Bayesian Modelling of Epidemics

Part I: Intro to Epidemic Models & Bayesian Inference

Part II: Stochastic Population-level Models

Part III: Individual-level Models

Dr. Rob Deardon[1,2] **& Dr. Caitlin Ward**[3]

Department of Mathematics & Statistics, University of Calgary[1]

Faculty of Veterinary Medicine, University of Calgary[2]

Division of Biostatistics & Health Data Science, School of Public Health, University of Minnesota[3]

rob.deardon@ucalgary.ca

ward-c@umn.edu

ENAR, Baltimore, March 2024

Bayesian Modelling of Epidemics

Part I: Intro to Epidemic Models & Bayesian Inference

Dr. Rob Deardon[1,2] **& Dr. Caitlin Ward**[1]

Department of Mathematics & Statistics[1]
Faculty of Veterinary Medicine[2]
Department of Community Health Sciences[3]

rob.deardon@ucalgary.ca
ward-c@umn.edu

ENAR, Baltimore, March 2024

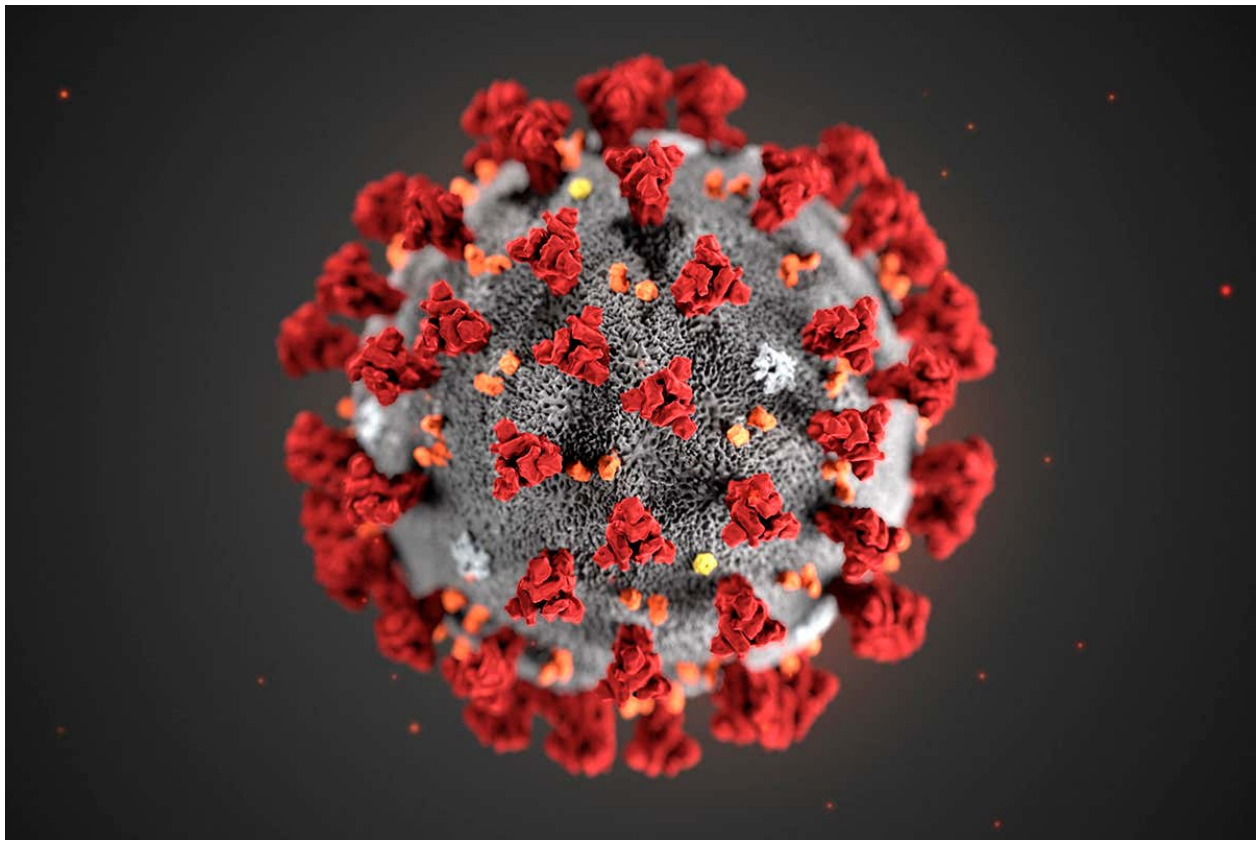
UNIVERSITY OF CALGARY

UNIVERSITY OF MINNESOTA

Outline

Outline

Example: COVID-19



CDC/SCIENCE PHOTO LIBRARY

Infectious (Communicable) Diseases

Examples:

- measles, AIDS, SARS, ebola, monkeypox (human)
- influenza (avian, swine, equine, human, etc.)
- foot-and-mouth disease (cloven-hooved animals)
- PRRS, PED, PHF (swine)
- citrus canker, tomato rust (crops)
- COVID-19

We may also be interested in modelling:

- invasive animal species
  (e.g. Asian giant hornet, emerald ash borer, mountain pine beetle)
- invasive plant species
  (e.g. giant hogweed, Japanese knotweed)
- fire spread
- internet / financial market contagion

Example: Asian giant hornet

https://www.cnn.com/2020/10/23/us/asian-giant-murder-hornet-nest-scn-trnd/index.html

Topic of Today's Workshop

Mechanistic models of disease transmission

These are models built to mimic underlying biological/ecological mechanisms associated with how transmission occurs

They are not designed with the goal of data analysis in mind
(although we will consider how to inform them via data)

Parameters are usually biologically / ecologically interpretable.

