

Weekly Oxford Worldwide

DEPARTMENT FOR
CONTINUING
EDUCATION



Infectious Disease Modelling: Applied Methods in R

Dr Emma L Davis

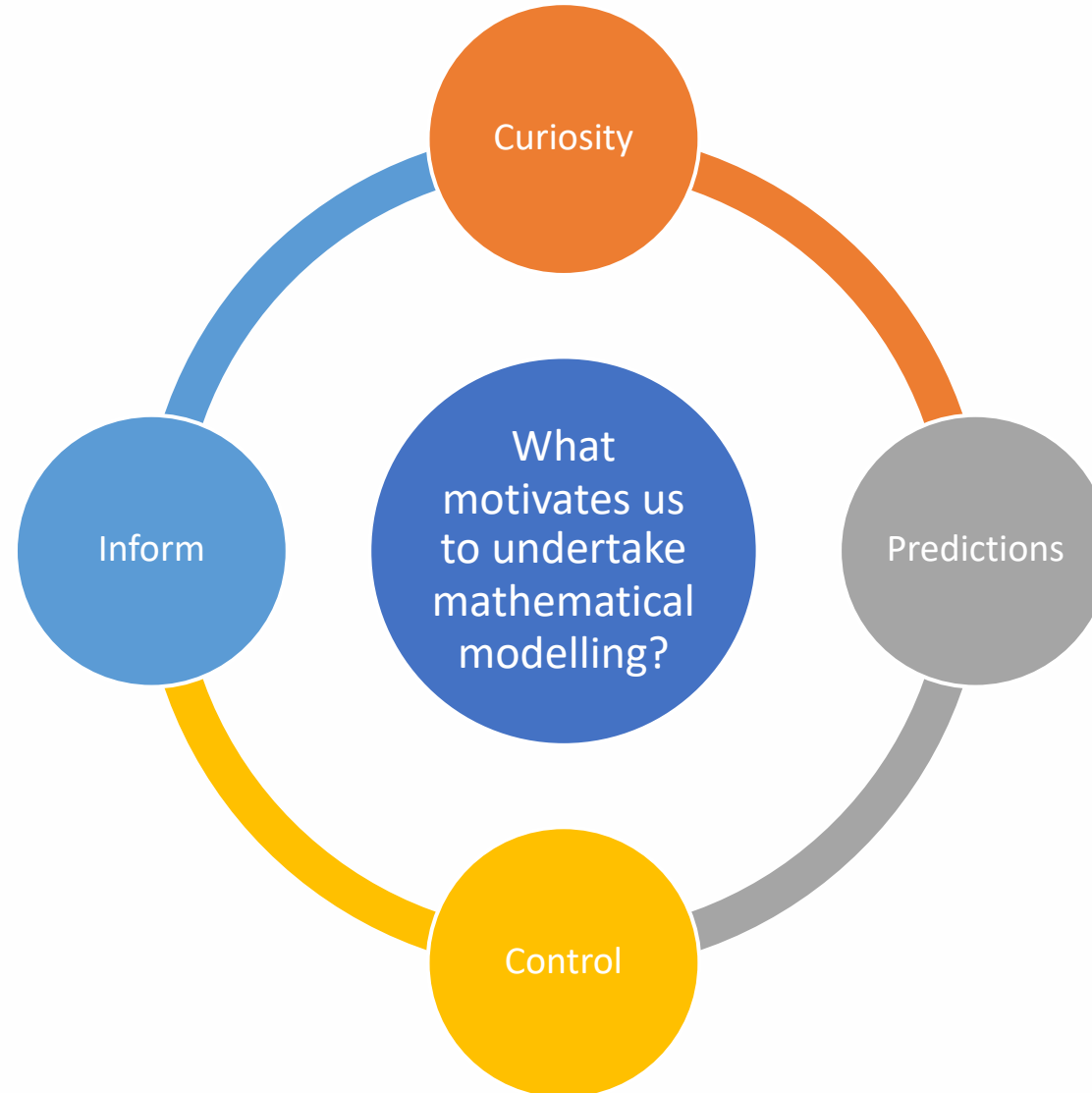
Week 1



Week 1: Introduction

- What is a model?
 - Example: COVID-19
 - Model formulation
- Using R and Rstudio
 - Installation
 - R markdown
 - Tidyverse
- Course overview

Motivation



What is a model?

Definition

A *model* is a simplified description, especially a mathematical one, of a system or process, to assist with calculations and predictions.

There are two main types of model:

1. Statistical models
2. Mechanistic models

What is a model?

Statistical models	Mechanistic models
Find patterns in existing data	Use underlying understanding of the system
Simple equations (e.g. linear models, regression)	Complex systems of equations
A lot of data needed	Minimal data required to observe qualitative behaviours of the system
Suitable for interpolation (between points, missing data)	Suitable for interpolation and extrapolation (prediction)*

What is a model?

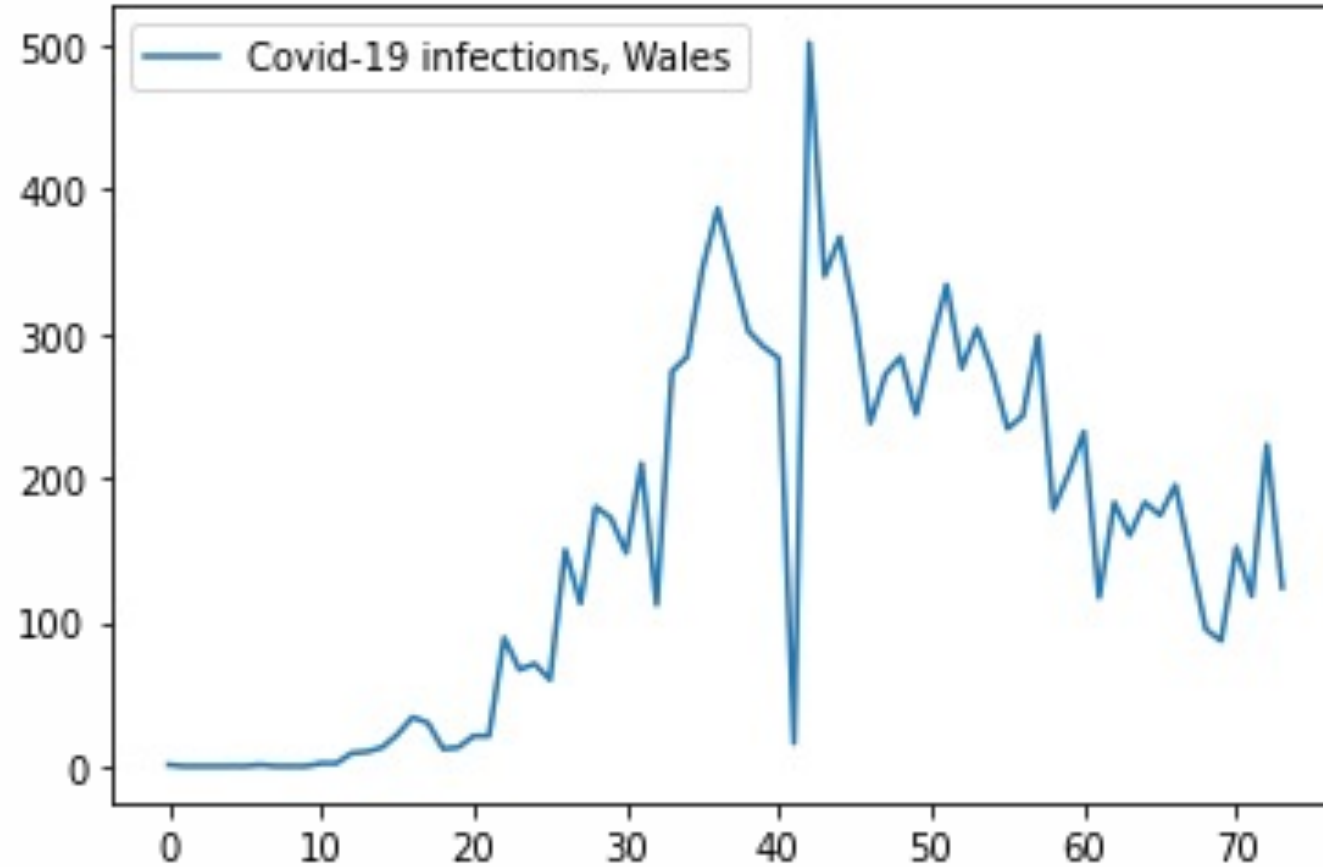
Types of mechanistic model:

1. A model is **deterministic** if its behaviour is entirely predictable
i.e. for a fixed set of initial conditions, the outcome will always be the same
2. A model is **stochastic** if it has random variables as inputs
the outputs of a stochastic model are random and will be different every time

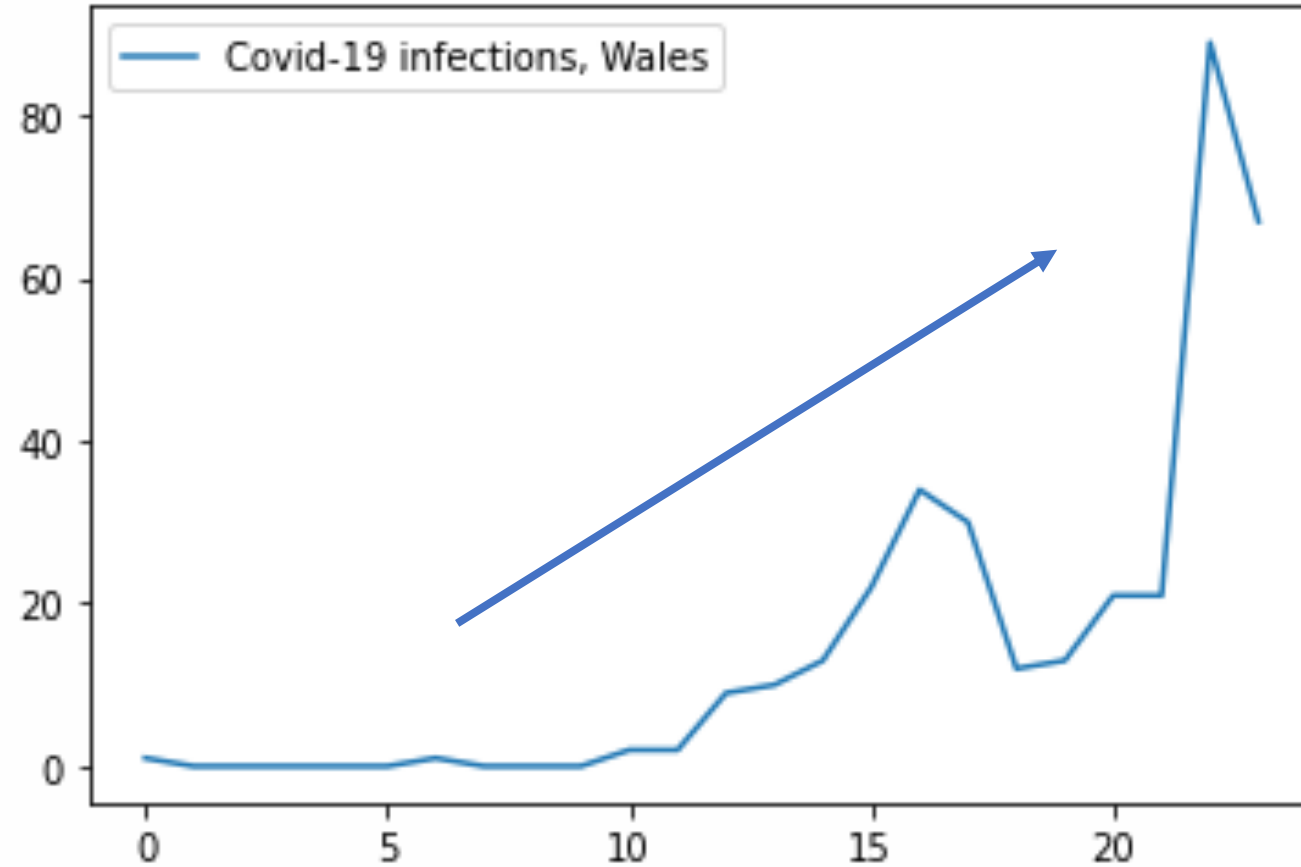
What is a model?

Deterministic models	Stochastic models
Model outcome is always the same	Model outcome is different every time
Output represents “mean” outcome	Output represents one possible outcome
Only need to run model once (less computation time)	Need to run model many times then look at summary statistics (more computation)
Hard to quantify uncertainty	Can quantify uncertainty using standard deviation or range of outcomes
More appropriate for large populations or high infection levels	More appropriate for small populations or low (close to zero) infection levels

