left(\frac{\delta+W}{\delta+WR_0}\right) \tag{5.29}$$

$$B = \mu_h^2 W\left(\frac{\delta+WR_0}{\delta+W}\right) + \mu_h\mu_v R_0 + \mu_v\mu_h W\left(\delta\frac{(R_0-1)}{\delta+WR_0}\right) \tag{5.30}$$

$$C = \mu_h^2\mu_v W(R_0-1) \tag{5.31}$$

Therefore, the coefficients A, B and C are positive and

$$AB > \mu_h^2 W R_0 > C \tag{5.32}$$

satisfies Routh-Hurwitz condition for the polynomial, $p(\lambda)$. This implies that $E_1(S_h^*, I_h^*, I_h^*)$ is *locally asymptotically stable* for $R_0 > 1$.

5.5 Results and Discussion

In order to show the dynamics of each epidemic and to study different strategies, a simulation was generated using ***R***.

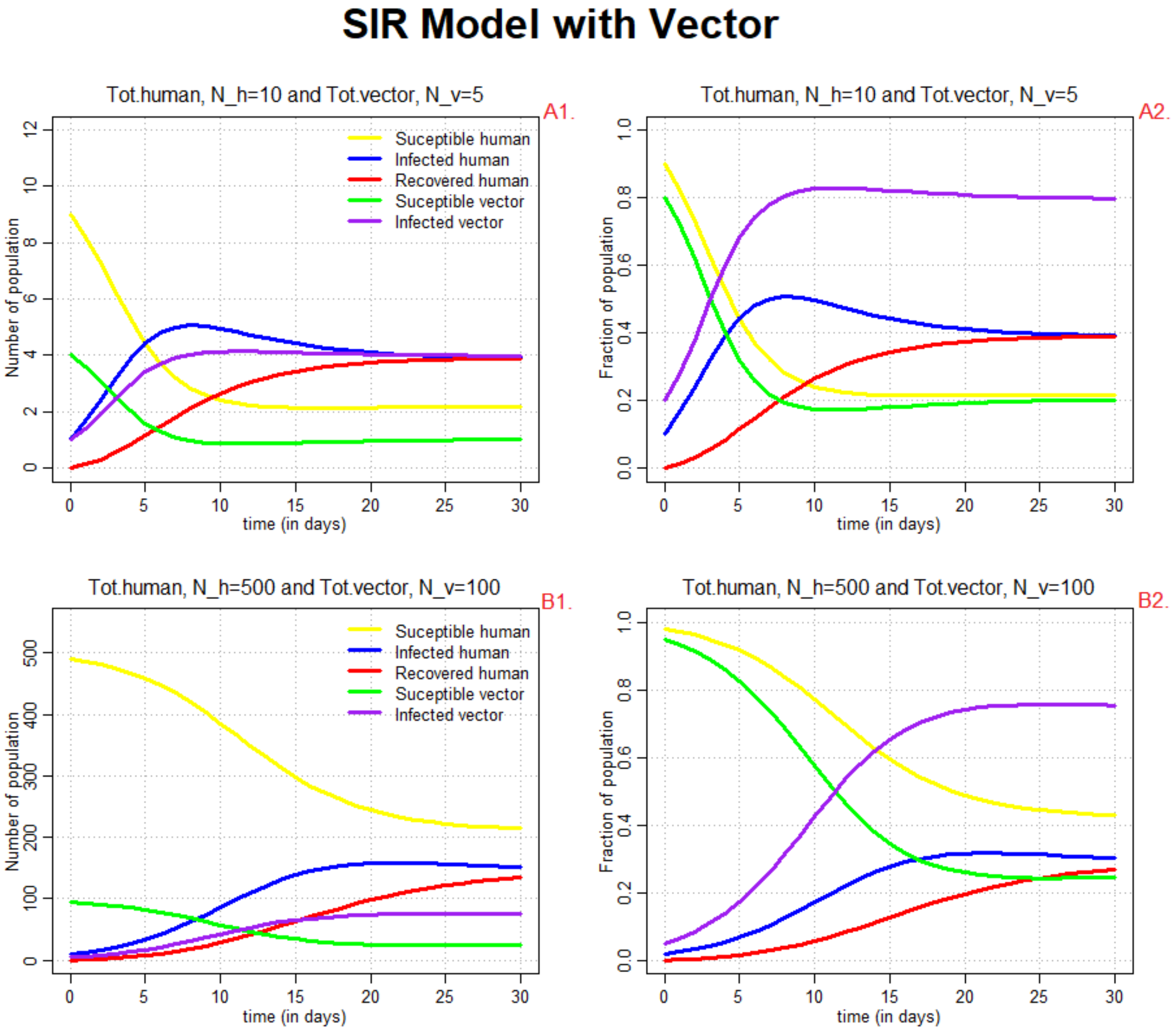

Figure 5.2: Transmission of dengue with initial condition, the graphs (A1) illustrate the number (resp. proportion in A2)) of susceptible human, infective human, recovered human versus time. The parameters in the simulations are A.): $N_h = 10, I_h(0) = 1, N_v = 5, I_v(0) = 1, \beta_{hv} = 1.0, \beta_{vh} = 1.0, b_{hv} = 1.0, b_{vh} = 1.0, \mu_h = 0.1, \mu_v = 0.1, \gamma_h = 0.1$ and B.): $N_h = 500, I_h(0) = 100, N_v = 10, I_v(0) = 5, \beta_{hv} = 1.0, \beta_{vh} = 1.0, b_{hv} = 1.0, b_{vh} = 1.0, \mu_h = 0.1, \mu_v = 0.1, \gamma_h = 0.1$.

There are two fundamentally different approaches that can be used to to control the disease: changing the initial number of mosquitoes, changing the number of of susceptible humans. According to Figure (5.2,A2), the outbreak will reach the maximum level within the next 7 days, with the initial values ($N_h = 10, N_v = 5,\ I_h(0) = 1, I_v(0) = 1$). Compare with the graph Figure (5.2,B2) that the infectious period will take longer to reach its maximum level. However, the number of contracted dengue cases has 20% less than in A2, where initial values are ($N_h = 500, N_v = 100,\ I_h(0) = 10, I_v(0) = 5$).

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis we studied generalizations of the basic SIR model to simulate few different epidemic scenarios. This provides knowledge about the how the evolution of biological diseases work. We use mathematical models to predict the disease. For instance, hypothetical zombie apocalypse is particularly interesting. By using a mathematical ordinary differential equation (ODE) model, it is possible to do the risk assessment by constructing and allocate few different compartments.

For the deterministic models, the theorem for "basic reproductive ratio, R_0", stated on the page 2, tells us that R_0 determines whether the disease is eliminated or persists. In the case where $R_0 > 1$, the occurrence of disease will become endemic (prevalent), whereas if $R_0 < 1$ the disease will die out. This work can be found on Chapter 3, page 17.

In chapter 4, on page 31, an extension of SIR model embedded with essential birth and death dynamics is discussed. The purpose of this approach was to study the effect of growth and change of human population. We believe that by including *demography* with *seasonal forcing*, the models yield more realistic results.

On page 47, the stochastic epidemic Markov Chain model was used. Following section on page 55, Metropolis-Hastings algorithm was implemented from `R` package which given a very desired of relatively precision for unknown parameter estimation. Lastly, we modified and added extra compartments to formulate the SI-SIR model in Chapter 6. On page 66 a case study was introduced to investigate the transmission dynamics of dengue single-serotype models.

6.2 Future Work

As a future work, I would like to extend the SIR models using *Bayesian Markov Chain Monte Carlo* estimation. This methods use the prior parameter distribution to estimate the best guess of parameters. This can helpful in providing better accuracy of parameters. Another approach is to use SDE to on considerably larger population sizes. SDE simulations can run almost as quick as deterministic ODE models, whereas the downside of Markov chains is that they can be very slow to converge for large populations especially having multiple iterations. Another possible tool is the *Latin Hypercube*, which has several advantages to when it comes to goodness-of-fit and assessing optimal parameters.

Bibliography

[1] Klaus Dietz and J.A.P. Heesterbeek. Daniel bernoulli's epidemiological model revisited. *Mathematical Biosciences*, 180(1):1 – 21, 2002.

[2] P.D.En'ko. The epidemic course of some infectious diseases. *Vrac*, 10::1008–1010, 1889.

[3] P.D.En'ko. On the course of epidemics of some infectious diseases. *International Journal of Epidemiology*, 18(4):749–755, 1989.

[4] W.H. Hamer. *Epidemic Disease in England: The Evidence of Variability and of Persistency of Type.* Milroy lectures. Bedford Press, 1906.

[5] Nicolas Bacaer. *Ross and malaria.* A Short History of Mathematical Population Dynamics. Springer, 1911.

[6] N. Bacaër. *A Short History of Mathematical Population Dynamics.* Springer London, 2011.

[7] W. O. Kermack and A. 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.

[8] Norman C. Severo. Generalizations of some stochastic epidemic models. *Mathematical Biosciences*, 4(3):395 – 402, 1969.

[9] Keith N. Crank. A method for approximating the probability functions of a markov chain. *Journal of Applied Probability*, 25(4):808–814, 1988.

[10] Raymond S. Koff. Infectious diseases of humans: Dynamics and control. by r.m. anderson and r.m. may, 757 pp. oxford: Oxford university press, 1991. $95.00. *Hepatology*, 15(1):169–169.

[11] Wei-min Liu, Herbert W. Hethcote, and Simon A. Levin. Dynamical behavior of epidemiological models with nonlinear incidence rates. *Journal of Mathematical Biology*, 25(4):359–380, Sep 1987.