Parameters can be 'estimated' in a piecemeal fashion

Typically used to simulate and forecast epidemics
and the effect of control strategies upon those epidemics

Topic of Today's Wokshop

Many other **data-driven** techniques exist for analyzing infectious disease data

We might collectively termed these **disease surveillance** methods/models

For example:

- CUSUM / process control type methods
- spatial / spatio-temporal disease mapping
- (longitudinal) regression models

We will not be discussing disease surveillance methods today

Types of Transmission Model

Examples of model types:

1) **Homogenous models**

   any infectious individual has equal chances of infecting any susceptible individual

2) **Population Structured models**

   homogeneous mixing within strata (e.g. age-category)
   homogeneous mixing between strata
   but rates within and between different strata can vary

Types of Transmission Model

Examples of model types:

3) **Individual-level models**

chances of any particular infectious individual
infecting any particular susceptible individual
depends upon covariates about the individuals concerned

4) **Agent-based models**

“high dimensional, high complexity ILMs”

Important characteristic: Non-independence of infection events

Non-independence of infection events

As more individuals are infectious, the risk of infection for any remaining needs to go up (and vice versa)

This invalidates assumptions of most off-the-shelf statistical models, so we require something a bit more specialized

Outline

Classic Model: SIR Compartmental Model

W. Kermack & A. McKendrick (1927). A contribution to the mathematical theory of epidemics. Proc. R. Soc. London A 115, 700-721.

M. Keeling & P. Rohani (2007). Modelling Infectious Diseases In Humans and Animals. Princeton University Press

Here, we assume a **closed population** of n individuals

Time is considered **continuous** (not discrete)

Units of time whatever is most suitable to our disease
(e.g., hours, days, weeks, months, years, etc.)

Individuals are assumed to be **homogeneous** with respect to the disease

Individuals are assumed to **mix homogeneously**

Individuals are assumed to become infectious as soon as they are infected with the disease
(i.e., no **latent period**)

Classic Model: SIR Compartmental Model

We assume that at a given time point t,
each individual can be in one, and only one, of three states (or compartments):

S	Susceptible	doesn't have disease; can contract it
I	Infectious	has contracted the disease; can pass it on
R	Removed	been removed from the susceptible population e.g. died from the diease; e.g. isolated from the susceptible population; e.g. recovered and developed immunity

Notation:

S_t is the proportion of susceptible individuals in the population at time t
I_t is the proportion of infectious individuals in the population at time t
R_t is the proportion of removed individuals in the population at time t

Note: $S_t + I_t + R_t = 1$ for all t

Classic Model: SIR Compartmental Model

These assumptions lead us to a set of three ordinary differential equations for S_t, I_t, and R_t:

$$\frac{dS_t}{dt} = -\beta S_t I_t \tag{1}$$

$$\frac{dI_t}{dt} = \beta S_t I_t - \gamma I_t \tag{2}$$

$$\frac{dR_t}{dt} = \gamma I_t. \tag{3}$$

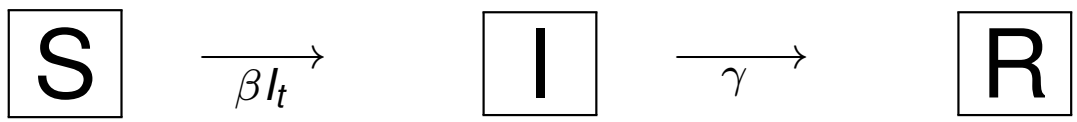
$\beta \geq 0$ is the **transmission rate** (rate at which individuals come together in a way disease can be transmiitted)

$\gamma \geq 0$ is the **removal rate** (e.g., rate of recovery with immunity, death, etc.)

Note: $\frac{1}{\gamma}$ defines the **infectious period** (the length of time individuals spend in the infectious state)

