

Metapops

Relaxing the homogeneous mixing assumption

A fundamental assumption of the simple compartmental model presented in [?@sec-Compartmental_models](#) is that any susceptible individual is equally likely to become infected by any of the infectious individuals (at a rate proportional to the size of each compartment in the population). While the infectivity of the disease defined by the β parameter is intended to represent the biological characteristics of the pathogen, it operates by modulating the force of infection determined by the ratio of susceptible to infectious individual. This assumption is known as **Homogeneous mixing**. However, real-world populations are not uniform in their interactions (particularly on a scale like that of [?@sec-Melb_SIR](#)), leading to the need for more realistic models that account for heterogeneous mixing patterns. In this section, we explore models that relax the assumption of homogeneous mixing.

Multi Patch ‘Metapopulation’ Model

The concept of multi-patch or subdivided models is a cornerstone in metapopulation modeling. Pioneering work by researchers such as Wilson [[@wilson1945](#)] and Haskey [[@haskey1957](#)] laid the foundation for these models. Within multi-patch models, each patch represents a distinct subpopulation, and interactions within a patch are assumed to be random, while interactions between patches are limited. **Key Assumptions in Multi-Patch Models**

Metapopulation models, including multi-patch models, often rely on certain assumptions to simplify the mathematical framework while retaining essential epidemiological insights. These assumptions typically include:

Constant Size: The population size within each patch remains constant over time, implying equal birth and death rates.

Constant Recovery Parameter (γ): The rate of recovery from infection (γ) is assumed to be consistent across all patches.

No Permanent Movement of Individuals Between Patches: Individuals do not permanently migrate between patches.

Disease Spread on Contact: Disease transmission occurs through contact between susceptible and infective individuals. The total number of new cases is typically modeled as a proportion, β , of the number of contacts.

Proportionate Mixing: Proportionate mixing, as proposed by Sattenspiel [sattenspiel1988], implies that the number of encounters is proportional to the size of the subpopulations involved. The mixing matrices used in these models are chosen to fit known patterns of incidence, often based on empirical data, rather than theoretical grounds. The importance of variation in mixing patterns for infection transmission is often explored within this framework.

Heterogeneous mixing

Metapopulation models heavily rely on mixing matrices to describe the dynamics of disease transmission within and between patches. Two key types of matrices are commonly used:

Migration Matrix (Bodmer [bodmer1968]): This matrix represents backward stochastic migration probabilities, where the elements m_{ij} give the probabilities that the parents of individuals in population i came from population j .

Mixing (Contact) Matrix (Sattenspiel [sattenspiel1987]): This matrix characterizes the probability of two individuals from different neighborhoods coming into contact. Unlike the migration matrix, the mixing matrix is a forward stochastic migration matrix, with each element m_{ij} representing the probability that an individual from population i moves to population j [sattenspiel1988].

In summary, metapopulation models offer a powerful framework for capturing the complexities of disease transmission in heterogeneous populations. These models rely on assumptions that simplify real-world interactions, while mixing matrices provide a means to quantify the interactions within and between subpopulations. Through these models, we can gain valuable insights into the spread of infectious diseases in diverse and interconnected communities.

Formulation

Let us more formally specify our SIR metapopulation.

As in the simple SIR from @sec_SIR there are three main compartments representing the disease status of individuals of all individuals in the population

$$S + I + R = N$$

However, each compartment is completely partitioned into a number, n , of separate patches i for $i \in \{1 \dots, n\}$.

$$\sum_i^n S_i = S$$

$$\sum_i^n I_i = I$$

$$\sum_i^n R_i = R$$

$$S_i + I_i + R_i = N_i$$

There is only one n , meaning each compartment has equal number of patches.

Mixing

- m_{ij} represents the probability that a susceptible individual from in patch i comes into contact with an infective individual from patch j .
 - $0 \leq m_{ij} \leq 1$ and $\sum_j m_{ij} = 1$
- We can summarise all m_{ij} in a $n \times n$ matrix M , which we call the ‘mixing matrix’
 - M is a stochastic matrix
 -

Force of infection

@moss2019

- Since the probability of infection is a product of the chance of coming in to contact with the constant infectiousness parameter β , we define the Force of Infection (FOI) between two patches, $i, j \in \{1, \dots, n\}$

$$\lambda_{ij} = \beta \cdot I_j \cdot M'_{j,i}$$

- total FOI in patch i (Λ_i)

$$\Lambda_i = \sum_{j=1}^r \lambda_{ij}$$

- and the FOI vector

$$\Lambda = \beta \cdot I \times M'$$

Dynamics

$$\begin{aligned} S'_i &= -\Lambda_i \cdot \frac{S_i}{N_i} \\ I'_i &= \Lambda_i \cdot \frac{S_i}{N_i} - \gamma I_i \end{aligned}$$

Aside: The basic reproduction number R_0 in multipatch models.