[12] L.J. Ping, B. Igor, and N.L. M. *Information Computing And Automation (In 3 Volumes) - Proceedings Of The International Conference*. World Scientific Publishing Company, 2008.

[13] Constantinos I. Siettos and Lucia Russo. Mathematical modeling of infectious disease dynamics. *Virulence*, 4(4):295–306, 2013. PMID: 23552814.

[14] K. Dietz and D. Schenzle. Mathematical models for infectious disease statistics. In Anthony C. Atkinson and Stephen E. Fienberg, editors, *A Celebration of Statistics*, pages 167–204, New York, NY, 1985. Springer New York.

[15] Frank Ball and Peter Neal. A general model for stochastic sir epidemics with two levels of mixing. *Mathematical Biosciences*, 180(1):73 – 102, 2002.

[16] A. Huppert and G. Katriel. Mathematical modelling and prediction in infectious disease epidemiology. *Clinical Microbiology and Infection*, 19(11):999 – 1005, 2013.

[17] Maia Martcheva. *An Introduction to Mathematical Epidemiology*. Springer Publishing Company, Incorporated, 1st edition, 2015.

[18] Nita Shah and Gupta Jyoti. Seir model and simulation for vector borne diseases. 4:13 – 17, 01 2013.

[19] James D. Murray. *Mathematical Biology I. An Introduction*, volume 17 of *Interdisciplinary Applied Mathematics*. Springer, New York, 3 edition, 2002.

[20] Pauline van den Driessche. Reproduction numbers of infectious disease models. *Infectious Disease Modelling*, 2(3):288 – 303, 2017.

[21] Koya Purnachandra Rao and Dejen Mamo. Ebola epidemic disease: Modelling, stability analysis, spread control technique, simulation study and data fitting. 2:476–484, 03 2015.

[22] Herbert W. Hethcote. *Three Basic Epidemiological Models*, pages 119–144. Springer Berlin Heidelberg, Berlin, Heidelberg, 1989.

[23] H. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653, 2000.

[24] M.Y. Kim and F.A. Milner. A mathematical model of epidemics with screening and variable infectivity. *Mathematical and Computer Modelling*, 21(7):29 – 42, 1995.

[25] Maia Martcheva. *The SIR Model with Demography: General Properties of Planar Systems*, pages 33–66. Springer US, Boston, MA, 2015.

[26] Herbert W. Hethcote. Qualitative analyses of communicable disease models. *Mathematical Biosciences*, 28(3):335 – 356, 1976.

[27] Chengjun Sun and Ying-Hen Hsieh. Global analysis of an seir model with varying population size and vaccination. *Applied Mathematical Modelling*, 34(10):2685 – 2697, 2010.

[28] Sarah A. Al-Sheikh. Modeling and analysis of an seir epidemic model with a limited resource for treatment. 2013.

[29] Amira Rachah and Delfim F. M. Torres. Analysis, simulation and optimal control of a seir model for ebola virus with demographic effects. 67, 05 2017.

[30] Azizur Rahman and Abdul Kuddus. A new model to study on physical behaviour among susceptible infective removal population. 46:115–135, 01 2014.

[31] Alun Lloyd. Introduction to epidemiological modeling: Basic models and their properties. *Theoretical Population Biology*, page 168, 2017.

[32] Virginia E. Pitzer and Marc Lipsitch. Exploring the relationship between incidence and the average age of infection during seasonal epidemics. *Journal of Theoretical Biology*, 260(2):175 – 185, 2009.

[33] W.R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC Interdisciplinary Statistics. Taylor & Francis, 1995.

[34] Gely P. Basharin, Amy N. Langville, and Valeriy A. Naumov. The life and work of a.a. markov. *Linear Algebra and its Applications*, 386:3 – 26, 2004. Special Issue on the Conference on the Numerical Solution of Markov Chains 2003.

[35] Anoop Korattikara Balan, Yutian Chen, and Max Welling. Austerity in MCMC land: Cutting the metropolis-hastings budget. *CoRR*, abs/1304.5299, 2013.

[36] L. S. T. Ho, F. W. Crawford, and M. A. Suchard. Direct likelihood-based inference for discretely observed stochastic compartmental models of infectious disease. *ArXiv e-prints*, August 2016.

[37] Lam Si Tung Ho, Jason Xu, Forrest W. Crawford, Vladimir N. Minin, and Marc A. Suchard. Birth/birth-death processes and their computable transition probabilities with biological applications. *Journal of mathematical biology*, 76 4:911–944, 2018.

[38] Andrew Martin, Kevin Quinn, and Jong Hee Park. Mcmcpack: Markov chain monte carlo in r. *Journal of Statistical Software, Articles*, 42(9):1–21, 2011.

[39] Mary Kathryn Cowles and Bradley P. Carlin. Markov chain monte carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883–904, 6 1996.

[40] Ghassan Hamra, Richard MacLehose, and David Richardson. Markov chain monte carlo: an introduction for epidemiologists. *International Journal of Epidemiology*, 42(2):627–634, 2013.

[41] Stephen P. Brooks and Gareth O. Roberts. Convergence assessment techniques for markov chain monte carlo. *Statistics and Computing*, 8(4):319–335, December 1998.

[42] World Health Organization., Special Programme for Research, Training in Tropical Diseases., World Health Organization., and World Health Organization.

Dengue : guidelines for diagnosis, treatment, prevention and control. World Health Organization Geneva, new ed. 2009 edition, 2009.

[43] M. Derouich, A. Boutayeb, and EH Twizell. A model of dengue fever. *BioMedical Engineering OnLine*, 2(1):4, Feb 2003.

[44] S.M. Garba, A.B. Gumel, and M.R. Abu Bakar. Backward bifurcations in dengue transmission dynamics. *Mathematical Biosciences*, 215(1):11 – 25, 2008.

[45] Silvia Raimundo Martorano, Marcos Amaku, and Eduardo Massad. Equilibrium analysis of a yellow fever dynamical model with vaccination. *Computational and Mathematical Methods in Medicine*, 03 2015.

[46] Daozhou Gao, Yijun Lou, Daihai He, Travis Porco, Yang Kuang, Gerardo Chowell, and Shigui Ruan. Prevention and control of zika as a mosquito-borne and sexually transmitted disease: A mathematical modeling analysis. 6:28070, 06 2016.

[47] Mathieu Andraud, Niel Hens, Christiaan Marais, and Philippe Beutels. Dynamic epidemiological models for dengue transmission: A systematic review of structural approaches. *PLOS ONE*, 7:1–14, 11 2012.

[48] Syafruddin Side and Moh Salmi Md. Noorani. A sir model for spread of dengue fever disease (simulation for south sulawesi indonesia and selangor malaysia), 08 2015.

[49] Norman Bailey. *The Mathematical Theory of Infectious Diseases and its Applications.* Griffin, London, 1975.

[50] Iurii Bakach James Braselton. A survey of mathematical models of dengue fever. 08, 01 2015.

[51] Hui-Ming Wei, Xue-Zhi Li, and Maia Martcheva. An epidemic model of a vector-borne disease with direct transmission and time delay. *Journal of Mathematical Analysis and Applications*, 342(2):895 – 908, 2008.

[52] Abhishek Pandey, Anuj Mubayi, and Jan Medlock. Comparing vector–host and sir models for dengue transmission. *Mathematical Biosciences*, 246(2):252 – 259, 2013.

Appendix A

Chapter 3: R script

A.1 SIR function:

```
require(”deSolve”)
require(”sfsmisc”)

derivative_calc_func=function(t, x, vparameters){

   S = x[1]
   I = x[2]
   R = x[3]

   with(as.list(vparameters),{

      npop = S+I+R

      dS = -beta*S*I/npop
      dI = +beta*S*I/npop - gamma*I
      dR = +gamma*I

      vout = c(dS,dI,dR)
      list(vout)
   })
}

derivative_calc_func_with_demographics=function(t, x, vparameters){
   S = x[1]
   I = x[2]
   R = x[3]

   with(as.list(vparameters),{
      npop = S+I+R
      dS = -beta*S*I/npop - mu*S + npop*mu
      dI = +beta*S*I/npop - gamma*I - mu*I
      dR = +gamma*I - mu*R
      out = c(dS,dI,dR)
      list(out)
   })
}
```

A.2 Basic reproductive number, $R_0 < 1$

```
npop = 10000
I_0 = 500
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 150
vt = seq(tbegin,tend,1)

gamma = 0.2
R0    = 0.82
beta  = R0*gamma

vparameters = c(gamma=gamma,beta=beta)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func, vparameters))

vS = solved_model$S
vI = solved_model$I
vR = solved_model$R
vtime = solved_model$time
vnpop = vS+vI+vR

mult.fig(mfrow=c(2,2),main="SIR_model_with_R0=0.82")

#1(
ymin = 0.9*min(vS/vnpop)
plot(vtime,vS/vnpop,type="l",xlab="time",
ylab="fraction_susceptible",ylim=c(ymin,1),lwd=3,main="Susceptible")
iind = which.min(abs(vS/vnpop-1/R0)) # find the index at which S/N is equal to 1/R0
lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend("topright",
legend=c("total_susceptibles","time_at_which_S=1/R0"),bty="n",lwd=3,col=c(1,3),cex=1.0)
#)


#2(
plot(vtime,vI/npop,type="l",xlab="time",
ylab="fraction_infected",ylim=c(0,0.2),lwd=3,col=4,main="Infective")
n=length(vtime)
lines(vtime[2:n],-diff(vS)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend("topright",
legend=c("total_infected_(prevalence)","newly_infected/day_(incidence)"),bty="n",lwd=3,col=c(4,2))
#)


#3(
plot(vtime,log(vI/vnpop),type="l",xlab="time",
ylab="log(fraction_infected)",lwd=3,col=4,main="log(Infected)")
text(25,-7,"Initial\n_exponential\n_drop",cex=0.7)
lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend("topright",legend=c("time_at_which_S=1/R0","log(Infected)"),bty="n",lwd=3,col=c(3,4),cex=0.9)
#)


#4(
plot(vtime,vR/npop,type="l",xlab="time",
ylab="fraction_recovered",ylim=c(0,0.4),lwd=3,col=1,main="Recovery")
lines(vtime[2:n],diff(vR)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend(10,0.41,legend=c("total_recovery",
"newly_recoverd/day_(incidence)","time_at_which_S=1/R0"),bty="n",lwd=3,col=c(1,2,3),cex=0.7)
#)
```