Classic Model: SIR Compartmental Model

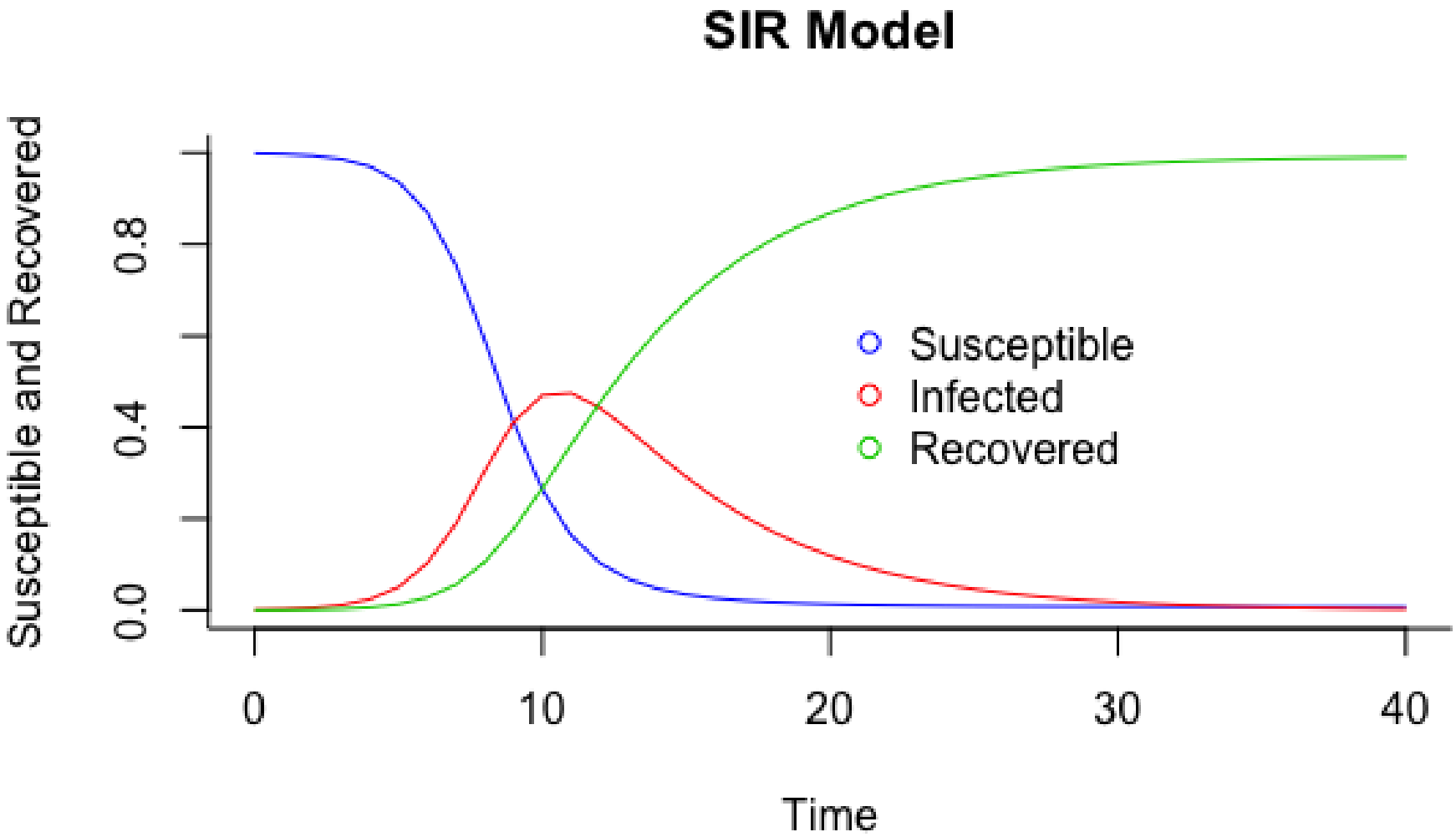
A flow diagram for the $\mathcal{SIR}$ model (at the individual level):



```
## Install and access deSolve package
install.packages("deSolve")
library(deSolve)



## Create an SIR function
SIR = function(time, state, parameters) {

  with(as.list(c(state, parameters)), {

    dS =  (-beta * S * I)
    dI =  (beta * S * I) - (gamma * I)
    dR =  gamma * I

    return(list(c(dS, dI, dR)))
  })
}
```

```
### Initialization

init       = c(S = 0.999, I = 0.001, R = 0.0)
parameters = c(beta = 1, gamma = 0.2)
times      = seq(0, 40, by = 1)



## Solve using ode (essentially produce output)
out = ode(y = init, times = times, func = SIR, parms = parameters)



## plot data
out = as.data.frame(out)
out$time = NULL #remove as I don't want this in plot

matplot(x = times, y = out, type = "l",
        xlab = "Time", ylab = "Susceptible and Recovered",
        main = "SIR Model", lwd = 1, lty = 1, bty = "l", col = c(4,2,3))

legend(20, 0.7, c("Susceptible", "Infected", "Recovered"),
         pch = 1, col = c(4,2,3), bty = "n")
```

Classic Model: SIR Compartmental Model



Outline

Basic Reproduction Number (R_0)

The **basic reproduction number** or R_0 is:

> the (expected) number of infections that result directly from a single infection in an otherwise wholly susceptible population.