- Simple formula derived by @old-guys doesnt necessarily hold (why?)
- Dominant eigenvalue of the Next Generation Matrix (NGM) is the expected number of infections in a population after the introduction of a single infectious individual @diekman
 - In the simple SIR this reduces to $\frac{\beta}{\gamma}$

NGM derivation @Feng et al 2015, @Diekman 2010

the above comprises the ‘infectious subsystem’ The jacobian of the infectious subsystem can be decomposed in to contributions of the - Transmission matrix T containing entries transmission events, - ij th entry of T is the rate at which an individual in infected state j produces individuals with infected state i , -transition, Σ , matrices encodein recoveries “The ij th entry of $2S2I$ can be interpreted as the expected time that an individual who presently has infected state j will spend in infected state i ” The Next generation Matrix $K = -T\Sigma^{-1}$ - the ij th entry of K is the expected number of infected off- spring with state i at infection produced throughout its entire future infected life by an individual presently in infected state j ^ Diekman

now, @Feng2015 The jacobian of the infectious subsystem of the metapopulation model given in @eq_SIR_Metapop_dynamix substituting

$$J = \left(\begin{array}{ccc} \frac{\partial I'_1}{\partial I_1} & \cdots & \frac{\partial I'_1}{\partial I_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial I'_n}{\partial I_1} & \cdots & \frac{\partial I'_n}{\partial I_n} \end{array} \right)_{(I_1=0, I_2=0)} = \left(\begin{array}{ccc} c_{11}\beta - \gamma & \cdots & c_{1n}\beta \\ \vdots & \ddots & \vdots \\ c_{n1}\beta & \cdots & c_{nn}\beta - \gamma \end{array} \right)$$

Is decomposed into

$$J = \begin{pmatrix} c_{11}\beta & \cdots & c_{1n}\beta \\ \vdots & \ddots & \vdots \\ c_{n1}\beta & \cdots & c_{nn}\beta \end{pmatrix} - \begin{pmatrix} \gamma & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \gamma \end{pmatrix}$$

and thus the NGM

$$\begin{aligned} K &= \begin{pmatrix} c_{11}\beta & \cdots & c_{1n}\beta \\ \vdots & \ddots & \vdots \\ c_{n1}\beta & \cdots & c_{nn}\beta \end{pmatrix} \times \begin{pmatrix} \gamma & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \gamma \end{pmatrix}^{-1} \\ &= \begin{pmatrix} c_{11}\frac{\beta}{\gamma} & \cdots & c_{1n}\frac{\beta}{\gamma} \\ \vdots & \ddots & \vdots \\ c_{n1}\frac{\beta}{\gamma} & \cdots & c_{nn}\frac{\beta}{\gamma} \end{pmatrix} \\ &= \frac{\beta}{\gamma} C \end{aligned}$$

- where k is the vector of R_0 per patch i.e. scalar in our case and

Implementation

The metapopulation model can be implemented as a Continuous Time Markov chain in a similar way to `?@sec-SIR_CTMC`, but with a few key differences. Following @moss2019, we will use as an example a metapopulation model of the Greater Melbourne region partitioned in to 40 subpopulations based on the @2023AustralianStatisticalGeography SA3 classification system. The population of each patch along with its numeric SA identifier is given in `?@sec-appendix1` and shown in @figGMelbSA3Pop

Warning: package 'ggplot2' was built under R version 4.2.3

Warning: package 'tibble' was built under R version 4.2.3

Warning: package 'dplyr' was built under R version 4.2.3

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.2      v readr      2.1.4
v forcats    1.0.0      v stringr    1.5.0
v ggplot2    3.4.2      v tibble     3.2.1
v lubridate  1.9.2      v tidyr      1.3.0
v purrr      1.0.1
-- Conflicts ----- tidyverse_conflicts() --
```

```
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

Warning: package 'sf' was built under R version 4.2.3

Linking to GEOS 3.9.3, GDAL 3.5.2, PROJ 8.2.1; sf_use_s2() is TRUE

Rows: 40 Columns: 3-- Column specification -----

Delimiter: ","

chr (1): SA3_Name

dbl (2): SA3_Code, SA3_pop

i Use `spec()` to retrieve the full column specification for this data.

i Specify the column types or set `show_col_types = FALSE` to quiet this message. Rows: 9 Columns: 3

Delimiter: ","

chr (1): SA4_Name

dbl (2): SA4_Code, SA4_pop

i Use `spec()` to retrieve the full column specification for this data.

i Specify the column types or set `show_col_types = FALSE` to quiet this message.