A.3 Basic reproductive number, $R_0 \approx 1$

```
npop = 10000
I_0 = 500
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 150
vt = seq(tbegin,tend,1)

gamma = 0.2093
R0    = 1.009
beta  = R0*gamma

vparameters = c(gamma=gamma,beta=beta)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func, vparameters))

vS = solved_model$S
vI = solved_model$I
vR = solved_model$R
vtime = solved_model$time
vnpop = vS+vI+vR

mult.fig(4,main="SIR_model_with_R0=1.009")

#1(
ymin = 0.9*min(vS/vnpop)
plot(vtime,vS/vnpop,type="l",xlab="time",
ylab="fraction_susceptible",ylim=c(ymin,1),lwd=3,main="Susceptible")

iind = which.min(abs(vS/vnpop-1/R0)) # find the index at which S/N is equal to 1/R0
lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend("topright",
legend=c("total_susceptibles","time_at_which_S=1/R0"),bty="n",lwd=3,col=c(1,3),cex=1.0)
#)


#2(
plot(vtime,vI/npop,type="l",xlab="time",
ylab="fraction_infected",ylim=c(0,0.2),lwd=3,col=4,main="Infective")

n=length(vtime)
lines(vtime[2:n],-diff(vS)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend("topright",
legend=c("total_infected_(prevalence)","newly_infected/day_(incidence)"),bty="n",lwd=3,col=c(4,2))
#)


#3(
plot(vtime,log(vI/vnpop),type="l",xlab="time",
ylab="log(fraction_infected)",lwd=3,col=4,main="log(Infected)")
text(25,-7,"Initial\n_exponential\n_drop",cex=0.7)
lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend("topright",
legend=c("time_at_which_S=1/R0","log(Infected)"),bty="n",lwd=3,col=c(3,4),cex=0.9)
#)


#4(
plot(vtime,vR/npop,type="l",xlab="time",
ylab="fraction␣recovered",ylim=c(0,0.4),lwd=3,col=1,main="Recovery")
lines(vtime[2:n],diff(vR)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend(10,0.41,legend=c("total␣recovery",
"newly␣recoverd/day␣(incidence)","time␣at␣which␣S=1/R0"),bty="n",lwd=3,col=c(1,2,3),cex=0.7)
#)
```

A.4 Basic reproductive number, $R_0 > 1$

```
npop = 10000
I_0 = 10
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 150
vt = seq(tbegin,tend,1)

gamma = 0.3
R0    = 1.48
beta  = R0*gamma

vparameters = c(gamma=gamma,beta=beta)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func, vparameters))

vS = solved_model$S
vI = solved_model$I
vR = solved_model$R
vtime = solved_model$time
vnpop = vS+vI+vR

mult.fig(4,main="SIR␣model␣with␣R0=1.5")

#1(
ymin = 0.9*min(vS/vnpop)
plot(vtime,vS/vnpop,type="l",xlab="time",
ylab="fraction␣susceptible",ylim=c(ymin,1),lwd=3,main="Susceptible")
iind = which.min(abs(vS/vnpop-1/R0)) # find the index at which S/N is equal to 1/R0
lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend("topright",
legend=c("total␣susceptibles","time␣at␣which␣S=1/R0"),bty="n",lwd=3,col=c(1,3),cex=1.0)
#)


#2(
plot(vtime,vI/npop,type="l",xlab="time",
ylab="fraction␣infected",ylim=c(0,0.2),lwd=3,col=4,main="Infective")
n=length(vtime)
lines(vtime[2:n],-diff(vS)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend(11.5,0.21,
legend=c("total␣infected␣(prevalence)","newly␣infected/day␣(incidence)"),bty="n",lwd=3,col=c(4,2))
#)


#3(
plot(vtime,log(vI/vnpop),type="l",xlab="time",
ylab="log(fraction␣infected)",lwd=3,col=4,main="log(Infected)")
text(20,-7,"Initial\n␣exponential\n␣increase",cex=0.7)
lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend("topright",
legend=c("time␣at␣which␣S=1/R0","log(Infected)"),bty="n",lwd=3,col=c(3,4),cex=0.85)
#)


#4(
plot(vtime,vR/npop,type="l",xlab="time",
ylab="fraction␣recovered",ylim=c(0,0.6),lwd=3,col=1,main="Recovery")
lines(vtime[2:n],diff(vR)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend(44,0.20,legend=c("total␣recovery",
"newly␣recoverd/day␣(incidence)","time␣at␣which␣S=1/R0"),bty="n",lwd=3,col=c(1,2,3),cex=0.75)
#)
```

Appendix B

Chapter 4: R script

B.1 SIR function

:

```
derivative_calc_func_with_demographics=function(t, x, vparameters){
  S = x[1]
  I = x[2]
  R = x[3]

  with(as.list(vparameters),{
    npop = S+I+R
      #beta = beta0*(1+epsilon*cos(2*pi*(t-phi)/365.25))
      #dS = -beta*S*I/npop + npop*B - nu*S
    dS = -beta*S*I/npop + npop*mu - nu*S
    dI = +beta*S*I/npop - gamma*I - nu*I
    dR = +gamma*I - nu*R
    out = c(dS,dI,dR)
    list(out)
  })
}
```

