# Model parameterisation

Alexander E. Zarebski

November 20, 2024

## Contents

This document contains derivation of the core model parameters used. Note that the definition of the differential equations can be found in the `sir_vacc` function in `ethics.model` (this is the function used by `create-grid-database.py`).

```
library(glue)
library(dplyr)
library(ggplot2)
gprint <- \(s) print(glue(s))
gpasteprint <- \(ss) print(glue(paste(ss, collapse = "\n")))
```

# 1 Parameters from the literature

## 1.1 Melbourne population size total

The population size is based off of Melbourne rounded to the nearest million.

```
pop_size <- 5000000
```

```
5000000.0
```

## 1.2 Contact matrix

The *beta* contact matrix is

```
beta_matrix <- matrix(c(0.38, 0.14, 0.14, 0.34), ncol = 2)
```

```
0.38  0.14
0.14  0.34
```

## 1.3 Reproduction number

We have an estimate of the effective and basic reproduction number. The basic reproduction number estimate of 9.5 is based on relative values between strains. The effective reproduction number estimate is 3.4.

```
repro_num <- 3.4
```

## 1.4 Average hospitalisation given vaccination (clinical burden of vaccination)

### 1.4.1 Proportion of vaccinations that result in hospitalisation

This is the proportion of vaccinations that result in hospitalisation as described in Table 2.

```
vacc_hosp_rate_female_00_29 <- 2 / 10^5
vacc_hosp_rate_male_00_29 <- 20 / 10^5
vacc_hosp_rate_30_pl <- 2 / 10^5
```

```
2e-05
```

### 1.4.2 Days in hospital given hospitalisation

```
average_vacc_hosp_stay <- 5.7
```

```
5.7
```

## 1.5 Average hospitalisation given infection (clinical burden of infection)

The average duration of a hospital stay as part of the clinical burden (i.e. the average number of days in hospital given an individual is hospitalised due to infection) comes from the average of a gamma distribution. These values are given in Table 3.

```
average_hospital_stay_00_39 <- 2.27 * 0.95
average_hospital_stay_40_69 <- 4.98 * 0.79
average_hospital_stay_70_xx <- 7.32 * 1.04

data.frame(
  age = c("00-39", "40-69", "70 plus"),
  val = c(average_hospital_stay_00_39,
          average_hospital_stay_40_69,
          average_hospital_stay_70_xx)
)
```

| | |
|---:|---:|
| 00-39 | 2.1565 |
| 40-69 | 3.9342 |
| 70 plus | 7.6128 |

## 1.6 Average infectious period

Table 1 has the average *recovery period* as 10.37 days. This gets used in the estimate of the rate of removal from the infectious compartment.

```
average_infectious_period <- 10.37
```

```
10.37
```

## 1.7 Proportion hospitalised given infection

The *infection hospitalistion ratio (rate)* (IHR) is the proportion of infections that result in hospitalisation. The following data comes from Table 3 of Skiwronski et al (2023). These values will need to be combined with the 1.8 to get the group specific proportions of infections that result in hospitalisation.

```
ihr_list <- list(
  age_00_04 = 0.0041,
  age_05_09 = 0.0010,
  age_10_19 = 0.00013,
  age_20_29 = 0.00022,
  age_30_39 = 0.0004,
  age_40_49 = 0.0003,
  age_50_59 = 0.0006,
  age_60_69 = 0.0025,
  age_70_79 = 0.0042,
  age_80_pl = 0.0217
)
```

```
0.0041   0.001   0.00013   0.00022   0.0004   0.0003   0.0006   0.0025   0.0042   0.0217
```