Example: COVID-19

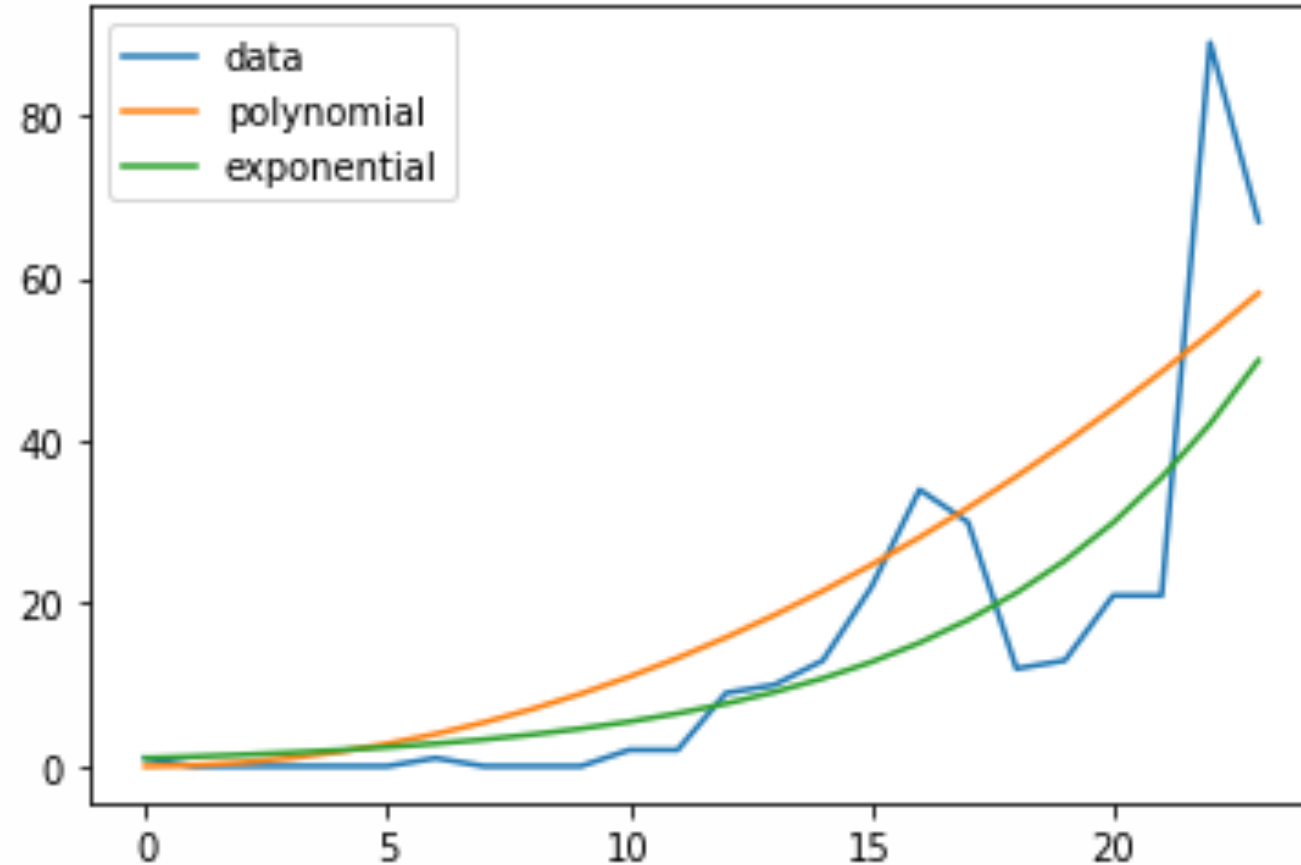


Example: COVID-19



Example: COVID-19

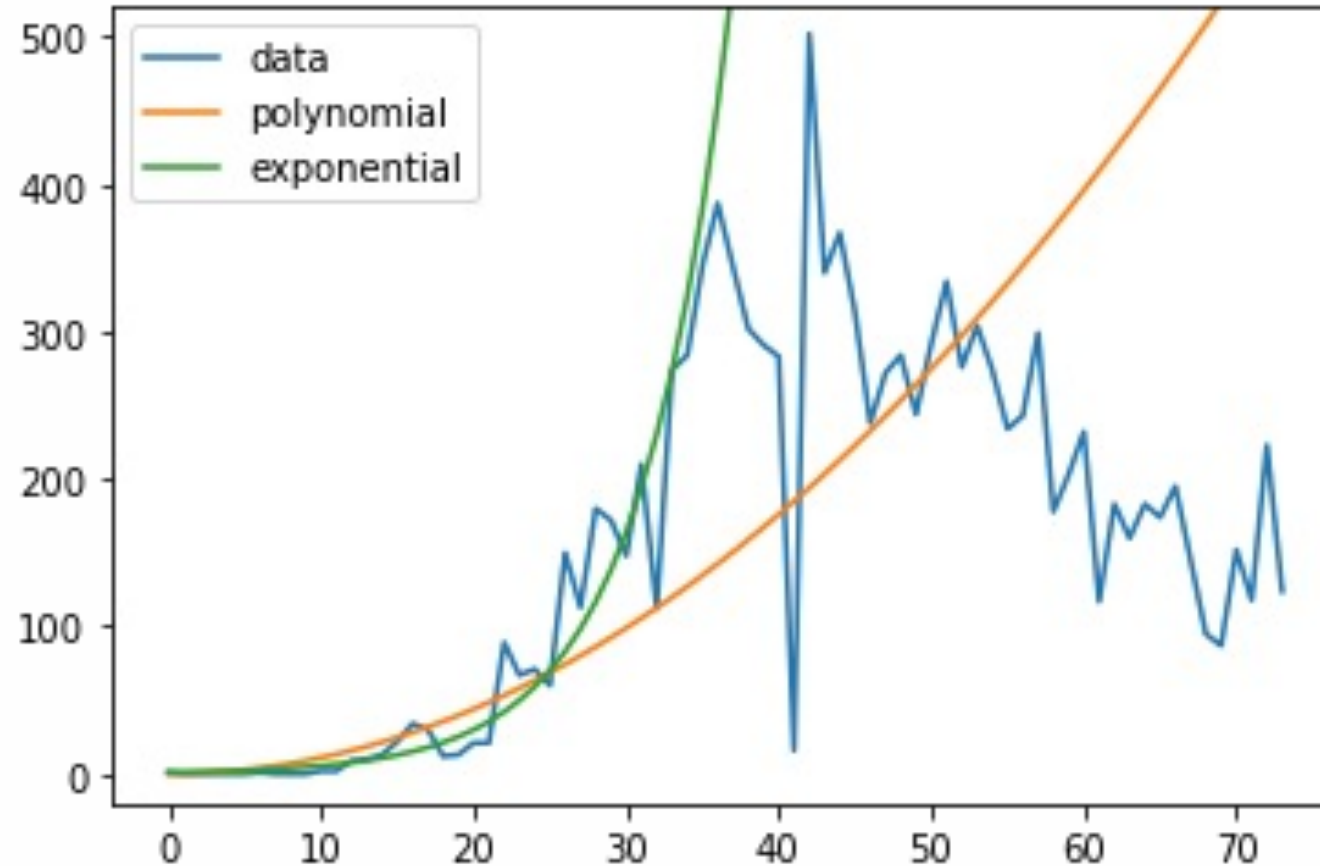
Try
polynomial
or
exponential



These are *statistical* models – just using the data, no other assumptions

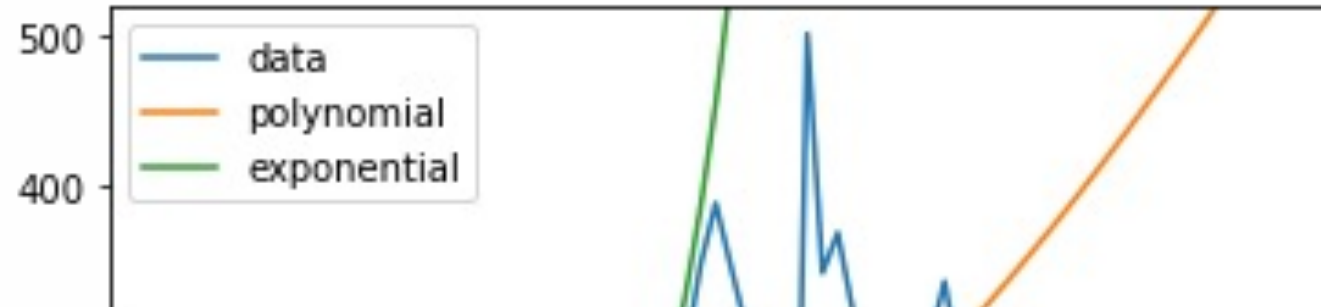
Example: COVID-19

Prediction:

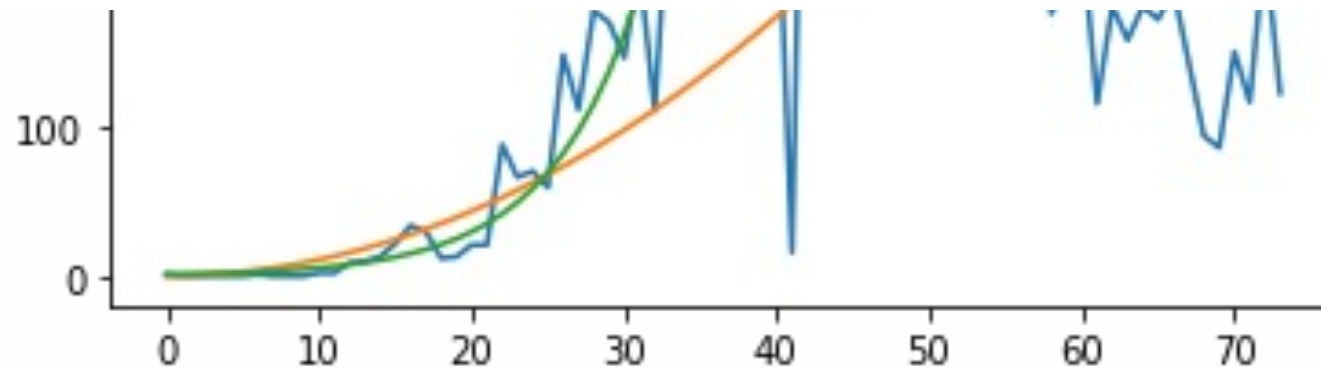


These are *statistical* models – just using the data, no other assumptions

Example: COVID-19



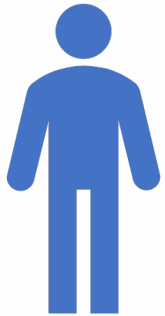
To do better we're going to need to combine the data with additional knowledge around how diseases spread through populations



Model formulation

We will focus on *compartmental models*, where we separate the population into “compartments”

e.g.



Susceptible
(to infection)



Exposed
(not infectious)