Note: deterministic in a deterministic model (e.g. our SIR-ODE);
an average in a stochastic model

In our SIR-ODE model it is given by:

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

Basic Reproduction Number (R_0)

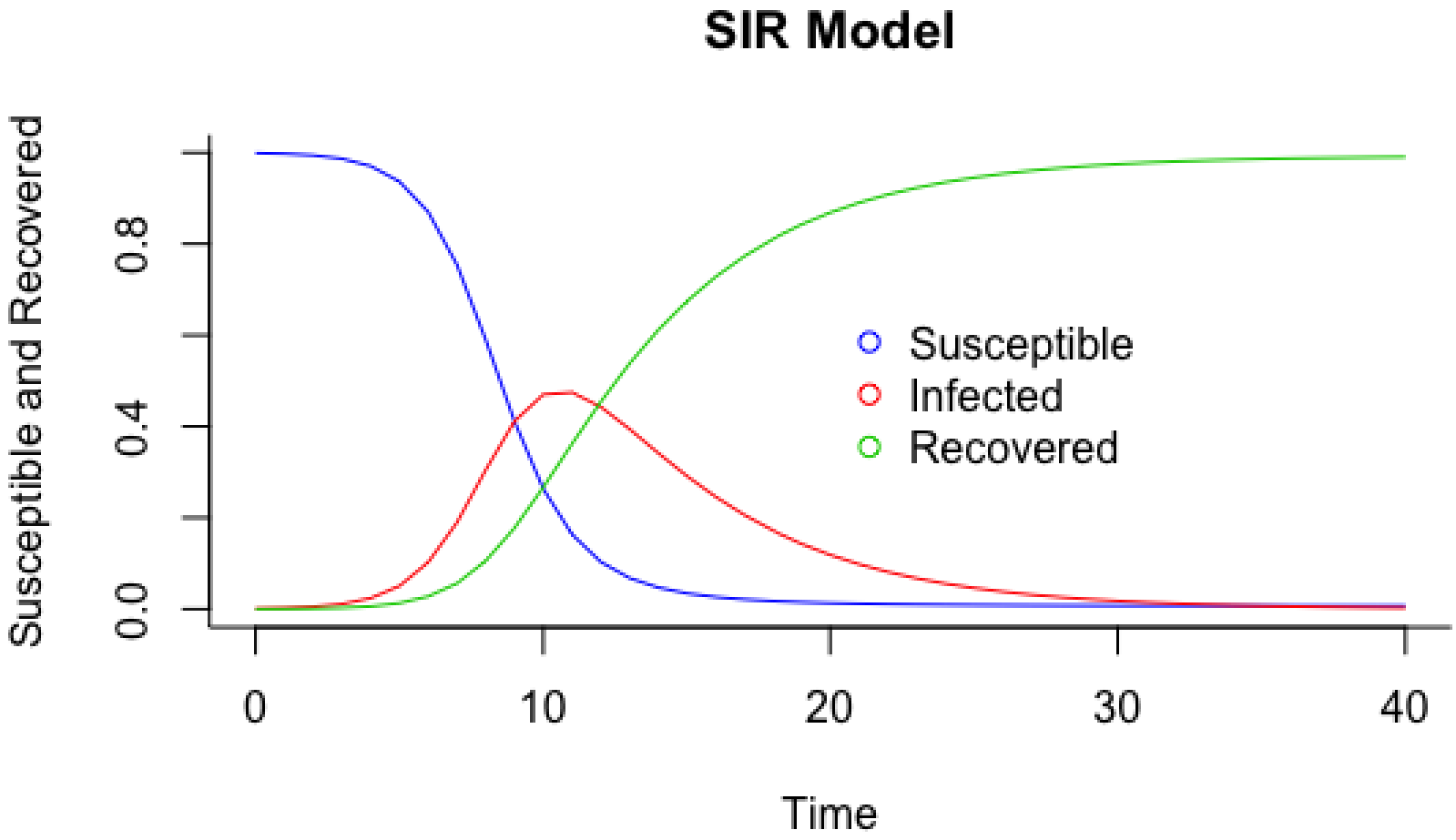
R_0 is mathematically and epidemiologically interesting due to the behaviour of epidemics in relation to the threshold $R_0 = 1$

Specifically, under a deterministic ODE model:

if $R_0 > 1$ an epidemic will take off
if $R_0 < 1$ an epidemic will die out

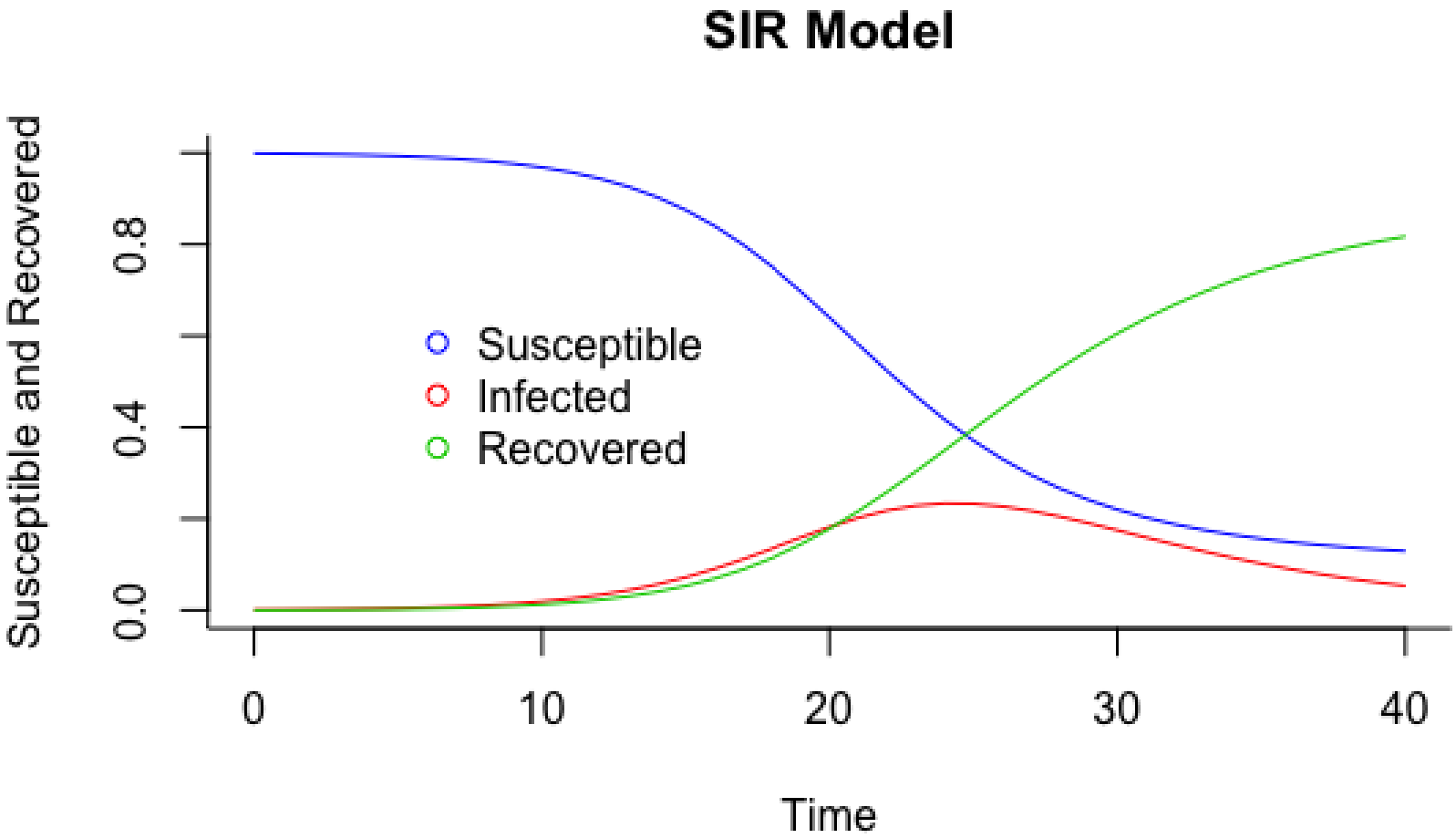
Basic Reproduction Number (R_0)

In our model $\beta = 1$ and $\gamma = 0.2$, so $R_0 = 5$:



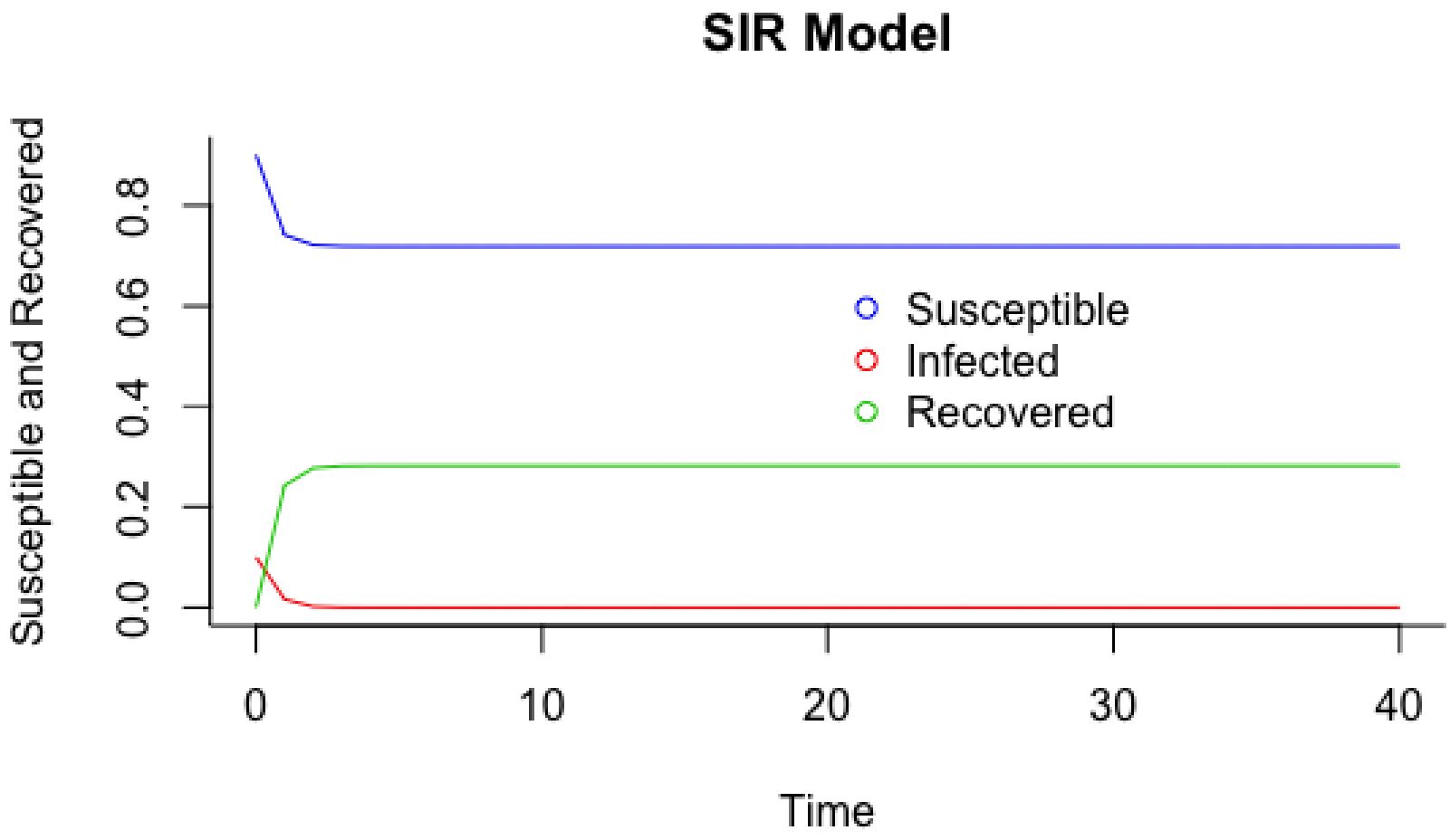
Basic Reproduction Number (R_0)