## 1.8 Demographic details from data ABS

There is one point where we have sex specific data that needs processing so we also need to keep a copy of `age_sex_df` for that, but otherwise, we only care about age, so for the most part this is just about `age_df`.

```
age_sex_csv <-
  "./extrascripts/demographics/Australia_National_Age_and_Sex_2023.csv"
```

```
age_sex_df <-
```

```
  age_sex_csv |>
  read.csv() |>
  rename(age = Age, num = N)

age_df <-
  age_sex_df |>
  dplyr::select(age, num) |>
  group_by(age) |>
  summarise(total_num = sum(num)) |>
  mutate(proportion = total_num / sum(total_num))

age_df |> head()
```

```
0   295859   0.0110979247943317
1   310421   0.0116441579015046
2   302429   0.0113443711282231
3   300411   0.0112686742177523
4   307780   0.0115450917268003
5   310831   0.0116595373530869
```

It will be useful to pull out a couple of summary statistics from `age_sex_df` here, and then we will drop `age_sex_df` as we no longer need it.

```
aus_num_female_00_29 <-
  age_sex_df |>
  filter(Female, age < 30) |>
  select(num) |>
  sum()
aus_num_male_00_29 <-
  age_sex_df |>
  filter(Male, age < 30) |>
  select(num) |>
  sum()
aus_num_people_30_69 <-
  age_df |>
  filter(30 <= age, age < 70) |>
  select(total_num) |>
  sum()
aus_num_people_00_69 <-
  aus_num_female_00_29 + aus_num_male_00_29 + aus_num_people_30_69
rm(age_sex_df)
```

# 2 Derived parameters YES

## 2.1 Group sizes

```
group_1_prop <- sum(age_df[age_df$age < 70, "proportion"])
group_2_prop <- 1 - group_1_prop
list(group_1_prop, group_2_prop)
```

```
0.878761119906157    0.121238880093843
```

```
group_1_size <- round(group_1_prop * pop_size)
group_2_size <- round(group_2_prop * pop_size)

gprint("Group 1 size {group_1_size}\nGroup 2 size {group_2_size}")
```

```
Group 1 size 4393806
Group 2 size 606194
```

## 2.2 Age group proportions

For some calculations, e.g. IHR, we need the age group proportions, so we compute these here.

```
age_prop_list <- list(
  age_00_04 = sum(age_df[age_df$age < 5, "proportion"]),
  age_05_09 = sum(age_df[5 <= age_df$age & age_df$age <= 9, "proportion"]),
  age_10_19 = sum(age_df[10 <= age_df$age & age_df$age <= 19, "proportion"]),
  age_20_29 = sum(age_df[20 <= age_df$age & age_df$age <= 29, "proportion"]),
  age_30_39 = sum(age_df[30 <= age_df$age & age_df$age <= 39, "proportion"]),
  age_40_49 = sum(age_df[40 <= age_df$age & age_df$age <= 49, "proportion"]),
  age_50_59 = sum(age_df[50 <= age_df$age & age_df$age <= 59, "proportion"]),
  age_60_69 = sum(age_df[60 <= age_df$age & age_df$age <= 69, "proportion"]),
  age_70_79 = sum(age_df[70 <= age_df$age & age_df$age <= 79, "proportion"]),
  age_80_pl = sum(age_df[80 <= age_df$age, "proportion"])
)

data.frame(age = names(age_prop_list),
           val = as.numeric(age_prop_list))
```

| | |
|---|---|
| $\text{age}_{0004}$ | 0.056900219768612 |
| $\text{age}_{0509}$ | 0.0604134491728631 |
| $\text{age}_{1019}$ | 0.122520138454076 |
| $\text{age}_{2029}$ | 0.137242924964631 |
| $\text{age}_{3039}$ | 0.147277266904906 |
| $\text{age}_{4049}$ | 0.127418643826456 |
| $\text{age}_{5059}$ | 0.120254370127433 |
| $\text{age}_{6069}$ | 0.106734106687181 |
| $\text{age}_{7079}$ | 0.0783156934774771 |
| $\text{age}_{80\text{pl}}$ | 0.0429231866163661 |

### 2.3 Gamma removal rate

```
gamma_est <- 1 / average_infectious_period
gprint("Estimated removal rate {gamma_est}")

Estimated removal rate 0.0964320154291225
```

### 2.4 Hospitalisation

People who are vaccinated but unprotected have the same dynamics as those who are unvaccinated. For the vaccinated and protected people, there is no infection possible. The *average hospital stay* (measured in days) is what we are using as our cost associated with infection.