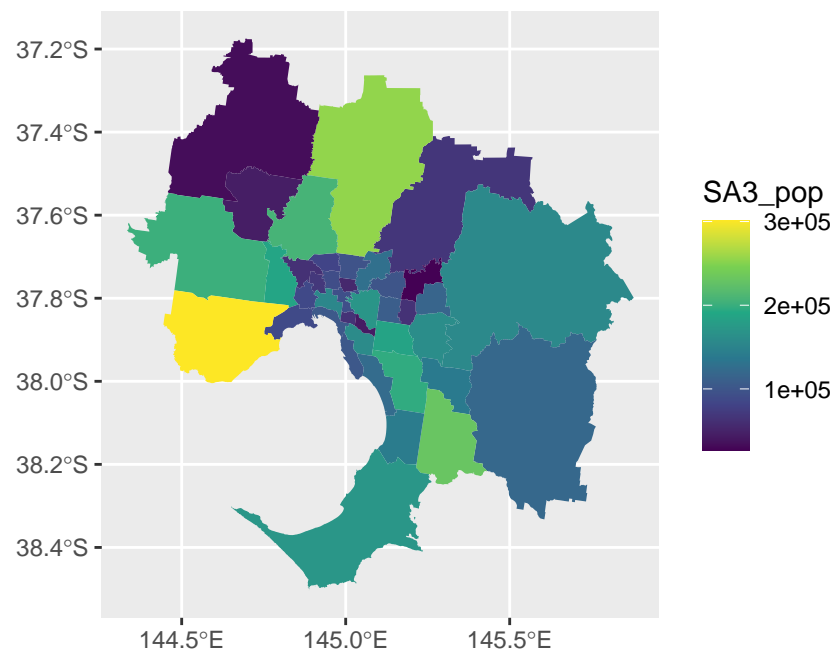


Figure 1: ?(caption)

Initial conditions

At the beginning of a simulation, all patches are composed of entirely susceptible individuals. A number, I_0 , of individuals in a randomly selected patch, α , become infected

$$S_i(0) = \begin{cases} N_i & \text{if } i \neq \alpha \\ N_i - E_0 & \text{if } i = \alpha \end{cases}$$

$$I_i(0) = \begin{cases} 0 & \text{if } i \neq \alpha \\ I_0 & \text{if } i = \alpha \end{cases}$$

for $i \in \{1, \dots, n\}$ where $\alpha \sim \mathcal{U}\{1, r\}$.

Origin-destination mixing matrix

The mixing matrix was developed after @moss2019 using an empirically informed origin-destination (OD) matrix derived from ‘Place of work’ data taken from the Australian Census @ABS_census2016. Rows (‘origin’) are the ‘usual residence’, and the columns (destination) are the ‘place of work’.

The empirically derived OD matrix can be seen in ?@fig-OD_mat_eg

$$F = \begin{pmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n,1} & f_{n,2} & \cdots & f_{n,n} \end{pmatrix}$$

$f_{i,j}$ is the proportion of people who usually reside in patch i listing patch j as their place of work. This empirical method is expected to describe contact patterns *between regions*, but contact patterns within *within* the region of residence are expected to result more from contact outside of a work context.

$$f_{i,i} = 0$$

$$\sum_{j=1}^r f_{i,j} = 1 \quad \forall i \in [1..r]$$

Within patch ‘local’ mixing is given by a parameter δ_i^H , with the remaining proportion, $\delta_r^* = 1 - \delta_i^H$ distributed among the non-local patches giving us the mixing matrix M

$$M = \begin{pmatrix} \delta_1^H & \delta_1^* f_{1,2} & \cdots & \delta_1^* f_{1,r} \\ \delta_2^* f_{2,1} & \delta_2^H & \cdots & \delta_2^* f_{2,r} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_r^* f_{r,1} & \delta_r^* f_{r,2} & \cdots & \delta_r^H \end{pmatrix}$$