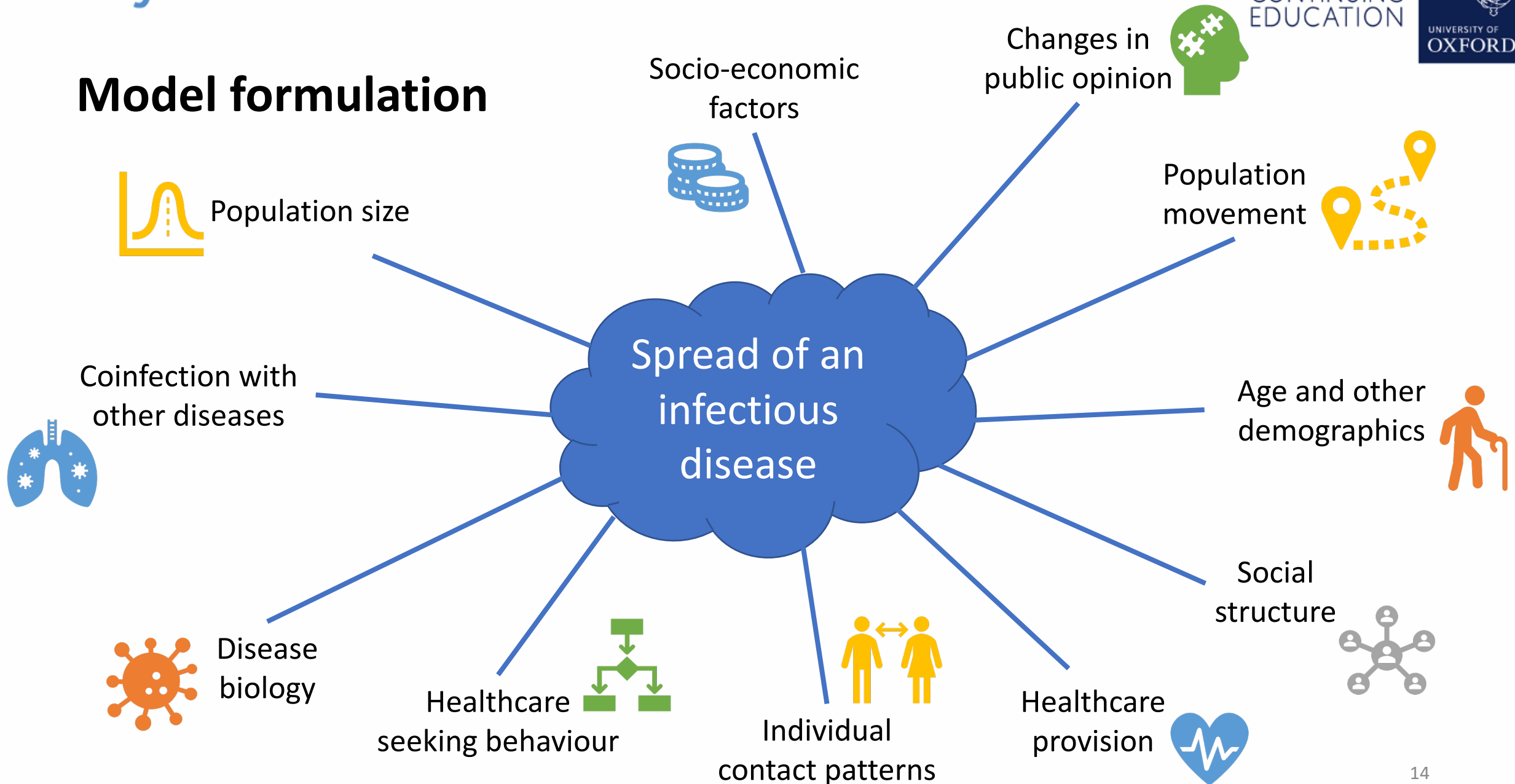


Infected
(and infectious)



Recovered
(or removed)

Model formulation



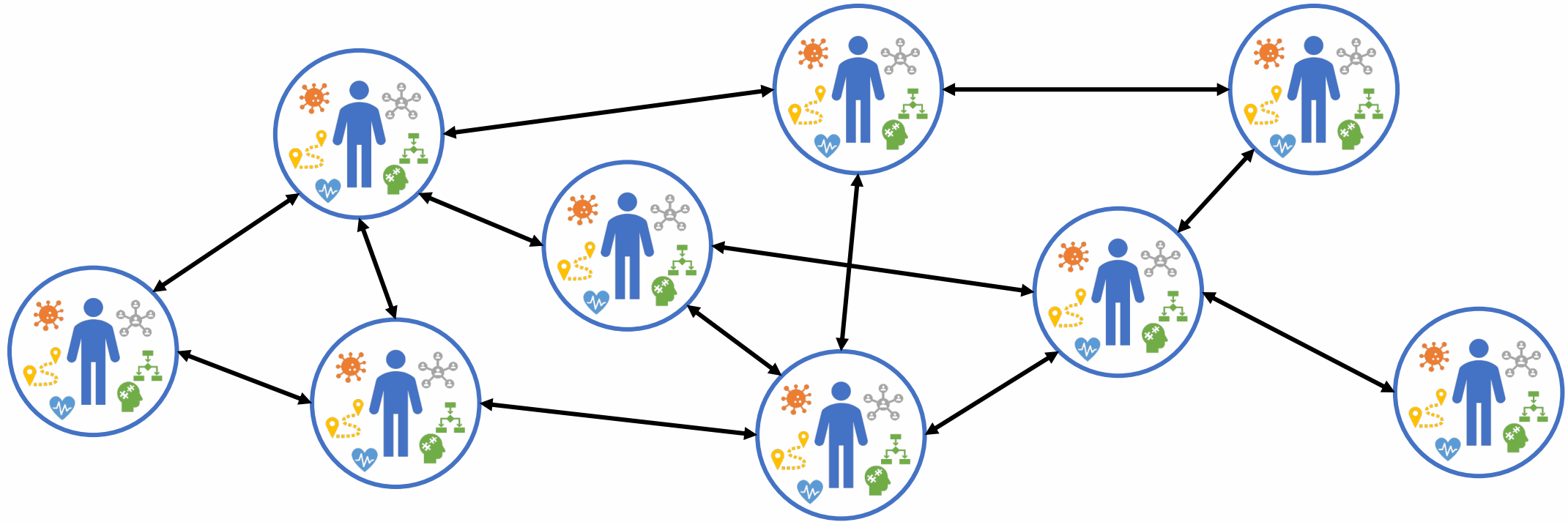
Model formulation

How can we model these systems when they have so many variables?



Model formulation

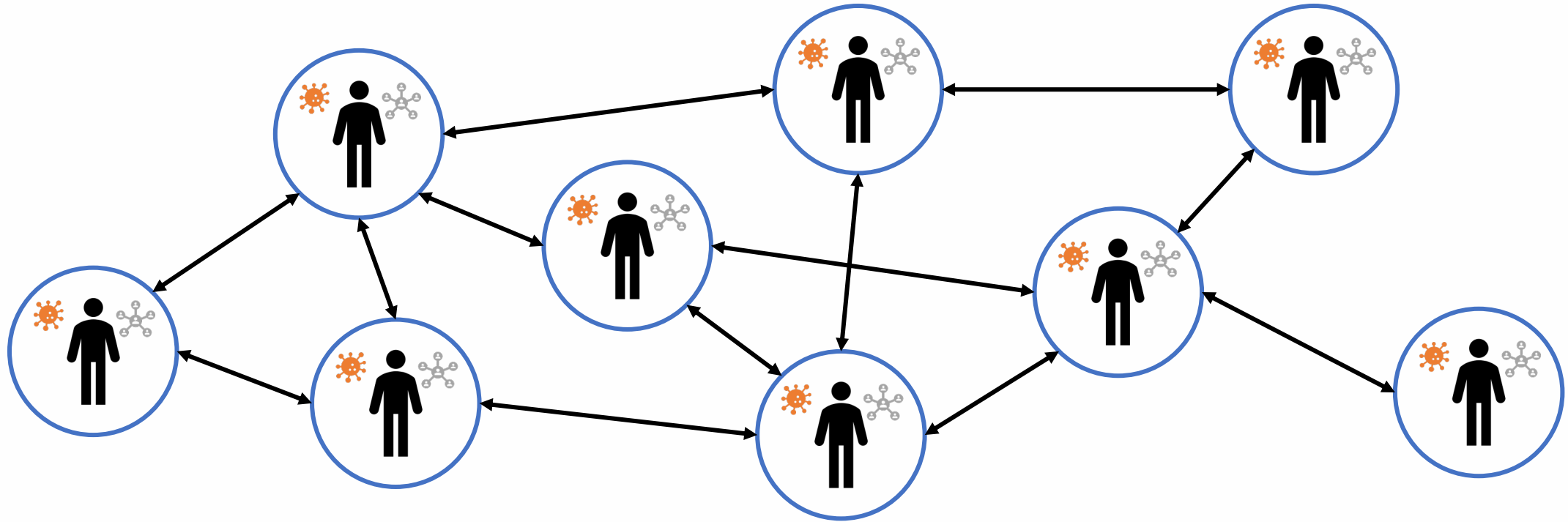
How can we model these systems when they have so many variables?



Idea: Stochastic model – run lots of times then look at summary statistics, e.g. mean?