B.2 SIR Model (with Birth and Death Rates)

```
#npop = 10000
npop = 1000
I_0 = 10
#I_0 = 500
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 500
vt = seq(tbegin,tend,1)

gamma = 1/3#0.3
R0    = 0.98#1.48
beta  = R0*gamma
```

```
mu = 0.01
nu = 0.01

vparameters = c(gamma=gamma, beta=beta ,mu=mu)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits , vt, derivative_calc_func_with_demographics , vparameters))

vS = solved_model$S
vI = solved_model$I
vR = solved_model$R
vtime = solved_model$time
vnpop = vS+vI+vR

mult.fig(mfrow=c(3,4),main=”SIR␣model␣with␣R0=0.98,␣R0=1.48␣and␣R0=1.7”,cex.main=2)

#1(
ymin = 0.9*min(vS/vnpop)
plot(vtime,vS/vnpop,type=”l”,xlab=”time”,
ylab=”fraction␣susceptible”,ylim=c(0.9,1),lwd=3,main=substitute(paste(”Susceptible ,␣R0=”,R0),list(R0=R0)))
iind = which.min(abs(vS/vnpop-1/R0)) # find the index at which S/N is equal to 1/R0
legend(”bottomright”,legend=c(”total␣susceptibles”),bty=”n”,lwd=3,col=c(1),cex=1.0)
#)


#2(
plot(vtime,vI/npop,type=”l”,xlab=”time”,
ylab=”fraction␣infected”,ylim=c(0,0.01),lwd=3,col=4,main=substitute(paste(”Infective ,␣R0=”,R0),list(R0=R0)))
n=length(vtime)
lines(vtime[2:n],-diff(vS)/(diff(vtime)*vnpop[1:(n-1)]),type=”l”,lwd=3,col=2)
legend(”topright”,
legend=c(”total␣infected␣(prevalence)”,”newly␣infected/day␣(incidence)”),bty=”n”,lwd=3,col=c(4,2))
#)


#3(
plot(vtime,log(vI/vnpop),type=”l”,xlab=”time”,
ylab=”log␣(fraction␣infected)”,lwd=3,col=4,main=substitute(paste(”Log␣(Infected),␣R0=”,R0),list(R0=R0)))
text(100,-10,”Initial\n␣exponential\n␣increase”,cex=0.7)
legend(”bottomleft”,legend=c(”log␣(Infected)”),bty=”n”,lwd=3,col=c(4),cex=1)
#)


#4(
plot(vtime,vR/npop,type=”l”,xlab=”time”,
ylab=”fraction␣recovered”,ylim=c(0,0.1),lwd=3,col=1,main=substitute(paste(”Recovery,␣R0=”,R0),list(R0=R0)))
lines(vtime[2:n],diff(vR)/(diff(vtime)*vnpop[1:(n-1)]),type=”l”,lwd=3,col=2)
#lines(c(vtime[iind],vtime[iind]),c(-1000,1000),col=3,lwd=3)
legend(”topright”,
legend=c(”total␣recovery”,”newly␣recoverd/day␣(incidence)”),bty=”n”,lwd=3,col=c(1,2),cex=0.9)
#)

####################################################################################################

#npop = 10000
npop = 1000
I_0 = 10
#I_0 = 500
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 500
vt = seq(tbegin,tend,1)

gamma = 1/3#0.3
```

```
R0     = 1.48#1.48
beta   = R0*gamma
mu = 0.01
nu = 0.01

vparameters = c(gamma=gamma,beta=beta,mu=mu)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func_with_demographics, vparameters))

vS = solved_model$S
vI = solved_model$I
vR = solved_model$R
vtime = solved_model$time
vnpop = vS+vI+vR

#1(
ymin = 0.9*min(vS/vnpop)
plot(vtime,vS/vnpop,type="l",xlab="time",
ylab="fraction_susceptible",
ylim=c(ymin,1),lwd=3,main=substitute(paste("Susceptible,_R0=",R0),list(R0=R0)))
iind = which.min(abs(vS/vnpop-1/R0)) # find the index at which S/N is equal to 1/R0
legend("bottomright",legend=c("total_susceptibles"),bty="n",lwd=3,col=c(1),cex=1.0)
#)


#2(
plot(vtime,vI/npop,type="l",xlab="time",
ylab="fraction_infected",
ylim=c(0,0.1),lwd=3,col=4,main=substitute(paste("Infective,_R0=",R0),list(R0=R0)))
n=length(vtime)
lines(vtime[2:n],-diff(vS)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend("topright",
legend=c("total_infected_(prevalence)","newly_infected/day_(incidence)"),bty="n",lwd=3,col=c(4,2))
#)


#3(
plot(vtime,log(vI/vnpop),type="l",xlab="time",
ylab="log_(fraction_infected)",lwd=3,col=4,main=substitute(paste("Log_(Infected),_R0=",R0),list(R0=R0)))
text(140,-10,"Initial\n_exponential\n_increase",cex=0.7)
legend("bottomright",legend=c("log_(Infected)"),bty="n",lwd=3,col=c(4),cex=1)
#)


#4(
plot(vtime,vR/npop,type="l",xlab="time",
ylab="fraction_recovered",
ylim=c(0,0.6),lwd=3,col=1,main=substitute(paste("Recovery,_R0=",R0),list(R0=R0)))
lines(vtime[2:n],diff(vR)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend("topright",
legend=c("total_recovery","newly_recoverd/day_(incidence)"),bty="n",lwd=3,col=c(1,2),cex=0.9)
#)44,0.20

########################################################################################################################

#npop = 10000
npop = 1000
I_0 = 10
#I_0 = 500
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 1000
vt = seq(tbegin,tend,1)
```

```
gamma = 1/3#0.3
R0    = 1.7#1.48
beta  = R0*gamma
mu = 0.01
nu = 0.01

vparameters = c(gamma=gamma, beta=beta,mu=mu)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func_with_demographics, vparameters))

vS = solved_model$S
vI = solved_model$I
vR = solved_model$R
vtime = solved_model$time
vnpop = vS+vI+vR

#1(
ymin = 0.9*min(vS/vnpop)
plot(vtime,vS/vnpop,type="l",xlab="time",
ylab="fraction␣susceptible",
ylim=c(ymin,1),lwd=3,main=substitute(paste("Susceptible,␣R0=",R0),list(R0=R0)))
iind = which.min(abs(vS/vnpop-1/R0)) # find the index at which S/N is equal to 1/R0
legend("bottomright",legend=c("total␣susceptibles"),bty="n",lwd=3,col=c(1),cex=1.0)
#)


#2(
plot(vtime,vI/npop,type="l",xlab="time",
ylab="fraction␣infected",
ylim=c(0,0.1),lwd=3,col=4,main=substitute(paste("Infective,␣R0=",R0),list(R0=R0)))
n=length(vtime)
lines(vtime[2:n],-diff(vS)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend("topright",
legend=c("total␣infected␣(prevalence)","newly␣infected/day␣(incidence)"),bty="n",lwd=3,col=c(4,2))
#)


#3(
plot(vtime,log(vI/vnpop),type="l",xlab="time",
ylab="log␣(fraction␣infected)",lwd=3,col=4,main=substitute(paste("Log␣(Infected),␣R0=",R0),list(R0=R0)))
text(140,-10,"Initial\n␣exponential\n␣increase",cex=0.7)
legend("bottomright",
legend=c("log␣(Infected)"),bty="n",lwd=3,col=c(4),cex=1)
#)


#4(
plot(vtime,vR/npop,type="l",xlab="time",
ylab="fraction␣recovered",
ylim=c(0,0.6),lwd=3,col=1,main=substitute(paste("Recovery,␣R0=",R0),list(R0=R0)))
lines(vtime[2:n],diff(vR)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend("topright",
legend=c("total␣recovery","newly␣recoverd/day␣(incidence)"),bty="n",lwd=3,col=c(1,2),cex=0.9)
#)
```

B.3 Seasonal Transmission in the Absence of Birth Seasonality

```
derivative_calc_func_with_demographics=function(t, x, vparameters){
   S = x[1]
   I = x[2]
   R = x[3]

   with(as.list(vparameters),{
      npop = S+I+R
         beta = beta0*(1+epsilon*cos(2*pi*(t-phi)/365.25))
         dS = -beta*S*I/npop + npop*B - nu*S
      #dS = -beta*S*I/npop + npop*mu - nu*S
      dI = +beta*S*I/npop - gamma*I - nu*I
      dR = +gamma*I - nu*R
      out = c(dS,dI,dR)
      list(out)
   })
}



npop = 1000
I_0 = 10
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 1250
vt = seq(tbegin,tend,1)

gamma = 1/3#0.3
R0     = 0.98#1.48
beta0  = R0*gamma
B = 0.01
mu = 0.01
nu = 0.01
phi   = 0
epsilon1 = 0.3


vparameters = c(gamma=gamma,beta0=beta0,phi=phi,epsilon=epsilon1,mu=mu,nu=nu,B=B)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func_with_demographics, vparameters))

vS = solved_model$S
vI = solved_model$I
vR = solved_model$R
vtime = solved_model$time
vnpop = vS+vI+vR

mult.fig(mfrow=c(3,4),main="Seasonal_SIR_model_with_R0=0.98,_R0=1.20_and_R0=1.70",cex.main=2)

#1(
ymin = 0.9*min(vS/vnpop)
plot(vtime,vS/vnpop,type="l",xlab="time",
ylab="fraction_susceptible",ylim=c(ymin,1),lwd=3,main=substitute(paste("Susceptible,_R0=",R0),list(R0=R0)))
iind = which.min(abs(vS/vnpop-1/R0)) # find the index at which S/N is equal to 1/R0
legend("bottomright",legend=c("total_susceptibles"),bty="n",lwd=3,col=c(1),cex=1.0)
#)


#2(
plot(vtime,vI/npop,type="l",xlab="time",
ylab="fraction_infected",ylim=c(0,0.01),lwd=3,col=4,main=substitute(paste("Infective,_R0=",R0),list(R0=R0)))
n=length(vtime)
lines(vtime[2:n],-diff(vS)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend(11.5,0.21,
legend=c("total_infected_(prevalence)","newly_infected/day_(incidence)"),bty="n",lwd=3,col=c(4,2))
#)


#3(
plot(vtime,log(vI/vnpop),type="l",xlab="time",
ylab="log(fraction_infected)",lwd=3,col=4,main=substitute(paste("Log_(Infected),_R0=",R0),list(R0=R0)))
legend("topright",legend=c("log(Infected)"),bty="n",lwd=3,col=c(4),cex=1)
#)


#4(
plot(vtime,vR/npop,type="l",xlab="time",
ylab="fraction_recovered",ylim=c(0,0.8),lwd=3,col=1,main=substitute(paste("Recovery,_R0=",R0),list(R0=R0)))
lines(vtime[2:n],diff(vR)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend("topright",
legend=c("total_recovery","newly_recoverd/day_(incidence)"),bty="n",lwd=3,col=c(1,2),cex=1)
#)

##################################################################################################################

npop = 1000
I_0 = 10
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 1250
vt = seq(tbegin,tend,1)

gamma = 1/3#0.3
R0     = 1.2#1.48
beta0  = R0*gamma
B = 0.01
mu = 0.01
nu = 0.01
phi   = 0
epsilon1 = 0.3

vparameters = c(gamma=gamma,beta0=beta0,phi=phi,epsilon=epsilon1,mu=mu,B=B)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func_with_demographics, vparameters))

vS = solved_model$S
vI = solved_model$I
vR = solved_model$R
vtime = solved_model$time
vnpop = vS+vI+vR

#1(
ymin = 0.9*min(vS/vnpop)
plot(vtime,vS/vnpop,type="l",xlab="time",
ylab="fraction_susceptible",ylim=c(ymin,1),lwd=3,main=substitute(paste("Susceptible,_R0=",R0),list(R0=R0)))
iind = which.min(abs(vS/vnpop-1/R0)) # find the index at which S/N is equal to 1/R0
legend("bottomright",legend=c("total_susceptibles"),bty="n",lwd=3,col=c(1),cex=1.0)
#)


#2(
plot(vtime,vI/npop,type="l",xlab="time",
ylab="fraction_infected",ylim=c(0,0.2),lwd=3,col=4,main=substitute(paste("Infective,_R0=",R0),list(R0=R0)))
n=length(vtime)
lines(vtime[2:n],-diff(vS)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend(11.5,0.21,
legend=c("total_infected_(prevalence)","newly_infected/day_(incidence)"),bty="n",lwd=3,col=c(4,2))
```

```
#)


#3(
plot(vtime,log(vI/vnpop),type="l",xlab="time",
ylab="log(fraction␣infected)",lwd=3,col=4,main=substitute(paste("Log␣(Infected),␣R0=",R0),list(R0=R0)))
legend(650,-19.5,legend=c("log(Infected)"),bty="n",lwd=3,col=c(4),cex=1)
#)


#4(
plot(vtime,vR/npop,type="l",xlab="time",
ylab="fraction␣recovered",ylim=c(0,0.8),lwd=3,col=1,main=substitute(paste("Recovery,␣R0=",R0),list(R0=R0)))
lines(vtime[2:n],diff(vR)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend("topright",legend=c("total␣recovery","newly␣recoverd/day␣(incidence)"),bty="n",lwd=3,col=c(1,2),cex=1)
#)

##################################################################################################################

npop = 1000
I_0 = 10
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 1250
vt = seq(tbegin,tend,1)

gamma = 1/3#0.3
R0    = 1.7#1.48
beta0  = R0*gamma
B = 0.01
mu = 0.01
nu = 0.01
phi   = 0
epsilon1 = 0.3


vparameters = c(gamma=gamma,beta0=beta0,phi=phi,epsilon=epsilon1,mu=mu)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func_with_demographics, vparameters))

vS = solved_model$S
vI = solved_model$I
vR = solved_model$R
vtime = solved_model$time
vnpop = vS+vI+vR

#1(
ymin = 0.9*min(vS/vnpop)
plot(vtime,vS/vnpop,type="l",xlab="time",
ylab="fraction␣susceptible",ylim=c(ymin,1),lwd=3,main=substitute(paste("Susceptible,␣R0=",R0),list(R0=R0)))
iind = which.min(abs(vS/vnpop-1/R0)) # find the index at which S/N is equal to 1/R0
legend("bottomright",legend=c("total␣susceptibles"),bty="n",lwd=3,col=c(1),cex=1.0)
#)


#2(
plot(vtime,vI/npop,type="l",xlab="time",
ylab="fraction␣infected",ylim=c(0,0.2),lwd=3,col=4,main=substitute(paste("Infective,␣R0=",R0),list(R0=R0)))
n=length(vtime)
lines(vtime[2:n],-diff(vS)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend(11.5,0.21,
legend=c("total␣infected␣(prevalence)","newly␣infected/day␣(incidence)"),bty="n",lwd=3,col=c(4,2))
#)
```

```
#3(
plot(vtime,log(vI/vnpop),type="l",xlab="time",
ylab="log(fraction infected)",lwd=3,col=4,main=substitute(paste("Log (Infected), R0=",R0),list(R0=R0)))
legend("bottomright",legend=c("log(Infected)"),bty="n",lwd=3,col=c(4),cex=1)
#)


#4(
plot(vtime,vR/npop,type="l",xlab="time",
ylab="fraction recovered",ylim=c(0,0.8),lwd=3,col=1,main=substitute(paste("Recovery, R0=",R0),list(R0=R0)))
lines(vtime[2:n],diff(vR)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend("topright",legend=c("total recovery","newly recoverd/day (incidence)"),bty="n",lwd=3,col=c(1,2),cex=1)
#)
```