Stochastic Simulation

- As per `?@sec-SIR_sim`
- as before, δt will be the sum of exponentially distributed times (`expt(sum.rates)`)
- State change will involve event type (as in single patch) *and* event location
- There are 2 possible events per patch so there are $2n$ possible events
- State change vector is now $2 \times n$ long
- i.e. containing elements for an infection and recovery in each patch
- v is a vector of length $2n$ with elements each of length $2n$ representing the change in compartments S_i and I_i for all patches $i \in \{1, \dots, n\}$ as follows

$$v_i = (\Delta S_1, \dots, \Delta S_n, \Delta I_1, \dots, \Delta I_n)$$

- v_i , where $i \leq n$, corresponds to an infection event in patch i , i.e.

$$v_i = (\{0\}^{i-1}, -1, \{0\}^{n-1}, +1, \{0\}^{n-i})$$

and v_i where $n < i \leq 2n$ corresponds to a recovery in patch $n - i$, i.e.

$$v_i = (\{0\}^{i-1}, -1, \{0\}^{2n-i})$$

- Propensity vector
- for infection events (i.e. all a_i where $i \leq n$) we use the FOI `?@eq-metapop_FOI` vector at patch i

$$a_i = \frac{S_i}{N_i} \cdot \Lambda_i = \frac{S_i}{N_i} \cdot \sum_{j=1}^r \beta \cdot I_j \cdot M'_{j,i}$$

and for recovery events

$$a_{n+j} = \gamma I_j$$

for $j \in \{1, \dots, n\}$, i.e. recovery events The event type and location are thus computed by the same random number

These additions to the standard SIR-simulation algorithm are summarised in Algorithm 1

- population vector \mathbf{N}
- seed at random patch

Algorithm 1 Stochastic simulation of SIR Metapopulation CTMC

Input: \mathbf{N} , I_0 , β , γ , mixing matrix M **Output:** \mathbf{df}

```

1:  $t \leftarrow 0.0$  ▷ Initialise time
2: for all  $i$  do ▷ Initialise susceptibles
3:    $S_i \leftarrow N_i$ 
4: end for
5: Select  $i \sim \mathcal{U}[1, n]$ ,  $I_i \leftarrow I_0$  ▷ Seed infection
6: while  $I \geq 0$  do
7:   for all  $i \in \{1, \dots, n\}$  do ▷ Update Infection Rates
8:      $a_i \leftarrow \frac{S_i}{N_i} \cdot \sum_{j=1}^n \beta \cdot I_j \cdot M'_{j,i}$ 
9:     update  $a_{net} = a_{net} + a_i$ 
10:  end for
11:  for all  $r \in \{n+1, \dots, 2n\}$  do ▷ Update Recovery Rates
12:     $a_r \leftarrow \gamma I_{r-n}$ 
13:    update  $a_{net} = a_{net} + a_r$ 
14:  end for
15:  generate two random numbers  $r_1, r_2 \sim \mathcal{U}(0, 1)$ 
16:  select  $\mu$  such that  $\sum_{j=1}^{\mu} a_j \leq r_1 a_{net}$ 
17:  compute  $\tau \leftarrow \frac{1}{a_{net}} \ln(1/r_2)$ 
18:  update  $X \leftarrow X + v_{\mu}$ 
19:  set  $t \leftarrow t + \tau$ 
20: end while

```

Simulation results

Overall outcome

```

library(tidyverse)

sim_dirs <- list.files("../data/sims/SA3", pattern = "OD_..._none_([0-9]|[1-9][0-9]).cs")

```

```
sims <- read_csv(sim_dirs, col_names = c("t", "S", "I"), skip = 1, id = "sim")
```

Rows: 506862 Columns: 4

-- Column specification -----

Delimiter: ","

dbl (3): t, S, I

i Use ``spec()`` to retrieve the full column specification for this data.

i Specify the column types or set ``show_col_types = FALSE`` to quiet this message.

```
sims <- sims |>
  filter() %>%
  mutate(
    S = as.numeric(S),
    I = as.numeric(I),
    Cum_I = S - 4976157,
    prop_I = I/4976157,
    t = as.numeric(t),
    SA = str_extract(sim, "SA./SA(.)", 1),
    Mixmat = str_extract(sim, "SA./SA._..._.(HPMM|HMM|OD)_", 1),
    mu = as.numeric(str_extract(sim, "SA./SA.*_OD_(...)", 1)),
    Intervention = factor(str_extract(sim, "/SA.*(none|local|travel|total)", 1), levels = c(
    beta = as.numeric(str_extract(sim, "SA./SA._(...)", group = 1)),
    gamma = as.numeric(str_extract(sim, "SA./SA._..._(...)", group = 1)),
    R0 = beta/gamma,
    sim_num = str_extract(sim, "_(.{1,3}).csv", 1))
```

```
sim_stats <- sims %>%
  group_by(t, mu, R0) %>%
  summarise(
    avg_I = median(I),
    sd_I = sd(I)
  )
```

``summarise()`` has grouped output by 't', 'mu'. You can override using the ``groups`` argument.