Model formulation

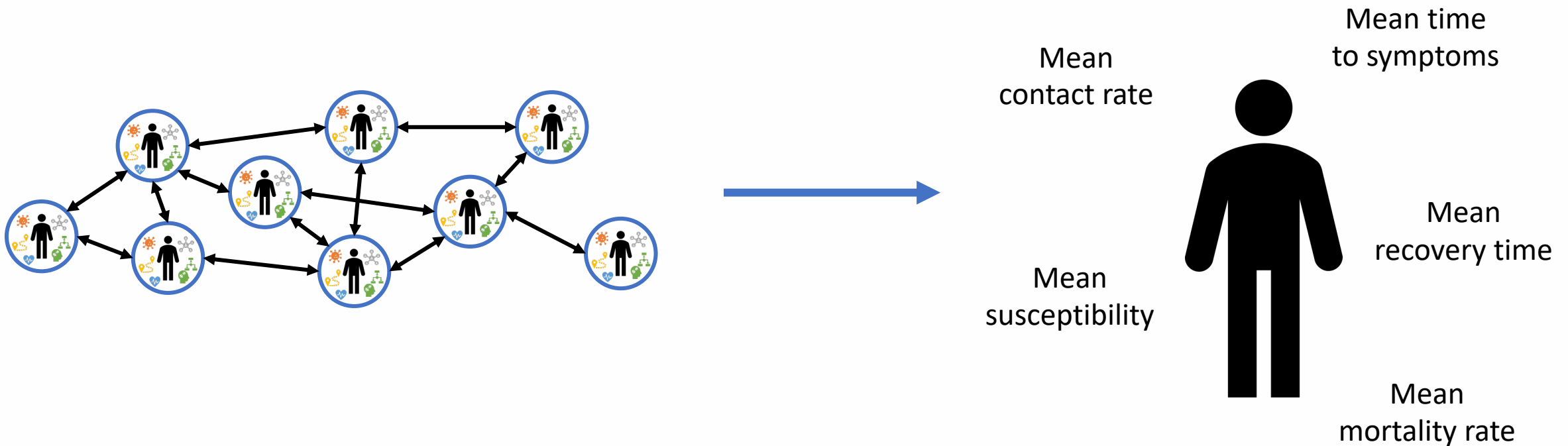
How can we model these systems when they have so many variables?



We can reduce complexity by removing variables / simplifying our assumptions

Model formulation

We can also reduce complexity by considering the average person and using a deterministic model



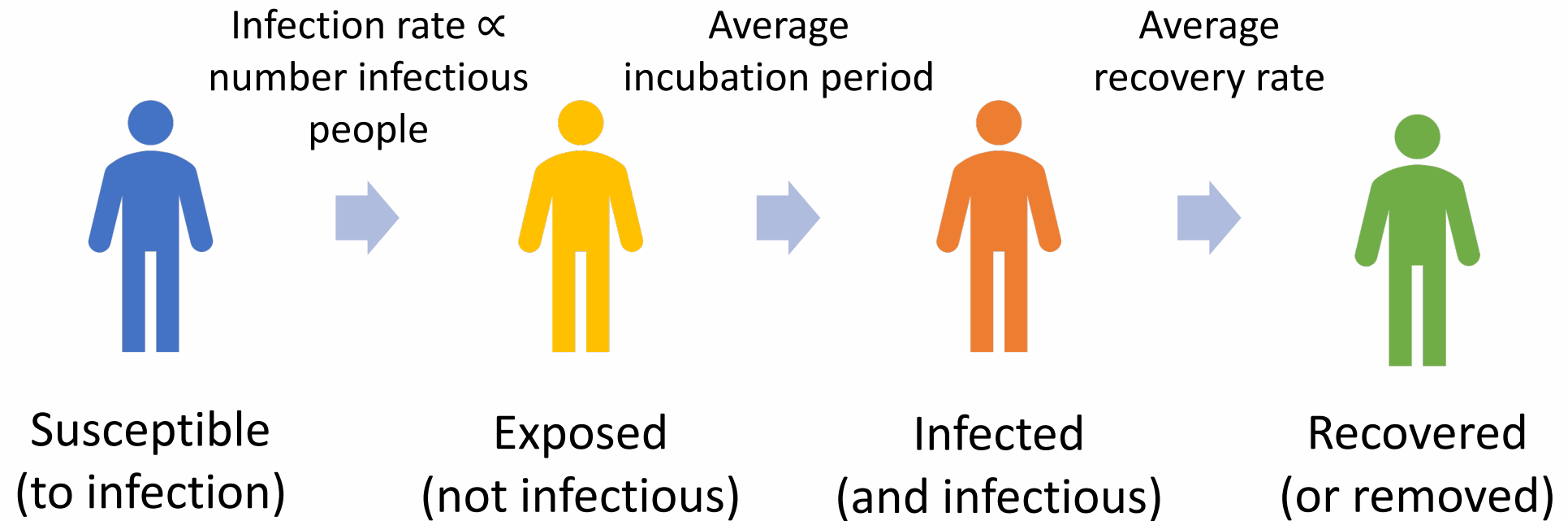
N-body system replaced by a 1-body system

Model formulation

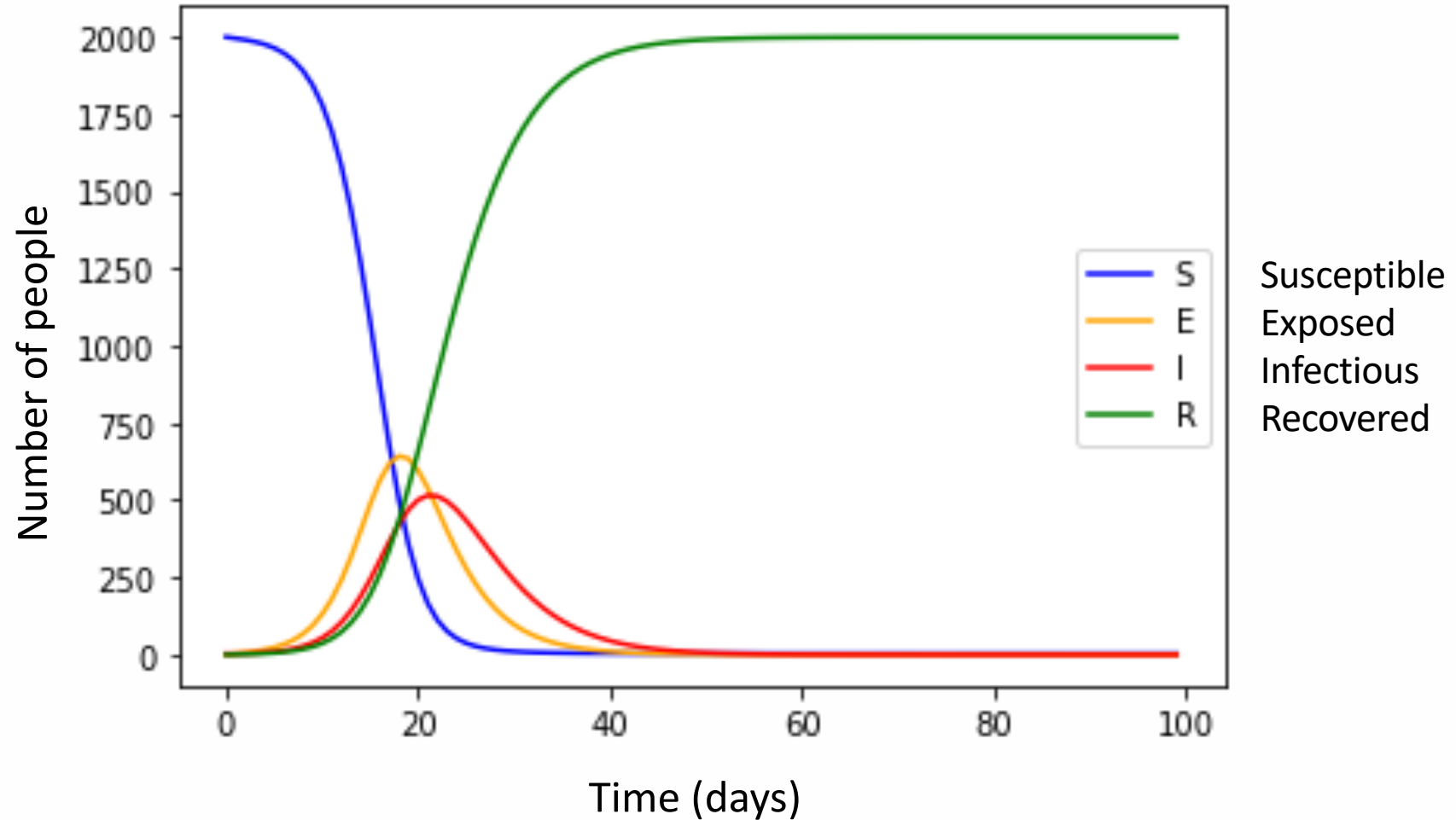
Deterministic model assumptions:

- Consider only the average person
- Homogeneous mixing (everyone just as likely to meet everyone else, i.e. no underlying social network structure)
- Fixed average rates of interaction between people
- Transmission rate / rate of contracting infection only dependent on number of currently infected people

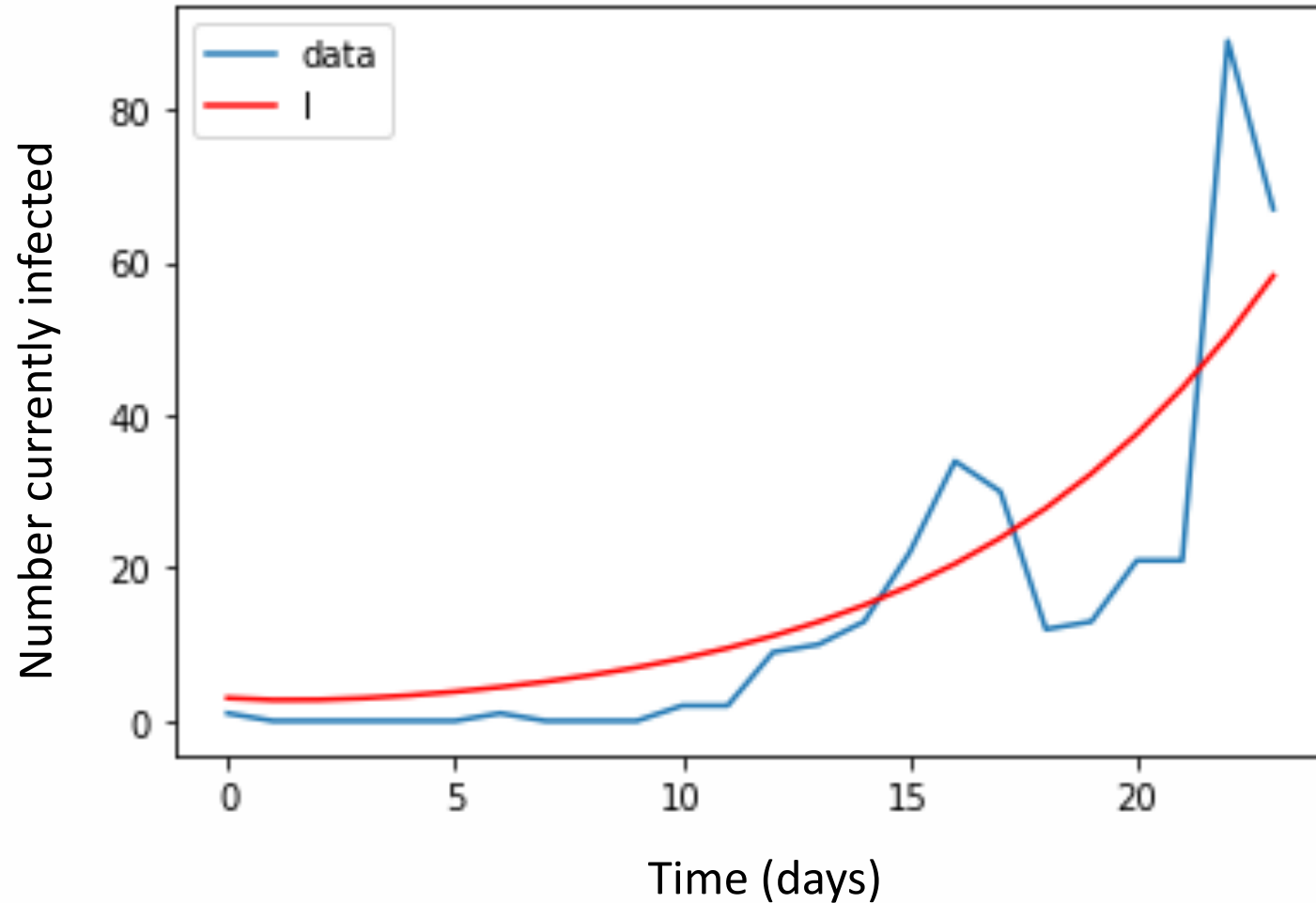
Model formulation



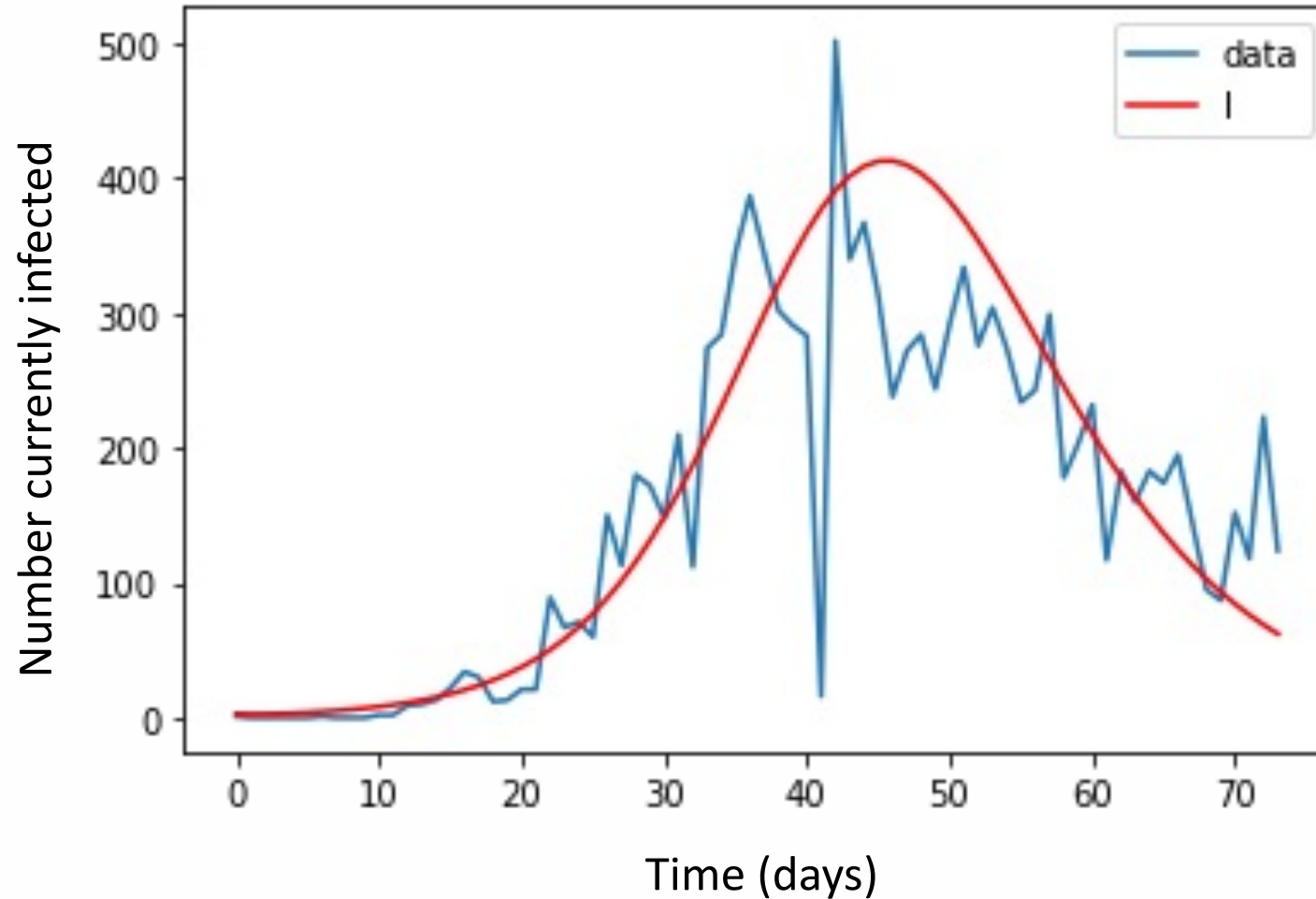
Example: COVID-19



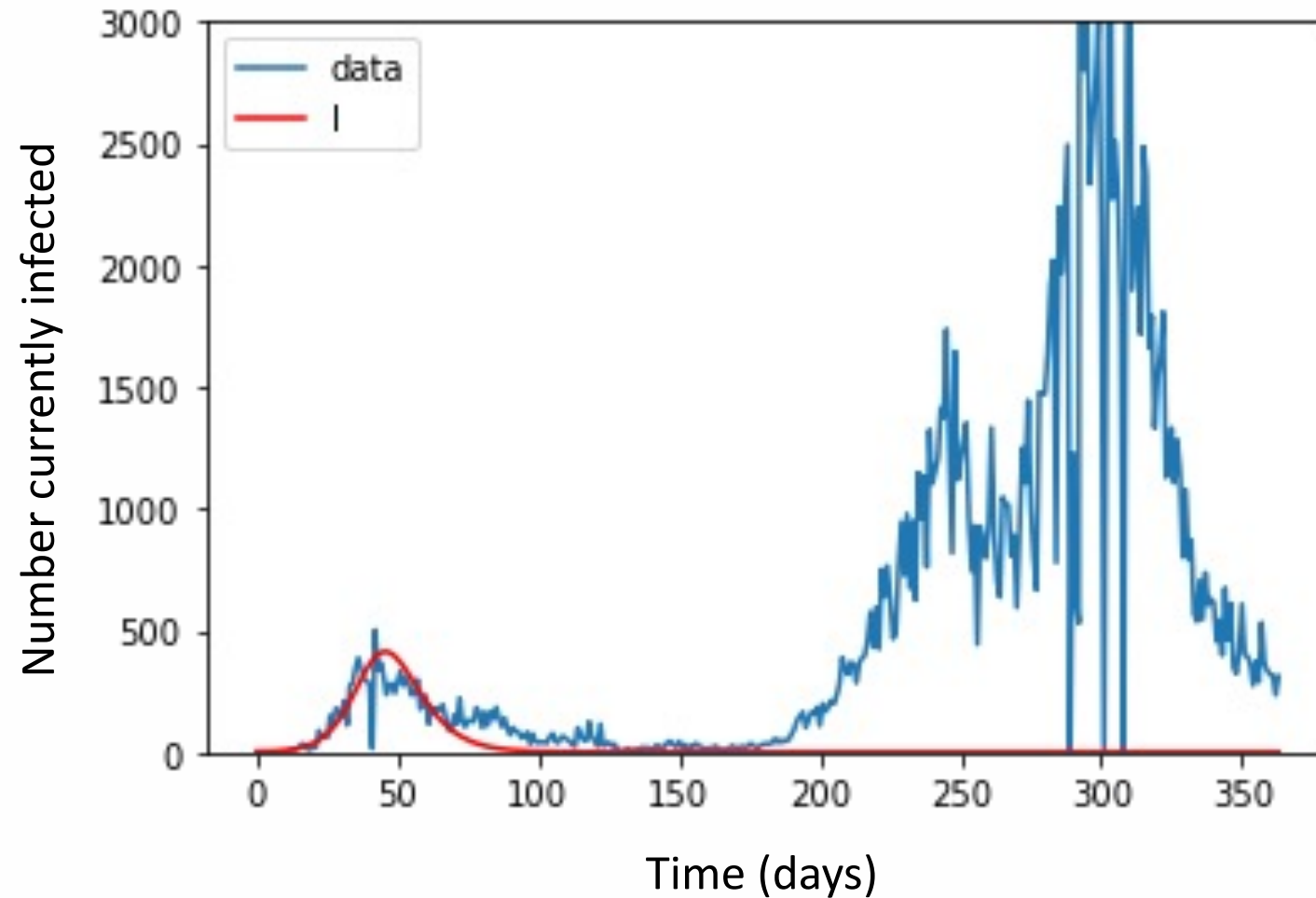
Example: COVID-19



Example: COVID-19



Example: COVID-19



Summary

In this course will be focusing on *mechanistic* models
i.e. using knowledge about how diseases spread to inform model structure

We will consider two types of models

- **Deterministic** models: average rates of movement between compartment
 - Good for large populations and/or high infection levels
 - Low computational effort – only need to run once
- **Stochastic** individual-based models (IBM): sample infection/recovery etc. from distributions
 - Good for small populations and/or low infection levels
 - High computational effort – need to run lots of times and take mean

Week 1: Introduction

- What is a model?
 - Example: COVID-19
 - Model formulation
- **Using R and Rstudio**
 - Installation
 - R markdown
 - Tidyverse
- Course overview

What is R?

R is a flexible and powerful suite of software facilities that is used for

- Data handling, storage and manipulation
- Mathematical and statistical calculations on arrays of numbers
- Data analysis and modelling
- Graphical display

It is open source and constantly evolving

Installing R

Stop the lecture now and go to:

<https://intro2r.com>

Chapter 1: Getting started with R and Rstudio

- Install R and RStudio
- Work through sections 1.3 – 1.12

When you're done, return to the lecture and continue

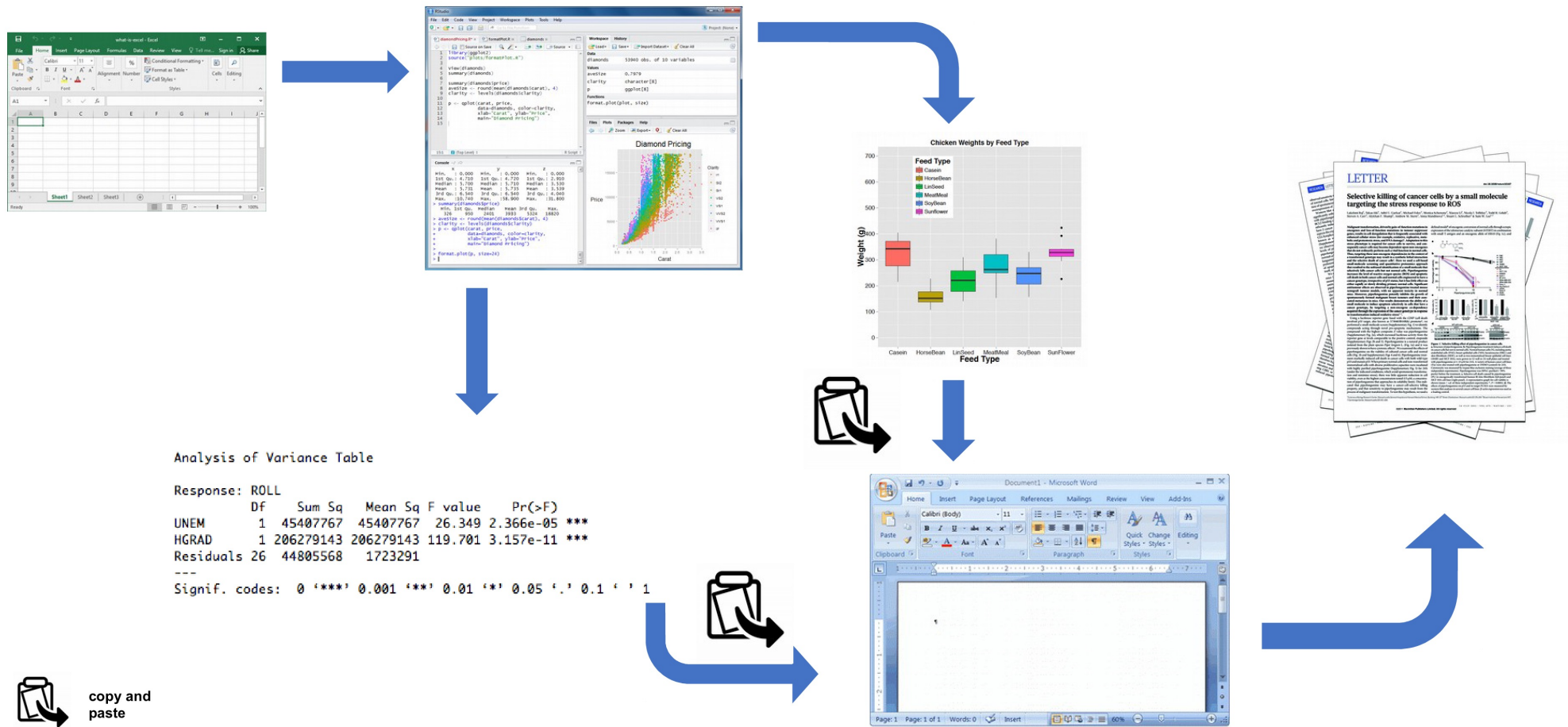
R markdown

A way to combine

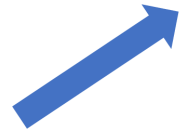
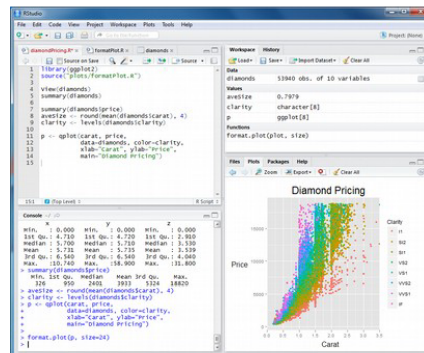
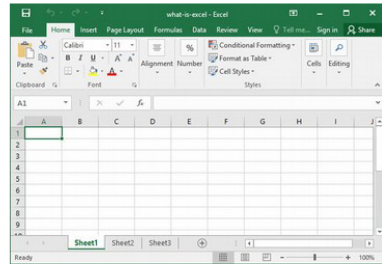
- R code
- Results (including plots and tables)
- Written aspects / commentary

into one neat document that's easy to reproduce.

Workflow without R markdown



Workflow with R markdown



```
1 |---
2 |title: "BESS82 Variance Heterogeneity Practical"
3 |author: "Alex Douglas"
4 |date: "17/10/2019"
5 |output:
6 |  pdf_document: default
7 |  html_document: default
8 |fontfamily: serif
9 |---
10
11 Setup global options for knitr package. Normally I would not display these
12 ""tidy = TRUE"" keeps the displayed code within the code boxes (see what
13
14
15 ""{r setup, include=TRUE}
16 knitr::opts_chunk$set(echo=TRUE, tidy.opts=list(width.cutoff=55), tidy=TRUE)
17 ""
18
19 ## Benthic Biodiversity experiment
20
21 These data were obtained from a mesocosm experiment which aimed to examine
22 nutrient release (NH4-, NO3- and PO4-). At the start of the experiment 1
23 of five polychaete biomass treatments (0, 0.5, 1, 1.5, 2 g per mesocosm). 1
24 either NH4-, NO3- or PO4- was measured in the water. The concentration (c
25 interested in whether the nutrient concentration differed between polychaet
```



	concentration	biomass	fnutrient	fbiomass
1	0.0500	0.0	NH4	0
2	0.1050	0.0	NH4	0
3	0.1050	0.0	NH4	0
4	0.7900	0.5	NH4	0.5
5	0.2100	0.5	NH4	0.5
6	2.1000	0.5	NH4	0.5
7	1.1590	1.0	NH4	1
8	1.0010	1.0	NH4	1
9	0.5227	1.0	NH4	1
10	1.1590	1.5	NH4	1.5
11	0.9200	1.5	NH4	1.5
12	1.5200	1.5	NH4	1.5
13	1.1060	2.0	NH4	2
14	3.2000	2.0	NH4	2
15	3.7400	2.0	NH4	2



R markdown

Step 1: Install R markdown and TinyTeX

Type the following into the terminal in RStudio:

```
# Install from CRAN  
install.packages('rmarkdown', dep = TRUE)
```

```
install.packages('tinytex')  
tinytex::install_tinytex() # install TinyTeX
```


R markdown

Step 2: Create a new R markdown file

Click through to:

File -> New File -> R Markdown...

Provide a document Title and Author and pick HTML as the Output Format

Click OK

R markdown

Markdown documents are made up of three components:

1. Header
2. Formatted text
3. Code chunks

R markdown

1. Header

This is generated for you from the information you provided when you created the file, but can be edited later.

R markdown

2. Formatted text

Formatting:

Goal	R markdown	output
bold text	<code>**mytext**</code>	mytext
italic text	<code>*mytext*</code>	<i>mytext</i>
strikethrough	<code>~~mytext~~</code>	mytext
superscript	<code>mytext^2^</code>	mytext ²
subscript	<code>mytext~2~</code>	mytext ₂

R markdown

2. Formatted text

Headings:

# My heading	My heading
## My heading	My heading
### My heading	My heading
#### My heading	My heading

More #’s = smaller font heading

R markdown

2. Formatted text

Lists:

```
- item 1  
- item 2  
  + sub-item 2  
  + sub-item 3  
- item 3  
- item 4
```

Bulleted

```
1. item 1  
2. item 2  
  + sub-item 2  
  + sub-item 3  
3. item 3  
4. item 4
```

Numbered

R markdown

3. Code chunks

All code chunks start and end with three ‘back quotes’: ```

E.g. Including your code in the document

```
```{r, code-chunk-name}
```

*code*

```
```
```

R markdown

3. Code chunks

All code chunks start and end with three ‘back quotes’: ```

E.g. Including only your code output (graph, table etc.) in the document

```
``{r, code-chunk-name, echo=FALSE}
```

code

```
```
```



## Tidyverse

- Universe of packages for data science

Core packages	Description
ggplot2	For creating graphics
tibble	For creating and manipulating tables
dplyr	For manipulating data
tidyr	For tidying up data
stringr	For working with characters/letters
readr	For importing data into R

## Tidyverse

### Style guide (some rules to follow)

- Avoid using more than 80 characters per line
- Always use a space after a comma (never before)
- Operators (e.g. ==, +, -, <-, %>%) must have a space before and after
- Avoid reusing names of functions and common variables
- Avoid accent marks or special characters in names, files, etc.
- Object names should follow a consistent structure
  - E.g. For denoting day 1 pick one of: Day\_1, day\_1, day\_one, Day\_one and use consistent structure for all similar variables

## Tidyverse

The pipe operator: %>%

This allows multiple functions to be combined iteratively.

Start with the object to be manipulated / functions applied to:

```
data %>% function1() %>% function2() %>% function 3
```

## Tidyverse

Key functions:

function	Description
<code>read_csv()</code>	Reads in data from a .csv file (file must be in your working directory)
<code>mutate()</code>	Add new variables or modify existing ones
<code>select()</code>	Select variables (by name or by characteristic)
<code>filter()</code>	Filter data
<code>summarise()</code>	Summarise/reduce
<code>arrange()</code>	Sort
<code>group_by()</code>	Group (by name or by characteristic)
<code>rename()</code>	Rename columns

# Tidyverse

## Key functions:

function	Example	What it does
<code>read_csv()</code>	<code>read_csv('mydata.csv')</code>	Reads in data from mydata.csv
<code>mutate()</code>	<code>data %&gt;% mutate(newvariable=variable1*variable2)</code>	Creates new variable
<code>select()</code>	<code>data %&gt;% select(variable1, variable2)</code>	Selects just columns for variables 1 & 2
<code>filter()</code>	<code>data %&gt;% filter(variable1==10)</code>	Selects all data for which variable 1 = 10
<code>summarise()</code>	<code>data %&gt;% summarise(mean(variable1))</code>	Calculates the mean for variable1
<code>arrange()</code>	<code>data %&gt;% arrange(variable1)</code>	Sorts data from least to greatest
<code>group_by()</code>	<code>data %&gt;% group_by(variable1)</code>	Groups alike data for further analysis
<code>rename()</code>	<code>data %&gt;% rename(variable1=newname)</code>	Renames variable 1 with newname

### Week 1: Introduction

- What is a model?
  - Example: COVID-19
  - Model formulation
- Using R and Rstudio
  - Installation
  - R markdown
  - Tidyverse
- **Course overview**

## Course overview

Week 1: Introduction to disease modelling and the R language

Week 2: Analysing data in R

Week 3: Plotting and visualization in R

Week 4: Epidemic models (SIR, SEIR etc.)

Week 5: Solving deterministic models in R

Week 6: Stochastic individual-based models

Week 7: Modelling interventions

Week 8: Fitting models to data

Week 9: Further methods

Week 10: Interfacing science and policy

## Weekly reading

<https://intro2r.com>

- Chapter 1: Getting started with R and Rstudio
  - Install R and RStudio
  - Work through sections 1.3 – 1.12
- Chapter 8: Reproducible reports with R markdown