If we change our parameters to $\beta = 0.5$ and $\gamma = 0.2$, so $R_0 = 2.5$:



Basic Reproduction Number (R_0)

Now, set the parameters to $\beta = 4$ and $\gamma = 5$, so $R_0 = 0.8$
but also infect 10% of the population at time 0, rather than 0.1%:



Effective Reproduction Number (R_t)

Conceptually R_0 is concerned with what happens at the intialization of the epidemic in a population that is behaving in its "normal" state

Of course, as the number of susceptibles goes down the number of individuals which can be **directly infected** via each new infection goes down.

This is captured by the **effective reproduction number** or R_t:

the expected number of infections to result directly from a

new single infection in the population at time t.

It is given by:

$$\begin{aligned} R_t &= \frac{\beta}{\gamma} S_t \\ &= R_0 S_t \end{aligned}$$

And **changes in population behaviour** can effect the number of direct infections occurring from each new infection via changes in both β and γ. (e.g., social distancing, vaccination, quarantine, mask wearing, etc)

Outline

1. Frequency vs. Density Dependence

Our model here is known as a **frequency dependent** or **mass action** model.
This results from the fact that S_t, I_t and R_t are **proportions** of the population

So transmission dynamics don't depend on population size

An alternative approach is to use a **density dependent** model
In such a model, S_t, I_t and R_t are the **numbers** of the population within each respective state

Here, transmission dynamics do depend on population size,
or assuming a fixed area in which the population resides, population density

2. Other Possible Disease Timelines

A number of **compartmental frameworks** are possible:

- SI
- SIR
- SEIR
- SIS
- SIRS
- etc.

Which we choose will depend on the disease we are studying (and maybe the quality of our data...)

Example: SEIR Compartmental Model

These assumptions lead us to a set of four ordinary differential equations for S_t, E_t, I_t, and R_t:

$$\begin{aligned}
\frac{dS_t}{dt} &= -\beta S_t I_t \\
\frac{dE_t}{dt} &= \beta S_t I_t - \sigma E_t \\
\frac{dI_t}{dt} &= \sigma E_t - \gamma I_t \\
\frac{dR_t}{dt} &= \gamma I_t.
\end{aligned}$$

$\beta \geq 0$ is the transmission rate

$\sigma \geq 0$ is the "infectious rate"

$\gamma \geq 0$ is the removal rate

$\frac{1}{\sigma}$ defines the **latent period**

$\frac{1}{\gamma}$ defines the **infectious period**

3. Discrete versus continuous time

We have a choice between modelling our transmission process in either discrete or continuous time.

In **discrete time**, model generates counts / list of individuals infected each discrete time unit (event times are **INTEGERS**)

e.g., days, weeks, etc.

In **continuous time** model generates EXACT times of infection for individuals (event times are **REAL NUMBERS**)

e.g., a particular point in in time on a particular day, week, etc.