Infection curves

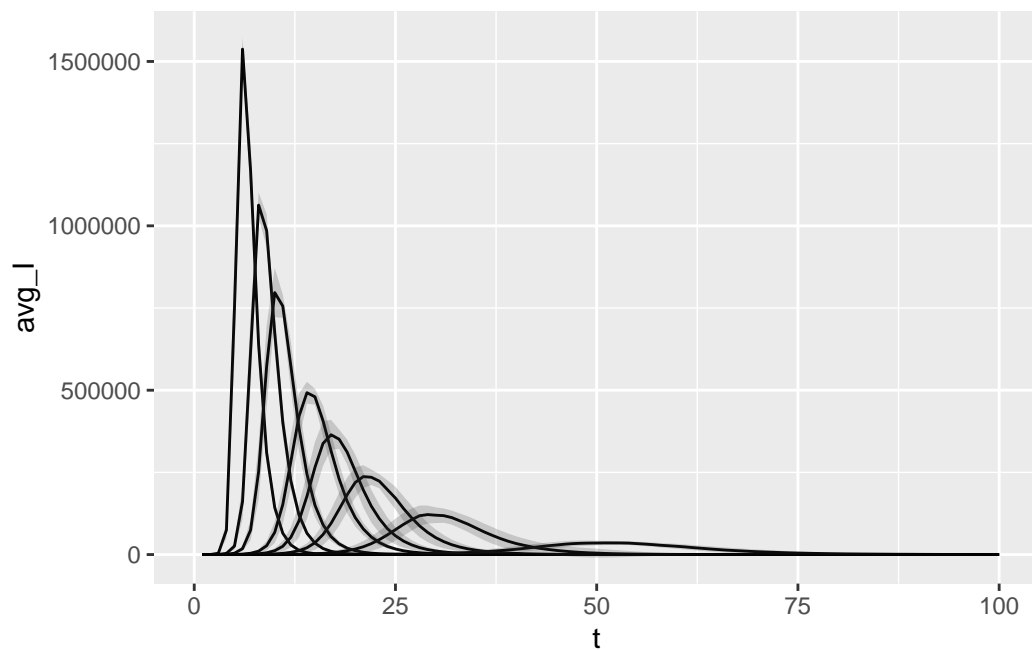
By R_0

```

sim_stats %>%
  filter(mu == "0.5" ) %>%
  ggplot() +
    aes(x = t, y = avg_I, group = R0) +
    geom_line()+
    geom_ribbon(aes(ymin = avg_I-sd_I, ymax= avg_I+sd_I), alpha = 0.2) +
    xlim(c(0,100))

```

Warning: Removed 404 rows containing missing values (`geom_line()`).