B.4 Seasonal Transmission in the Presence of Birth Seasonality

```
npop = 1000
I_0 = 10
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 1250
vt = seq(tbegin,tend,1)

gamma = 1/3#0.3
R0    = 0.98#1.48
beta0  = R0*gamma
B = 0.011
mu = 0.01
nu = 0.01
phi   = 0
epsilon1 = 0.3

vparameters = c(gamma=gamma,beta0=beta0,phi=phi,epsilon=epsilon1,mu=mu,nu=nu,B=B)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func_with_demographics, vparameters))

vS = solved_model$S
vI = solved_model$I
vR = solved_model$R
vtime = solved_model$time
vnpop = vS+vI+vR

mult.fig(mfrow=c(3,4),main="Seasonal SIR model with B=0.011, R0=0.98, R0=1.20 and R0=1.70",cex.main=2)

#1(
ymin = 0.9*min(vS/vnpop)
plot(vtime,vS/vnpop,type="l",xlab="time",
ylab="fraction susceptible",ylim=c(ymin,1),lwd=3,main=substitute(paste("Susceptible, R0=",R0),list(R0=R0)))
iind = which.min(abs(vS/vnpop-1/R0)) # find the index at which S/N is equal to 1/R0
legend("bottomright",legend=c("total susceptibles"),bty="n",lwd=3,col=c(1),cex=1.0)
#)


#2(
plot(vtime,vI/npop,type="l",xlab="time",
ylab="fraction_infected",ylim=c(0,0.03),lwd=3,col=4,main=substitute(paste("Infective,_R0=",R0),list(R0=R0)))
n=length(vtime)
lines(vtime[2:n],-diff(vS)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend(11.5,0.21,legend=c("total_infected_(prevalence)","newly_infected/day_(incidence)"),bty="n",lwd=3,col=c(4,2))
#)


#3(
plot(vtime,log(vI/vnpop),type="l",xlab="time",
ylab="log(fraction_infected)",lwd=3,col=4,main=substitute(paste("Log_(Infected),_R0=",R0),list(R0=R0)))
legend("topright",legend=c("log(Infected)"),bty="n",lwd=3,col=c(4),cex=1)
#)


#4(
plot(vtime,vR/npop,type="l",xlab="time",
ylab="fraction_recovered",ylim=c(0,2),lwd=3,col=1,main=substitute(paste("Recovery,_R0=",R0),list(R0=R0)))
lines(vtime[2:n],diff(vR)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend("topright",legend=c("total_recovery","newly_recoverd/day_(incidence)"),bty="n",lwd=3,col=c(1,2),cex=1)
#)

##########################################################################################################

#npop = 10000
npop = 1000
I_0 = 10
#I_0 = 500
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 1250
vt = seq(tbegin,tend,1)

gamma = 1/3#0.3
R0     = 1.2#1.48
beta0  = R0*gamma
B = 0.011
mu = 0.01
nu = 0.01
phi   = 0
epsilon1 = 0.3

vparameters = c(gamma=gamma,beta0=beta0,phi=phi,epsilon=epsilon1,mu=mu,B=B)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func_with_demographics, vparameters))

vS = solved_model$S
vI = solved_model$I
vR = solved_model$R
vtime = solved_model$time
vnpop = vS+vI+vR


#1(
ymin = 0.9*min(vS/vnpop)
plot(vtime,vS/vnpop,type="l",xlab="time",
ylab="fraction_susceptible",ylim=c(ymin,1),lwd=3,main=substitute(paste("Susceptible,_R0=",R0),list(R0=R0)))
iind = which.min(abs(vS/vnpop-1/R0)) # find the index at which S/N is equal to 1/R0
legend("bottomright",legend=c("total_susceptibles"),bty="n",lwd=3,col=c(1),cex=1.0)
#)


#2(
plot(vtime,vI/npop,type="l",xlab="time",
ylab="fraction_infected",ylim=c(0,0.25),lwd=3,col=4,main=substitute(paste("Infective,_R0=",R0),list(R0=R0)))
n=length(vtime)
lines(vtime[2:n],-diff(vS)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend("topright",legend=c("total_infected_(prevalence)","newly_infected/day_(incidence)"),bty="n",lwd=3,col=c(4,2))
#)


#3(
plot(vtime,log(vI/vnpop),type="l",xlab="time",
ylab="log(fraction_infected)",lwd=3,col=4,main=substitute(paste("Log_(Infected),_R0=",R0),list(R0=R0)))
legend(650,-18.8,legend=c("log(Infected)"),bty="n",lwd=3,col=c(4),cex=1)
#)


#4(
plot(vtime,vR/npop,type="l",xlab="time",
ylab="fraction_recovered",ylim=c(0,2),lwd=3,col=1,main=substitute(paste("Recovery,_R0=",R0),list(R0=R0)))
lines(vtime[2:n],diff(vR)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend("topright",legend=c("total_recovery","newly_recoverd/day_(incidence)"),bty="n",lwd=3,col=c(1,2),cex=1)
#)

##################################################################################################################


npop = 1000
I_0 = 10
#I_0 = 500
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 1250
vt = seq(tbegin,tend,1)

gamma = 1/3#0.3
R0     = 1.7#1.48
beta0  = R0*gamma
B = 0.011
mu = 0.01
nu = 0.01
phi   = 0
epsilon1 = 0.3


vparameters = c(gamma=gamma,beta0=beta0,phi=phi,epsilon=epsilon1,mu=mu)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func_with_demographics, vparameters))

vS = solved_model$S
vI = solved_model$I
vR = solved_model$R
vtime = solved_model$time
vnpop = vS+vI+vR

#1(
ymin = 0.9*min(vS/vnpop)
plot(vtime,vS/vnpop,type="l",xlab="time",
ylab="fraction_susceptible",ylim=c(ymin,1),lwd=3,main=substitute(paste("Susceptible,_R0=",R0),list(R0=R0)))
iind = which.min(abs(vS/vnpop-1/R0)) # find the index at which S/N is equal to 1/R0
legend("bottomright",legend=c("total_susceptibles"),bty="n",lwd=3,col=c(1),cex=1.0)
#)



#2(
plot(vtime,vI/npop,type="l",xlab="time",
ylab="fraction␣infected",ylim=c(0,0.22),lwd=3,col=4,main=substitute(paste("Infective,␣R0=",R0),list(R0=R0)))
n=length(vtime)
lines(vtime[2:n],-diff(vS)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend("topright",
legend=c("total␣infected␣(prevalence)","newly␣infected/day␣(incidence)"),bty="n",lwd=3,col=c(4,2))
#)


#3(
plot(vtime,log(vI/vnpop),type="l",xlab="time",
ylab="log(fraction␣infected)",lwd=3,col=4,main=substitute(paste("Log␣(Infected),␣R0=",R0),list(R0=R0)))
legend("bottomright",legend=c("log(Infected)"),bty="n",lwd=3,col=c(4),cex=1)
#)


#4(
plot(vtime,vR/npop,type="l",xlab="time",
ylab="fraction␣recovered",ylim=c(0,2),lwd=3,col=1,main=substitute(paste("Recovery,␣R0=",R0),list(R0=R0)))
lines(vtime[2:n],diff(vR)/(diff(vtime)*vnpop[1:(n-1)]),type="l",lwd=3,col=2)
legend("topright",
legend=c("total␣recovery","newly␣recoverd/day␣(incidence)"),bty="n",lwd=3,col=c(1,2),cex=1)
#)
```