Outline

Statistical inference via the likelihood function

Parameter estimation typically done via the likelihood function

Likelihood function, $L(D|\boldsymbol{\theta})$ is the

- the probability of observing data, D
- given parameters, $\boldsymbol{\theta}$

Can maximize to get 'best fit' $\boldsymbol{\theta} \implies$ maximum likelihood estimation (MLE)

- then use sampling distribution to:
  - carry out statistical tests
  - construct confidence intervals

Or we combine it with a prior to get a posterior distribution for θ:

$\implies$ Bayesian inference

Bayesian inference

The **posterior** distribution is given by **Bayes' Theorem**

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{D}) = \frac{L(\boldsymbol{D} \mid \boldsymbol{\theta})\ Pr(\boldsymbol{\theta})}{Z(\boldsymbol{D})}$$

where:

$L(\boldsymbol{D}|\boldsymbol{\theta})$ is the **likelihood function**

$Pr(\boldsymbol{\theta})$ is the **prior distribution** of $\boldsymbol{\theta}$

$Z(\boldsymbol{D}) = \int L(\boldsymbol{D} \mid \boldsymbol{\theta}) Pr(\boldsymbol{\theta})\ d\boldsymbol{\theta}$ is a **normalization constant**

Priors

Priors are a distributional representation of our belief about the parameters **before** we have seen the data

We can choose any distribution which represents our belief
(e.g., normal, gamma, exponential, beta)

Can represent **expert knowledge** about the parameters expressed in the form of suitable statistical distributions.

Can be posterior from the analysis of **different data**

Priors

In some cases little or prior knowledge may be available and we can use

non-informative priors (e.g. Jeffrey's priors)
vague priors (e.g. normal distribution with large variance)

We can do the analysis with more than one type of prior to check sensitivity of the results to the prior.

Bayesian inference

Generally we cannot calculate the normalization constant

So we concern ourselves with “the posterior up to proportionality":

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

This “equation" is the essence of Bayesian parameter estimation:

- We start with a prior for our parameter(s) expressing our belief about the parameters before observing our data
- We update this prior belief with information in the data via the likelihood function to arrive at our posterior belief about the parameter(s)

Statistical inference via the likelihood function

Usually we use a Bayesian approach for fitting infectious disease models to data

There are a number of advantages:

1) Bayesian approach makes sense conceptually when the idea of a sampling distribution doesn't

e.g., modelling the chances of rain occurring today in Calgary
e.g., modelling an epidemic outbreak

2) Can incorporate missing/uncertain data easily (e.g. dates-of-infection)

3) Can incorporate info from different studies through prior

Outline

What's the problem?

The posterior summarizes all the information we have about our parameters and as such is a very useful object.

However, mathematically, even the simplest posterior can be very difficult to deal with.

This is due (in the first instance) to the problem of integrating to obtain the normalization constant:

$$Z(\boldsymbol{D}) = \int L(\boldsymbol{D} \mid \theta) Pr(\theta)\, d\theta$$

What's the solution?

However, if we can:

sample (i.e., generate random observations) from the posterior distribution,

then we can use the sampled sequence of observations $(\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(T)})$

to provide an approximation to the posterior.

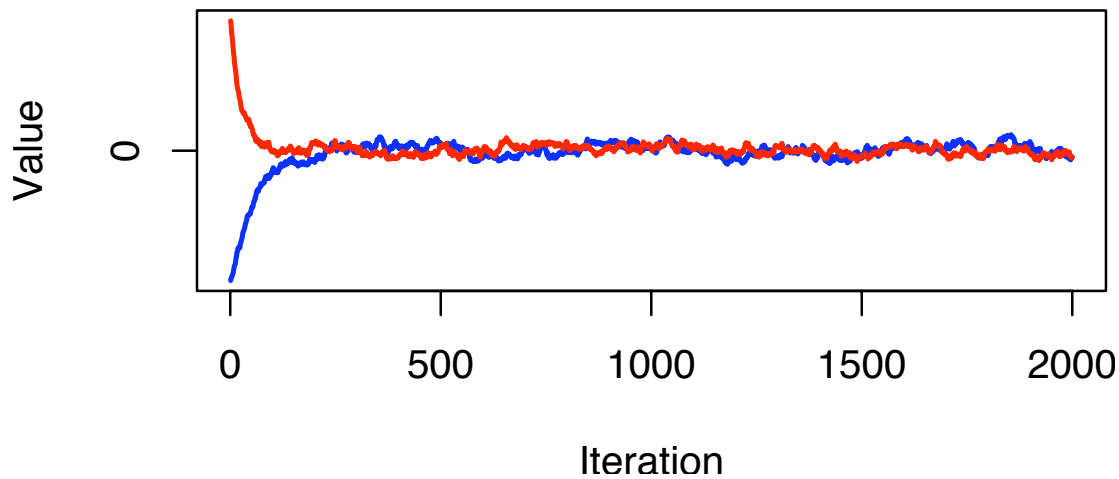
This is known as **Monte Carlo inference**.

Problem: how can we sample from the posterior?

One solution: use **Markov chain Monte Carlo (MCMC)**.

Markov chain Monte Carlo (MCMC)

Markov chain is a stochastic process where the value at each state of the chain depends only on the previous state.