We have the proportion of infections that result in hospitalisation, via the IHR, and the average hopsital stay given hospitalisation; we can then combine these to get the average number of days in hospital given infection. Note that we conditioning on infection in an age group here so we need to divide through by the total proportion in that age group to get the age conditioning correct.

#### 2.4.1 Proportion of infections that end up hospitalised

The following values are the probability of going to hospital, given you are infected. These values are needed for the simulation configuration file.

```
prob_hosp_given_inf_group_1 <-
  (ihr_list$age_00_04 * age_prop_list$age_00_04 +
  ihr_list$age_05_09 * age_prop_list$age_05_09 +
  ihr_list$age_10_19 * age_prop_list$age_10_19 +
  ihr_list$age_20_29 * age_prop_list$age_20_29 +
  ihr_list$age_30_39 * age_prop_list$age_30_39 +
  ihr_list$age_40_49 * age_prop_list$age_40_49 +
  ihr_list$age_50_59 * age_prop_list$age_50_59 +
  ihr_list$age_60_69 * age_prop_list$age_60_69) / group_1_prop

prob_hosp_given_inf_group_2 <-
  (ihr_list$age_70_79 * age_prop_list$age_70_79 +
  ihr_list$age_80_pl * age_prop_list$age_80_pl) / group_2_prop

data.frame(
  group = c("Group 1", "Group 2"),
  val = c(prob_hosp_given_inf_group_1, prob_hosp_given_inf_group_2)
)
```

```
Group 1   0.000883004246367426
Group 2     0.0103956673074263
```

### 2.4.2 Average days in hospital for members of each group given hospitalised

```
avg_days_given_hosp_group_1 <-
  (average_hospital_stay_00_39 * age_prop_list$age_00_04 +
  average_hospital_stay_00_39 * age_prop_list$age_05_09 +
  average_hospital_stay_00_39 * age_prop_list$age_10_19 +
  average_hospital_stay_00_39 * age_prop_list$age_20_29 +
  average_hospital_stay_00_39 * age_prop_list$age_30_39 +
  average_hospital_stay_40_69 * age_prop_list$age_40_49 +
  average_hospital_stay_40_69 * age_prop_list$age_50_59 +
  average_hospital_stay_40_69 * age_prop_list$age_60_69) / group_1_prop

avg_days_given_hosp_group_2 <-
  (average_hospital_stay_70_xx * age_prop_list$age_70_79 +
  average_hospital_stay_70_xx * age_prop_list$age_80_pl) / group_2_prop

data.frame(
  group = c("Group 1", "Group 2"),
  val = c(avg_days_given_hosp_group_1, avg_days_given_hosp_group_2)
)
```

```
Group 1   2.87345199536241
Group 2             7.6128
```

### 2.4.3 Putting it all together

```
avg_hosp_given_inf_group_1 <-
  (average_hospital_stay_00_39 * ihr_list$age_00_04 * age_prop_list$age_00_04 +
  average_hospital_stay_00_39 * ihr_list$age_05_09 * age_prop_list$age_05_09 +
  average_hospital_stay_00_39 * ihr_list$age_10_19 * age_prop_list$age_10_19 +
  average_hospital_stay_00_39 * ihr_list$age_20_29 * age_prop_list$age_20_29 +
  average_hospital_stay_00_39 * ihr_list$age_30_39 * age_prop_list$age_30_39 +
  average_hospital_stay_40_69 * ihr_list$age_40_49 * age_prop_list$age_40_49 +
  average_hospital_stay_40_69 * ihr_list$age_50_59 * age_prop_list$age_50_59 +
  average_hospital_stay_40_69 * ihr_list$age_60_69 * age_prop_list$age_60_69) / group_

avg_hosp_given_inf_group_2 <-
  (average_hospital_stay_70_xx * ihr_list$age_70_79 * age_prop_list$age_70_79 +
  average_hospital_stay_70_xx * ihr_list$age_80_pl * age_prop_list$age_80_pl) / group_

data.frame(
  group = c("Group 1", "Group 2"),
  val = c(avg_hosp_given_inf_group_1, avg_hosp_given_inf_group_2)
)
```

```
Group 1   0.00266728704577231
Group 2    0.0791401360779753
```