B.5 Comparisons: Using MCMC for parameter estimation

```
##################################################################################
##################################################################################
##################################################################################
SIRfunc=function(t, x, vparameters){
  S = x[1]  # the value of S at time t
  I = x[2]  # the value of I at time t
  R = x[3]  # the value of R at time t
  if (I<0) I=0 # this is a cross check to ensure that we always have sensical values of I

  with(as.list(vparameters),{
    npop = S+I+R   # the population size is always S+I+R because there are no births or deaths in the model
    dS = -beta*S*I/npop            # the derivative of S wrt time
    dI = +beta*S*I/npop - gamma*I  # the derivative of I wrt time
    dR = +gamma*I                  # the derivative of R wrt time
    out = c(dS,dI,dR)
    list(out)
  })
}

##################################################################################
##################################################################################
# Let's set up the model parameters, and some initial conditions at time t=0
##################################################################################
gamma = 0.3          # recovery period in days^{-1}
R0    = 1.48          # R0 of R of the disease
beta = gamma*R0

N = 100   # population size
I_0 = 10    # number intially infected people in the population
S_0 = N-I_0
R_0 = 0

vt = seq(0,30,1)
```

```
vparameters=c(gamma=gamma, beta=beta)
inits=c(S=S_0,I=I_0,R=R_0)
solved_model1 = as.data.frame(lsoda(inits, vt, SIRfunc, vparameters))


library(MultiBD)


loglik_sir <- function(param, data) {
alpha <- exp(param[1]) # Rates must be non-negative
beta <- exp(param[2])
# Set-up SIR model
drates1 <- function(a, b) { 0 }
brates2 <- function(a, b) { 0 }
drates2 <- function(a, b) { alpha * b }
trans12 <- function(a, b) { beta * a * b }
sum(sapply(1:(nrow(data) - 1), # Sum across all time steps k
function(k) {
log(
dbd_prob( # Compute the transition probability matrix
t = data$time[k + 1] - data$time[k], # Time increment
a0 = data$S[k], b0 = data$I[k], # From: S(t_k), I(t_k)
drates1, brates2, drates2, trans12,
a = data$S[k + 1], B = data$S[k] + data$I[k] - data$S[k + 1],
computeMode = 4, nblocks = 80 # Compute using 4 threads
)[1, data$I[k + 1] + 1] # To: S(t_(k+1)), I(t_(k+1))
)
}))
}

logprior <- function(param) {
log_alpha <- param[1]
log_beta <- param[2]
dnorm(log_alpha, mean = 0, sd = 100, log = TRUE) +
dnorm(log_beta, mean = 0, sd = 100, log = TRUE)
}

source("http://bioconductor.org/biocLite.R")
biocLite("graph")
biocLite("Rgraphviz")
#install.packages("MCMCpack", repos = 'http://cran.us.r-project.org')
library(MCMCpack)

(alpha0 <- 0.4)
(beta0 <- 0.35)

inter1=1000
post_sample <- MCMCmetrop1R(fun = function(param) { loglik_sir(param, solved_model1) + logprior(param) },
theta.init = log(c(alpha0, beta0)),
mcmc = inter1, burnin = 200)

#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
#The Metropolis acceptance rate was 0.xxxx
#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

mult.fig(mfrow=c(2,1))
plot(as.vector(post_sample[,1]), type = "l", xlab = "Iteration", ylab = expression(log(hat(beta)[1])))
plot(as.vector(post_sample[,2]), type = "l", xlab = "Iteration", ylab = expression(log(hat(gamma)[1])))



##################################################################################
#alpha0 <- 3.39
#beta0 <- 0.0212
```

```
#post_sample <- MCMCmetrop1R(fun = function(param) { loglik_sir(param, Eyam) + logprior(param) },
#theta.init = log(c(alpha0, beta0)),
#mcmc = 1000, burnin = 0)
################################

#alpha0 <- beta
#beta0 <- gamma

#post_sample <- MCMCmetrop1R(fun = function(param) { loglik_sir(param, solved_model2) + logprior(param) },
#theta.init = log(c(alpha0, beta0)),
#mcmc = 1000, burnin = 200)

#mult.fig(mfrow=c(3,4))
#plot(as.vector(post_sample[,1]), type = ”l”, xlab = ”Iteration”, ylab = expression(log(hat(beta)[2])))
#plot(as.vector(post_sample[,2]), type = ”l”, xlab = ”Iteration”, ylab = expression(log(hat(gamma)[2])))

#############################################################################################
#############################################################################################
#############################################################################################


#npop = 10000
npop = 1000
npop2 = npop
I_0 = 10
#I_0 = 500
R_0 = 0
S_0 = npop-I_0-R_0

tbegin = 0
tend   = 20
vt = seq(tbegin,tend,1)

gamma = 0.3#1/3
R0    = 1.48#0.98
beta  = R0*gamma
mu = 0.01
nu = 0.01



vparameters = c(gamma=gamma, beta=beta,mu=mu)
inits = c(S=S_0,I=I_0,R=R_0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func_with_demographics, vparameters))
Time = solved_model$time
##DS = ceiling(solved_model$S)
##DI = ceiling(solved_model$I)
##DR = ceiling(solved_model$R)
DS = floor(solved_model$S)
DI = floor(solved_model$I)
DR = floor(solved_model$R)
solved_model2 = data.frame(time=Time,S=DS,I=DI,R=DR)
solved_model2

vS = solved_model2$S
vI = solved_model2$I
vR = solved_model2$R
vtime = solved_model2$time
vnpop = vS+vI+vR



library(MultiBD)

loglik_sir <- function(param, data) {
```

```
alpha <- exp(param[1]) # Rates must be non-negative
beta <- exp(param[2])
# Set-up SIR model
drates1 <- function(a, b) { 0 }
brates2 <- function(a, b) { 0 }
drates2 <- function(a, b) { alpha * b }
trans12 <- function(a, b) { beta * a * b }
sum(sapply(1:(nrow(data) - 1), # Sum across all time steps k
function(k) {
log(
dbd_prob( # Compute the transition probability matrix
t = data$time[k + 1] - data$time[k], # Time increment
a0 = data$S[k], b0 = data$I[k], # From: S(t_k), I(t_k)
drates1, brates2, drates2, trans12,
a = data$S[k + 1], B = data$S[k] + data$I[k] - data$S[k + 1],
computeMode = 4, nblocks = 80 # Compute using 4 threads
)[1, data$I[k + 1] + 1] # To: S(t_(k+1)), I(t_(k+1))
)
}))
}

logprior <- function(param) {
log_alpha <- param[1]
log_beta <- param[2]
dnorm(log_alpha, mean = 0, sd = 100, log = TRUE) +
dnorm(log_beta, mean = 0, sd = 100, log = TRUE)
}

(alpha0 <- 0.4)
(beta0 <- 0.35)

inter2=1000
post_sample1 <- MCMCmetrop1R(fun = function(param) { loglik_sir(param, solved_model2) + logprior(param) },
theta.init = log(c(alpha0, beta0)),
mcmc = inter2, burnin = 200)


#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
#The Metropolis acceptance rate was 0.xxxx
#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@


mult.fig(mfrow=c(2,1))
plot(as.vector(post_sample1[,1]), type = "l", xlab = "Iteration", ylab = expression(log(hat(beta)[2])))
plot(as.vector(post_sample1[,2]), type = "l", xlab = "Iteration", ylab = expression(log(hat(gamma)[2])))

mult.fig(mfrow=c(4,1))
plot(as.vector(post_sample[,1]), type = "l", col = "red", xlab = "Iteration",
ylab = expression(log(hat(beta)[1])), main = paste("Population␣=",npop1, ",",inter1,"interations"))#alpha
plot(as.vector(post_sample[,2]), type = "l", col = "blue", xlab = "Iteration",
ylab = expression(log(hat(gamma)[1])),main = paste("Population␣=",npop1, ",",inter1,"interations"))
plot(as.vector(post_sample1[,1]), type = "l", col = "red", lty = 6, lwd = 1.9, xlab = "Iteration",
ylab = expression(log(hat(beta)[2])), main = paste("Population␣=",npop2, ",",inter2,"interations"))
plot(as.vector(post_sample1[,2]), type = "l", col = "blue", lty = 6, xlab = "Iteration",
ylab = expression(log(hat(gamma)[2])), main = paste("Population␣=",npop2, ",",inter2,"interations"))


library(ggplot2)
x = as.vector(post_sample[,1])
y = as.vector(post_sample[,2])
df <- data.frame(x, y)