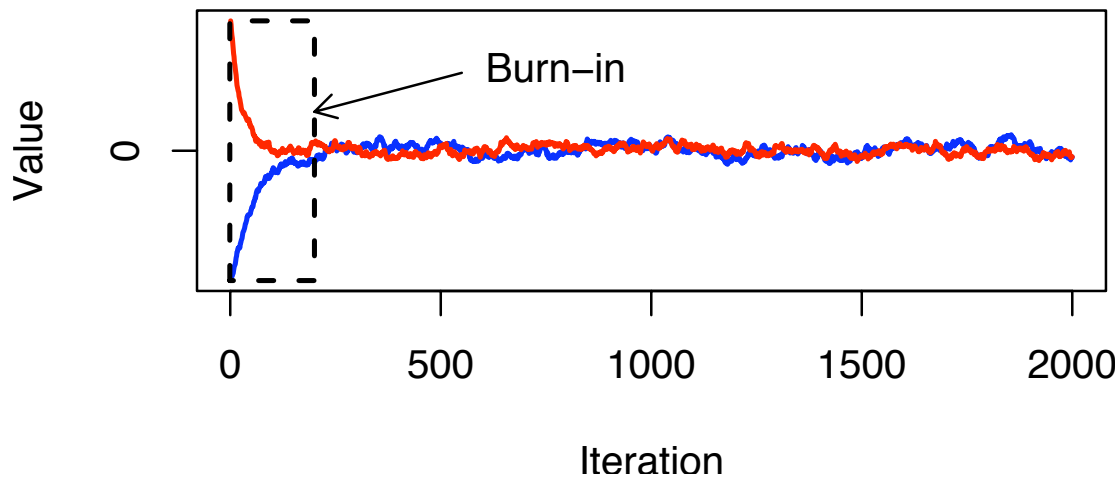


Under certain conditions a Markov chain will converge to its **stationary** distribution.

Markov chain Monte Carlo (MCMC)

Our aim is to set up a Markov chain whose stationary distribution is the posterior distribution we wish to sample from

Once converged, we can discard the **burn-in** and continue to run the chain until the required number of samples have been obtained.



We can use the **Metropolis Hasting** algorithm to do this

Metropolis-Hastings algorithm

The **Metropolis-Hastings** algorithm is a simple yet powerful way to produce a Markov chain whose **stationary distribution** is the **posterior** of interest.

We shall discuss a special case of this algorithm:

random-walk Metropolis Hastings or **Metropolis Algorithm**

We start by picking an arbitrary starting point for our chain, $\boldsymbol{\theta}^{(0)}$

Random-walk Metropolis-Hastings algorithm

At iteration i of the MCMC chain with current state $\theta^{(i)}$:

Generate $z \sim f$, where f is a distribution symmetric around zero (e.g. normal with mean zero)

Generate a candidate value for θ, $\theta' = \theta^{(i)} + z$

Calculate the acceptance probability α as:

$$\alpha = \min\left(1, \frac{\pi(\boldsymbol{\theta}' \mid \boldsymbol{D})}{\pi(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{D})}\right).$$

Random-walk Metropolis-Hastings algorithm

At iteration i of the MCMC chain with current state $\theta^{(i)}$:

Generate $z \sim f$, where f is a distribution symmetric around zero (e.g. normal with mean zero)

Generate a candidate value for θ, $\theta' = \theta^{(i)} + z$

Calculate the acceptance probability α as:

$$\alpha = \min\left(1, \frac{\frac{L(\boldsymbol{D}|\boldsymbol{\theta}')P(\boldsymbol{\theta}')}{\cancel{Z(\boldsymbol{D})}}}{\frac{L(\boldsymbol{D}|\boldsymbol{\theta}^{(i)})P(\boldsymbol{\theta}^{(i)})}{\cancel{Z(\boldsymbol{D})}}}\right).$$

Random-walk Metropolis-Hastings algorithm

At iteration i of the MCMC chain with current state $\theta^{(i)}$:

Generate $z \sim f$, where f is a distribution symmetric around zero (e.g. normal with mean zero)

Generate a candidate value for θ, $\theta' = \theta^{(i)} + z$

Calculate the acceptance probability α as:

$$\alpha = \min\left(1, \frac{L(\boldsymbol{D} \mid \boldsymbol{\theta}')Pr(\boldsymbol{\theta}')}{L(\boldsymbol{D} \mid \boldsymbol{\theta}^{(i)})Pr(\boldsymbol{\theta}^{(i)})}\right).$$

Random-walk Metropolis-Hastings algorithm

At iteration i of the MCMC chain with current state $\theta^{(i)}$:

Generate $z \sim f$, where f is a distribution symmetric around zero (e.g. normal with mean zero)

Generate a candidate value for θ, $\theta' = \theta^{(i)} + z$

Calculate the acceptance probability α as:

$$\alpha = \min\left(1, \frac{L(\boldsymbol{D} \mid \boldsymbol{\theta}')Pr(\boldsymbol{\theta}')}{L(\boldsymbol{D} \mid \boldsymbol{\theta}^{(i)})Pr(\boldsymbol{\theta}^{(i)})}\right).$$

Accept $\theta^{(i+1)} = \theta'$ with probability α
else reject and set $\theta^{(i+1)} = \theta^{(i)}$

Go to Step 1.

Random-walk Metropolis-Hastings algorithm

MCMC Notes

In theory, we can use any distribution f to generate our MCMC chains.

In practice, we generally have to choose an f
with suitable variance to explore the posterior distribution efficiently.

There are adaptive methods for "self tuning" our MCMC proposal distributions/updates.

When dealing with multidimensional posterior distributions, it turns out
we can update parameters one at a time rather than using one multidimensional update.

This turns out to be extremely powerful in higher dimensional problems.