## 2.5 Vaccination

### 2.5.1 Proportion of vaccinations that end up hospitalised

The `prob_hosp_given_vacc_group_<X>` is the probability of being hospitalised given you are vaccinated (and a member of group `X`).

```
prob_hosp_given_vacc_group_1 <-
  vacc_hosp_rate_male_00_29 * (aus_num_male_00_29 / aus_num_people_00_69) +
  vacc_hosp_rate_female_00_29 * (aus_num_female_00_29 / aus_num_people_00_69) +
  vacc_hosp_rate_30_pl * (aus_num_people_30_69 / aus_num_people_00_69)

prob_hosp_given_vacc_group_2 <- vacc_hosp_rate_30_pl

gprint("Group 1 prob {prob_hosp_given_vacc_group_1}\nGroup 2 prob {prob_hosp_given_vacc
```

```
Group 1 prob 5.96677888407262e-05
Group 2 prob 2e-05
```

### 2.5.2 Average days in hospital for members of each group given vaccine related hospitalisation

This is just re-using the value from the literature, but we add the definition here so that there is consistent variable naming across the two reasons for hospitalisation here.

```
avg_days_given_vacc_group_1 <- average_vacc_hosp_stay
avg_days_given_vacc_group_2 <- average_vacc_hosp_stay

gprint("Group 1 days {avg_days_given_vacc_group_1}\nGroup 2 days {avg_days_given_vacc_g
```

Group 1 days 5.7
Group 2 days 5.7

## 2.6 Basic reproduction number

We given the *beta* matrix and a value for *gamma*, we can compute the value of the basic reproduction number. The expression is

$$\frac{\beta_{11} + \beta_{22} + \sqrt{\beta_{11}^2 - 2\beta_{22}\beta_{11} + \beta_{22}^2 + 4\beta_{12}\beta_{21}}}{2\gamma}.$$

Since this is linear in a multiple of the beta matrix, if we have $\gamma$ and a desired $R_0$ we can compute the missing multipler. For example, with the current *beta* matrix and *gamma*, the value of $R_0$ is given in the following snippet.

```
r0_from_beta_and_gamma <- function(b, g) {
  (b[1,1] + b[2,2] + sqrt(b[1,1]^2 - 2 * b[2,2] * b[1,1] + b[2,2]^2 + 4 * b[1,2] * b[2
}

print(r0_from_beta_and_gamma(beta_matrix, gamma_est))
```

```
5.1997394641809
```

If you wanted $R_0 = 3.4$ you would need the following *beta* matrix.

```
bb <- beta_matrix
bb_multiplier <- 2 * gamma_est * 3.4 / (bb[1,1] + bb[2,2] + sqrt(bb[1,1]^2 - 2 * bb[2,2
rm(bb)
if (3.4 != r0_from_beta_and_gamma(bb_multiplier * beta_matrix, gamma_est)) {
  stop()
}
print(bb_multiplier * beta_matrix)
```

| | |
|---|---|
| 0.248473987764217 | 0.0915430481236588 |
| 0.0915430481236588 | 0.222318831157457 |

Or if you wanted $R_0 = 9.5$ you would need the following *beta* matrix

```
bb <- beta_matrix
bb_multiplier <- 2 * gamma_est * 9.5 / (bb[1,1] + bb[2,2] + sqrt(bb[1,1]^2 - 2 * bb[2,
rm(bb)
if (9.5 != r0_from_beta_and_gamma(bb_multiplier * beta_matrix, gamma_est)) {
  stop("bad calculation")
}
print(bb_multiplier * beta_matrix)
```

| | |
|---|---|
| 0.694265554047076 | 0.25578204622787 |
| 0.25578204622787 | 0.621184969410542 |