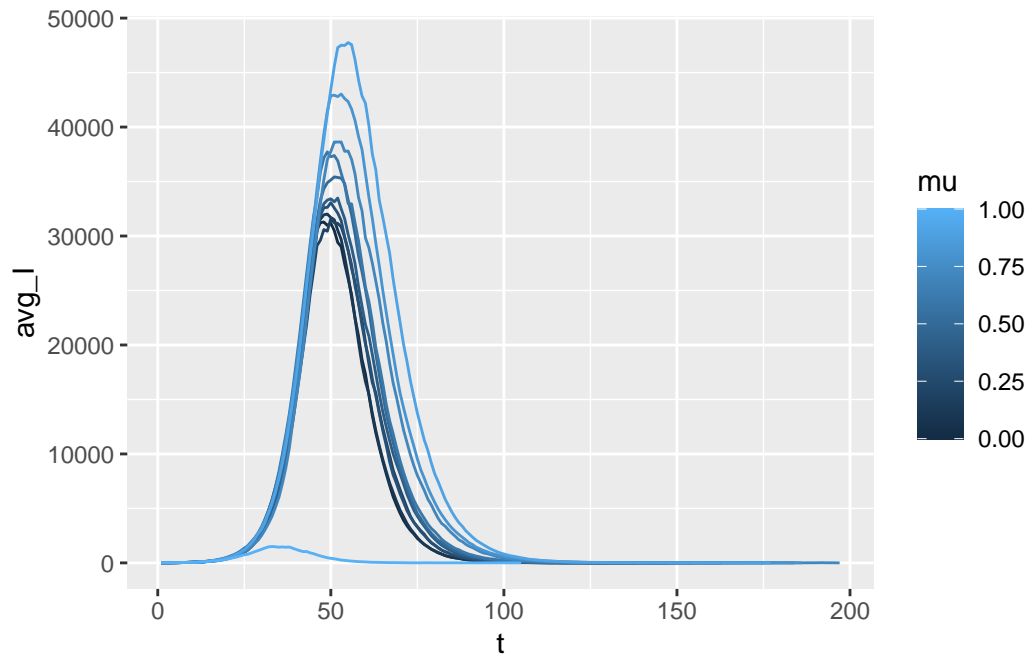


by δ_H

```

sim_stats %>%
  filter(R0 == 1.2) %>%
  ggplot() +
    aes(x = t, y = avg_I, group = mu, col = mu) +
    geom_line()

```



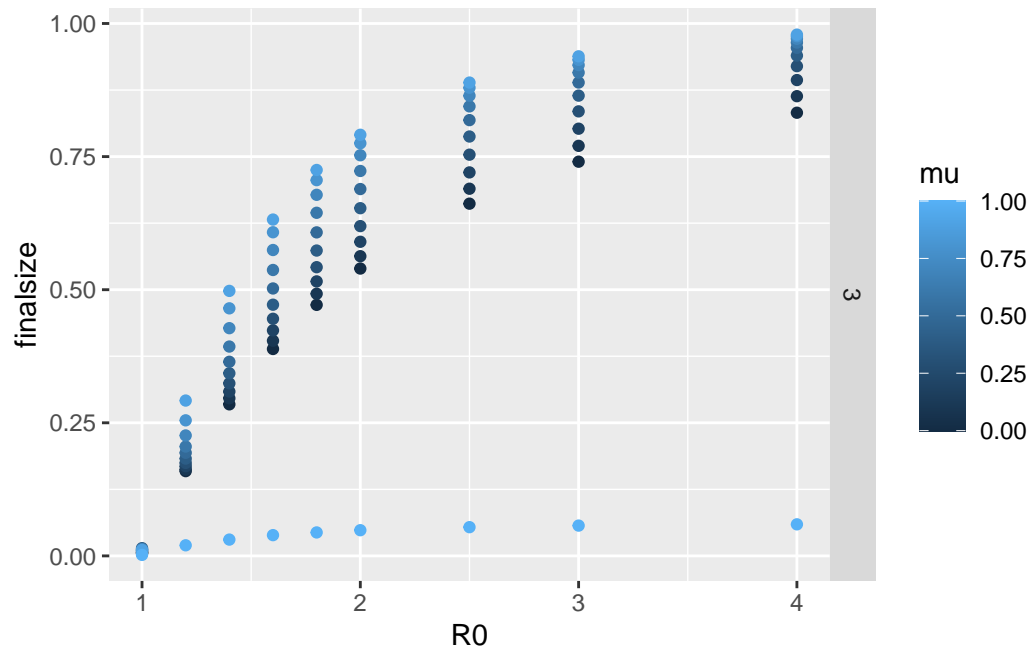
Summary statistics

```
sim_summary <- sims %>%
  group_by(SA, Mixmat, mu, Intervention, R0) %>%
  summarise(peaksize = max(I)/4976157,
            finalsize = (4976157 - min(S))/4976157,
            duration = max(t),
            peakttime = t[which(I == max(I))])
```

`summarise()` has grouped output by 'SA', 'Mixmat', 'mu', 'Intervention'. You can override using the `.groups` argument.