plotA = ggplot(df,aes(x = x)) + geom_density() + labs(x = expression("log␣"(hat(beta)[1])),
y = "Probability␣density", title = paste("Population␣=",npop1, ",",inter1,"interations"))
plotB = ggplot(df,aes(x = y)) + geom_density() + labs(x = expression("log␣"(hat(gamma)[1])),
y = "Probability␣density", title = paste("Population␣=",npop1, ",",inter1,"interations"))
```

```
plotAB = ggplot(df,aes(x = x,y = y)) +
stat_density2d(aes(fill = ..level..), geom = "polygon", h = 0.3) +
scale_fill_gradient(low = "grey85", high = "grey35", guide = FALSE) +
xlab(expression(log(hat(beta)[1]))) +
ylab(expression(log(hat(gamma)[1]))) + ggtitle(paste("Population␣=",npop1, ",",inter1,"interations"))



library(ggplot2)
x = as.vector(post_sample1[,1])
y = as.vector(post_sample1[,2])
df2 <- data.frame(x, y)

plotC = ggplot(df2,aes(x)) + geom_density() + labs(x = expression("log␣"(hat(beta)[2])),
y = "Probability␣density", title = paste("Population␣=",npop2, ",",inter2,"interations"))
plotD = ggplot(df2,aes(y)) + geom_density() + labs(x = expression("log␣"(hat(gamma)[2])),
y = "Probability␣density", title = paste("Population␣=",npop2, ",",inter2,"interations"))

plotCD = ggplot(df2,aes(x = x,y = y)) +
stat_density2d(aes(fill = ..level..), geom = "polygon", h = 0.009) +
scale_fill_gradient(low = "grey85", high = "grey35", guide = FALSE) +
xlab(expression(log(hat(beta)[2]))) +
ylab(expression(log(hat(gamma)[2]))) + ggtitle(paste("Population␣=",npop2, ",",inter2,"interations"))

#install.packages("ggpubr")
library(ggpubr)
ggarrange(plotA, plotB, plotC, plotD, ncol = 1, nrow = 4)
ggarrange(plotAB, plotCD, ncol = 1, nrow = 2)


Quantile1 = quantile(exp(post_sample[,1]), probs = c(0.025,0.5,0.975))
#c(Mean = mean(exp(post_sample[,1])),Median = median(exp(post_sample[,1])),SD = sd(exp(post_sample[,1])))
summary(exp(post_sample[,1]))

Quantile2 = quantile(exp(post_sample[,2]), probs = c(0.025,0.5,0.975))
#c(Mean = mean(exp(post_sample[,2])),Median = median(exp(post_sample[,2])),SD = sd(exp(post_sample[,2])))
summary(exp(post_sample[,2]))


Quantile3 = quantile(exp(post_sample1[,1]), probs = c(0.025,0.5,0.975))
#c(Mean = mean(exp(post_sample1[,1])),Median = median(exp(post_sample1[,1])),SD = sd(exp(post_sample1[,1])))
summary(exp(post_sample1[,1]))

Quantile4 = quantile(exp(post_sample1[,2]), probs = c(0.025,0.5,0.975))
#c(Mean = mean(exp(post_sample1[,2])),Median = median(exp(post_sample1[,2])),SD = sd(exp(post_sample1[,2])))
summary(exp(post_sample1[,2]))


##################################################################################

## Define as a function
epi203 <- function(pars){

    ## Show parameters
    print(pars)

    ## Additional parameters
    times <- seq(from = 0, to = 60, by = 1)              # we want to run the model for 3000 time steps
    yinit <- c(Susc = npop1 - I_0, Infected = I_0, Recovered = 0) # this parameter sets the initial conditions

    ## below is the code for the actual model including the equations that you should recognize
    SIR_model <- function(times, yinit, pars){

    with(as.list(c(yinit,pars)), {
            dSusc      <- birth - beta*Infected*Susc/(Susc+Infected+Recovered)                     - death*Susc
            dInfected  <-         beta*Infected*Susc/(Susc+Infected+Recovered) - recovery*Infected - death*Infected
            dRecovered <-                              recovery*Infected - death*Recovered

            return(list(c(dSusc, dInfected, dRecovered)))})
    }

    ## run the ode solver for the function specified (function defined above is used)
    ## return the value of each compartment (Susc, Infected, Recovered) for each time step.
    results <- ode(func = SIR_model, times = times, y = yinit, parms = pars)
    results <- as.data.frame(results)

    ## Return result
    return(results)
}

## Define as a function
epi204 <- function(pars){

    ## Show parameters
    print(pars)

    ## Additional parameters
    times <- seq(from = 0, to = 60, by = 1)              # we want to run the model for 3000 time steps
    yinit <- c(Susc = npop2 - I_0, Infected = I_0, Recovered = 0) # this parameter sets the initial conditions

    ## below is the code for the actual model including the equations that you should recognize
    SIR_model <- function(times, yinit, pars){

    with(as.list(c(yinit,pars)), {

            dSusc      <- birth - beta*Infected*Susc/(Susc+Infected+Recovered)                     - death*Susc
            dInfected  <-         beta*Infected*Susc/(Susc+Infected+Recovered) - recovery*Infected - death*Infected
            dRecovered <-                              recovery*Infected - death*Recovered

            return(list(c(dSusc, dInfected, dRecovered)))})
    }

    ## run the ode solver for the function specified (function defined above is used)
    ## return the value of each compartment (Susc, Infected, Recovered) for each time step.
    results <- ode(func = SIR_model, times = times, y = yinit, parms = pars)
    results <- as.data.frame(results)

    ## Return result
    return(results)
}

##############################################################################

test.pars1 <- c(beta = 1.48*0.3, recovery = 0.3, death = 0, birth = 0)
results1   <- epi203(test.pars1)

test.pars2 <- c(beta = Quantile1[[1]], recovery = Quantile1[[3]], death = 0, birth = 0)
results2   <- epi203(test.pars2)
test.pars3 <- c(beta = Quantile2[[1]], recovery = Quantile2[[3]], death = 0, birth = 0)
results3   <- epi203(test.pars3)

test.pars1.1 <- c(beta = 1.48*0.3, recovery = 0.3, death = 0, birth = 0)
results1.1   <- epi204(test.pars1.1)

test.pars4 <- c(beta = Quantile3[[1]], recovery = Quantile4[[1]], death = 0, birth = 0)
results4   <- epi204(test.pars4)
test.pars5 <- c(beta = Quantile3[[3]], recovery = Quantile4[[3]], death = 0, birth = 0)
results5   <- epi204(test.pars5)
```

```
mult.fig(mfrow=c(2,1))
#-------------------------------------------------------
##First parameter estimation
plot(results1[, 1], results1[, 2], type="o", col="blue", pch="", lty=1, ylim=c(0,npop1+10),
       xlab = "Time", ylab = "Individuals",
       main = paste("SIR␣epidemics:␣Population␣=",npop1, ",",inter1,"interations"))
legend("topright", col=1:3, legend=c("S", "I", "R"), lwd=1)
lines(results1[, 1], results1[, 3], col="red", pch=1)
lines(results1[, 1], results1[, 4], col="green", pch=1)

#Stochastic------------------
lines(results2[, 1], results2[, 2], col="blue", lty=3)
lines(results2[, 1], results2[, 3], col="red",  lty=3)
lines(results2[, 1], results2[, 4], col="green",  lty=3)

lines(results3[, 1], results3[, 2], col="blue",  lty=3)
lines(results3[, 1], results3[, 3], col="red",  lty=3)
lines(results3[, 1], results3[, 4], col="green",  lty=3)
#-------------------------------------------------------


#-------------------------------------------------------
##Second parameter estimation
plot(results1.1[, 1], results1.1[, 2], type="o", col="blue", pch="", lty=1, ylim=c(0,npop2+10),
       xlab = "Time", ylab = "Individuals",
       main = paste("SIR␣epidemics:␣Population␣=",npop2, ",",inter2,"interations"))
legend("topright", col=1:3, legend=c("S", "I", "R"), lwd=1)
lines(results1.1[, 1], results1.1[, 3], col="red", pch=1)
lines(results1.1[, 1], results1.1[, 4], col="green", pch=1)

#Stochastic------------------
lines(results4[, 1], results4[, 2], col="blue", lty=4)
lines(results4[, 1], results4[, 3], col="red", lty=4)
lines(results4[, 1], results4[, 4], col="green", lty=4)

lines(results5[, 1], results5[, 2], col="blue", lty=4)
lines(results5[, 1], results5[, 3], col="red", lty=4)
lines(results5[, 1], results5[, 4], col="green", lty=4)
#-------------------------------------------------------
```

Appendix C

Chapter 5: R script

C.1 SIR function (MCMC)

:

```
rm(list = ls(all = TRUE))  # resets R to fresh

require(”sfsmisc”)
require(”deSolve”)

niter = 20
SIRfunc=function(t, x, vparameters){
    S = x[1]
    I = x[2]
    R = x[3]
    if (I<0) I=0

    with(as.list(vparameters),{
        npop = S+I+R
        dS = -beta*S*I/npop
        dI = +beta*S*I/npop - gamma*I
        dR = +gamma*I
        out = c(dS,dI,dR)
        list(out)
    })
}

gamma = 0.3
R0    = 1.48
beta = gamma*R0

N = 10000
I_0 = 10
S_0 = N-I_0
R_0 = 0

vt = seq(0,1000,1)
vparameters=c(gamma=gamma, beta=beta)
inits=c(S=S_0,I=I_0,R=R_0)
sirmodel = as.data.frame(lsoda(inits, vt, SIRfunc, vparameters))

zstate = list()
i = 1
```

```
for (iter in 1:niter){
  time = 0
  vstate = c(S_0,I_0,R_0)

  K = length(vstate)
  J = 2
  lambda = matrix(0,nrow=J,ncol=length(vstate))
  lambda[1,] = c(-1,1,0)
  lambda[2,] = c(0,-1,1)

  while(vstate[2]>0&vstate[1]>0){
    zstate[[i]] = c(vstate,time,iter)
    S = vstate[1]
    I = vstate[2]
    R = vstate[3]

    vec_p = c(beta*S*I/N
              ,gamma*I)

    delta_t = 1/sum(vec_p)

    vec_l = rpois(length(vec_p),vec_p*delta_t)
    vstate = vstate + vec_l%*%lambda

    vstate[vstate<0] = 0
    i = i+1
    time = time + delta_t
  }
  cat(”Doing␣realisation:”,iter,niter,”␣”,time,vstate,”\n”)
}


par(mfrow=c(2,2))
vS = sapply(zstate, ”[[”, 1)
vI = sapply(zstate, ”[[”, 2)
vR = sapply(zstate, ”[[”, 3)
vtime = sapply(zstate, ”[[”, 4)
viter = sapply(zstate, ”[[”, 5)

mult.fig(4)

for (iter in 1:niter){
  l = which(viter==iter)
  if (iter==1){
    plot(vtime[l],vS[l]/N,type=”l”,xlab=”Time”,col=4,cex.lab=1.2,
    ylab=”Fraction␣susceptible”,main=”Susceptible”,ylim=c(0,1),xlim=c(0,max(vtime)))
  }else{
    lines(vtime[l],vS[l]/N,type=”l”,col=4)
  }
}
lines(sirmodel$time,sirmodel$S/N,col=2,lwd=2)
legend(”topright”,legend=c(”Deterministic”,”MCMC”),col=c(2,4),lwd=4,bty=”o”,cex=1)

for (iter in 1:niter){
  l = which(viter==iter)
  if (iter==1){
    plot(vtime[l],vI[l]/N,xlab=”Time”,col=4,cex.lab=1.2,ylab=”Fraction␣infective”,main=”Infective”,ylim=c(0,1.1*max(
  }else{
    lines(vtime[l],vI[l]/N,type=”l”,col=4)
  }
}
lines(sirmodel$time,sirmodel$I/N,col=2,lwd=2)
legend(”topright”,legend=c(”Deterministic”,”MCMC”),col=c(2,4),lwd=4,bty=”o”,cex=1)


vfinal = numeric(0)
```

```
for (iter in 1:niter){
  l = which(viter==iter)
  vfinal = append(vfinal,max(vR[l]/N))
  if (iter==1){
    plot(vtime[l],vR[l]/N,xlab="Time",col=4,cex.lab=1.2,
    ylab="Fraction␣recovered",main="Recovery",ylim=c(0,1.0),type="l",xlim=c(0,max(vtime)))
  }else{
    lines(vtime[l],vR[l]/N,type="l",col=4)
  }
}
lines(sirmodel$time,sirmodel$R/N,col=2,lwd=2)
legend("topright",legend=c("Deterministic","MCMC"),col=c(2,4),lwd=4,bty="o",cex=1)

hist(vfinal,breaks=seq(0,1,0.05),xlab="Total␣number␣infected",main=paste("Final␣size␣and␣N=",N))
Rmax = max(sirmodel$R)/N
lines(c(Rmax,Rmax),c(0,1e6),col=2,lty=3,lwd=3)

cat("The␣expected␣probability␣of␣outbreak␣is␣=␣",1-(1/R0)^I_0,"\n")
```

Appendix D

Chapter 6: R script

D.1 SIR function (Case study: Dengue Vector-Host)

:

```
npop_h = 10
I_h0 = 1
S_h0 = npop_h - I_h0
R_h0 = 0

npop_v = 5
I_v0 = 2
S_v0 = npop_v - I_v0

beta_hv = 1.0
beta_vh = 0.9
b_hv = 1.0
b_vh = 1.0
C_hv = beta_hv*b_hv
C_vh = beta_vh*b_vh
mu_h = 0.1 #1/25000
mu_v = 0.1 #1/4
gamma_h = 0.1 #1/3
p = 0

R1_0=S_h0*(C_hv*I_v0)/(gamma_h+npop_h)

tbegin = 0
tend   = 30
vt = seq(tbegin,tend,1)

vparameters = c(beta_hv=beta_hv,
                beta_vh=beta_vh,
                b_hv=b_hv,
                b_vh=b_vh,
                C_hv=C_hv,
                C_vh=C_vh,
                mu_h=mu_h,
                mu_v=mu_v,
                gamma_h=gamma_h,
                p=p)
inits = c(S_h=S_h0,
              I_h=I_h0,
              R_h=R_h0,
              S_v=S_v0,
              I_v=I_v0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func_with_dengue, vparameters))

vS_h = solved_model$S_h
vI_h = solved_model$I_h
vR_h = solved_model$R_h
vS_v = solved_model$S_v
vI_v = solved_model$I_v
vtime = solved_model$time
vnpop_h = vS_h + vI_h + vR_h
vnpop_v = vS_v + vI_v

mult.fig(mfrow=c(2,2),main="SIR_Model_with_Vector",cex.main=2)

#1(
#ymin1 = 0.9*min(vS_h/vnpop_h)
plot(vtime,vS_h,type="l",col="yellow",panel.first = grid(col="darkgray")
,xlab="time_(in_days)",ylab="Number_of_population",ylim=c(0,12),lwd=3,
main=substitute(paste("Tot.human,_N_h=",npop_h,"_and
Tot.vector,_N_v=",npop_v),list(npop_h=npop_h,npop_v=npop_v)))

lines(vtime,vI_h,type="l",col="blue",lwd=3)
lines(vtime,vR_h,type="l",col="red",lwd=3)
lines(vtime,vS_v,type="l",col="green",lwd=3)
lines(vtime,vI_v,type="l",col="purple",lwd=3)
legend("topright",legend=c("Suceptible_human",
"Infected_human","Recovered_human","Suceptible_vector","Infected_vector"),bty="n",
lwd=3,col=c("yellow","blue","red","green","purple"),cex=1)

#2(
#ymin1 = 0.9*min(vS_h/vnpop_h)
plot(vtime,vS_h/vnpop_h,type="l",col="yellow",panel.
first = grid(col="darkgray"),xlab="time_(in_days)",ylab="Fraction_of_population",ylim=c(0,1),lwd=3,
main=substitute(paste("Tot.human,_N_h=",npop_h,"_and
Tot.vector,_N_v=",npop_v),list(npop_h=npop_h,npop_v=npop_v)))
lines(vtime,vI_h/vnpop_h,type="l",col="blue",lwd=3)
lines(vtime,vR_h/vnpop_h,type="l",col="red",lwd=3)
lines(vtime,vS_v/vnpop_v,type="l",col="green",lwd=3)
lines(vtime,vI_v/vnpop_v,type="l",col="purple",lwd=3)

##################################################################################################

npop_h = 500
I_h0 = 10
S_h0 = npop_h - I_h0
R_h0 = 0

npop_v = 100
I_v0 = 5
S_v0 = npop_v - I_v0

beta_hv = 1.0
beta_vh = 0.9
b_hv = 1.0
b_vh = 1.0
C_hv = beta_hv*b_hv
C_vh = beta_vh*b_vh
mu_h = 0.1 #1/25000
mu_v = 0.1 #1/4
gamma_h = 0.1 #1/3
p = 0
```

```
#(R0 = ((C_hv)/npop_h)*S_h0)/gamma_h
#(R_0 = R_0*I_h0)

#(R_0=(C_hv*C_vh)/(mu_v*(mu_h+gamma_h)))
R2_0=S_h0*(C_hv*I_v0)/(gamma_h+npop_h)

tbegin = 0
tend   = 30
vt = seq(tbegin,tend,1)

vparameters = c(beta_hv=beta_hv,
                beta_vh=beta_vh,
                b_hv=b_hv,
                b_vh=b_vh,
                C_hv=C_hv,
                C_vh=C_vh,
                mu_h=mu_h,
                mu_v=mu_v,
                gamma_h=gamma_h,
                p=p)
inits = c(S_h=S_h0,
          I_h=I_h0,
          R_h=R_h0,
          S_v=S_v0,
          I_v=I_v0)

solved_model = as.data.frame(lsoda(inits, vt, derivative_calc_func_with_dengue, vparameters))

vS_h = solved_model$S_h
vI_h = solved_model$I_h
vR_h = solved_model$R_h
vS_v = solved_model$S_v
vI_v = solved_model$I_v
vtime = solved_model$time
vnpop_h = vS_h + vI_h + vR_h
vnpop_v = vS_v + vI_v


#1(
ymax1 = 1.1*max(vnpop_h)
plot(vtime,vS_h,type="l",col="yellow",panel.first = grid(col="darkgray"),xlab="time␣(in␣days)",
ylab="Number␣of␣population",ylim=c(0,ymax1),lwd=3,main=substitute(paste("Tot.human,␣N_h=",npop_h,"
and␣Tot.vector,␣N_v=",npop_v),list(npop_h=npop_h,npop_v=npop_v)))
lines(vtime,vI_h,type="l",col="blue",lwd=3)
lines(vtime,vR_h,type="l",col="red",lwd=3)
lines(vtime,vS_v,type="l",col="green",lwd=3)
lines(vtime,vI_v,type="l",col="purple",lwd=3)
legend("topright",legend=c("Suceptible␣human","Infected␣human","Recovered␣human","Suceptible␣vector",
"Infected␣vector"),bty="n",lwd=3,col=c("yellow","blue","red","green","purple"),cex=1)

#2(
#ymin1 = 0.9*min(vS_h/vnpop_h)
plot(vtime,vS_h/vnpop_h,type="l",col="yellow",panel.first = grid(col="darkgray"),
xlab="time␣(in␣days)",ylab="Fraction␣of␣population",ylim=c(0,1),lwd=3,
main=substitute(paste("Tot.human,␣N_h=",npop_h,"␣and␣Tot.vector,␣N_v=",npop_v),
list(npop_h=npop_h,npop_v=npop_v)))
lines(vtime,vI_h/vnpop_h,type="l",col="blue",lwd=3)
lines(vtime,vR_h/vnpop_h,type="l",col="red",lwd=3)
lines(vtime,vS_v/vnpop_v,type="l",col="green",lwd=3)
lines(vtime,vI_v/vnpop_v,type="l",col="purple",lwd=3)

c(R1_0,R2_0)
```