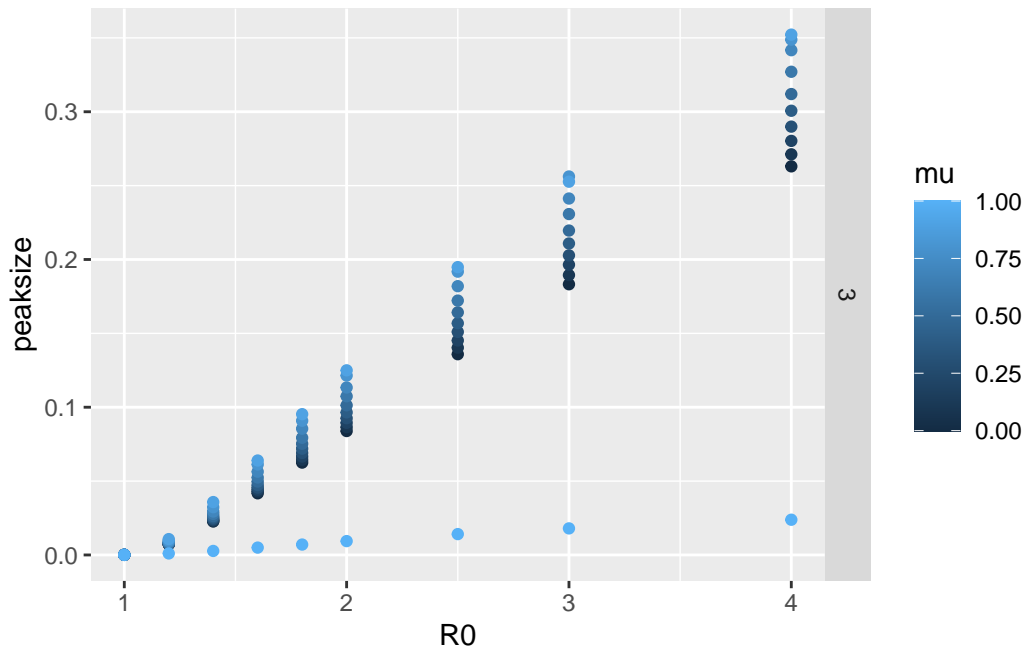
Final size

```
sim_summary %>%
  filter(Intervention == "none", SA == "3") %>%
  ggplot() +
  aes(x = R0, y = finalsize) +
  geom_point(aes(col = mu)) +
  facet_grid(rows = vars(SA))
```



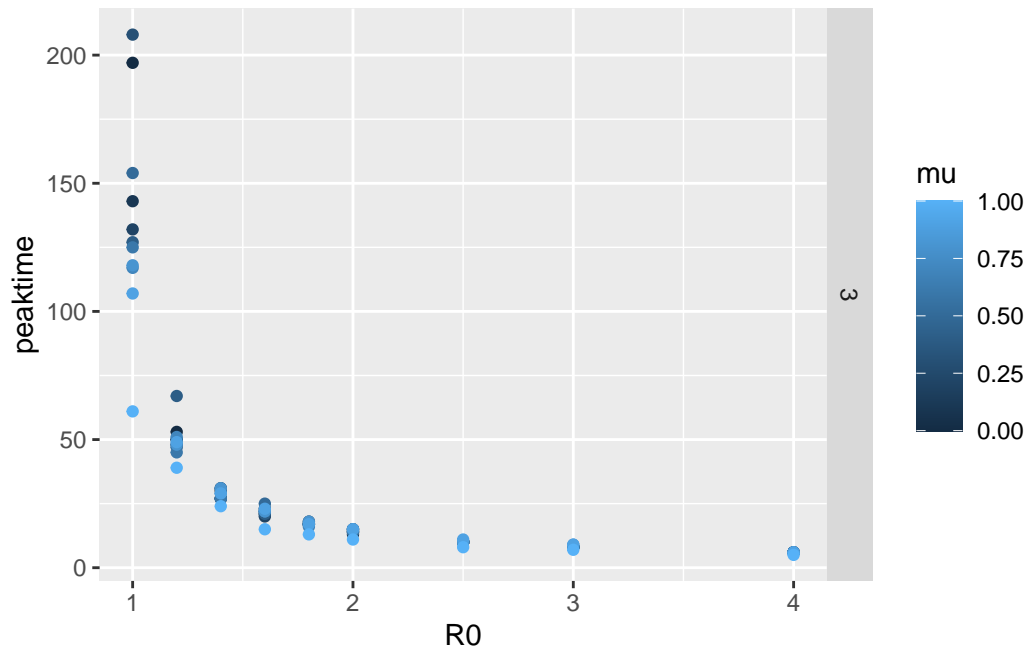
Peak size

```
sim_summary %>%
  filter(Intervention == "none", SA == "3") %>%
  ggplot() +
  aes(x = R0, y = peaksize) +
  geom_point(aes(col = mu)) +
  facet_grid(rows = vars(SA))
```



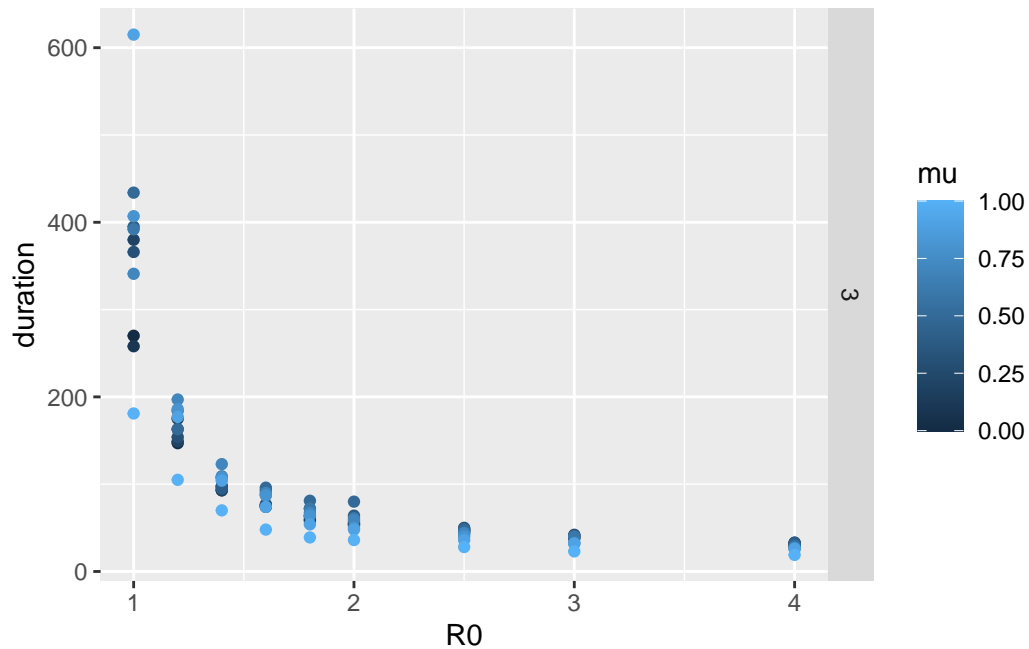
Peak time

```
sim_summary %>%
  filter(Intervention == "none", SA == "3") %>%
  ggplot() +
  aes(x = R0, y = peaktime) +
  geom_point(aes(col = mu)) +
  facet_grid(rows = vars(SA))
```



Duration

```
sim_summary %>%
  filter(Intervention == "none", SA == "3") %>%
  ggplot() +
  aes(x = R0, y = duration) +
  geom_point(aes(col = mu)) +
  facet_grid(rows = vars(SA))
```



patch example

```
SA3_OD_Patch_dirs <- list.files("../..data/sims/SA3", pattern = "OD..._none_(patchinf|pa
SA3_OD_Patch_logs <- read_csv(SA3_OD_Patch_dirs, id = "sim")
```

Rows: 1013724 Columns: 41

-- Column specification -----

Delimiter: ","

dbl (40): Brunswick - Coburg, Darebin - South, Essendon, Melbourne City, Por...

i Use `spec()` to retrieve the full column specification for this data.

i Specify the column types or set `show_col_types = FALSE` to quiet this message.

```
GMelb_Popns <- read_csv("../..data/GmelbSA3Pop21.csv")
```

Rows: 40 Columns: 3

-- Column specification -----

Delimiter: ","


```
chr (1): SA3_Name
dbl (2): SA3_Code, SA3_pop
```

- i Use ``spec()`` to retrieve the full column specification for this data.
- i Specify the column types or set ``show_col_types = FALSE`` to quiet this message.

```
SA3_patch_sims <- SA3_OD_Patch_logs %>%
  group_by(sim) %>%
  mutate(t = row_number()) %>%
  ungroup() %>%
  pivot_longer(cols = c(-sim, -t), values_to = "Count", names_to = "patch") %>%
  mutate(
    SA = str_extract(sim, "SA./SA(.)", 1),
    Mixmat = str_extract(sim, "SA./SA._..._._(HPMM|HMM|OD)_", 1),
    mu = as.numeric(str_extract(sim, "SA./SA.*_OD_(...)", 1)),
    Intervention = "none",
    beta = as.numeric(str_extract(sim, "SA./SA._(...)", group = 1)),
    gamma = as.numeric(str_extract(sim, "SA./SA._..._(...)", group = 1)),
    R0 = beta/gamma,
    sim_num = str_extract(sim, "_(.{1,3}).csv", 1),
    Measure = str_extract(sim, "(inf|sus)_.{1,3}.csv", 1),
  ) %>%
  select(-sim) %>%
  left_join(GMelb_Popns, by = join_by("patch" == "SA3_Name")) %>%
  pivot_wider( names_from = Measure, values_from = Count) %>%
  mutate(cum_inf = SA3_pop - sus,
         prop_inf = inf/SA3_pop)

SA3_patch_sims %>%
  filter(R0 == 1.4, mu == "0.5", sim_num == "5") %>%
  ggplot()+
  aes(x = t, y = prop_inf) +
  geom_line(aes(col = patch))
```