

Homework 1

Spencer Matthews

10/11/2021

Problem 1

Complete the following programming exercises in any language of your choice.

(a) Discrete time individual-based modeling

(i) Write a function that simulates a fixed number of generations of a discrete time individual-based SIR model. Your function should have the following user specified arguments: initial vector of individual labels, infection probability p , and number of generations r

```
#' Individual Discrete-Time SIR Model
#'
#' @param init_labels The initial labels of the population, a character vector of S,I,R
#' @param p The infection probability, a number between 0 and 1
#' @param r The number of generations (ie, the number of time steps), an integer
individual_discrete <- function(init_labels, p, r) {
  step_labs <- init_labels
  res <- data.frame(step_0 = init_labels)
  for (i in 1:r) {
    i_t <- sum(step_labs == "I")
    step_prob <- 1 - (1-p)^i_t
    step_labs <- purrr::map_chr(
      .x = step_labs,
      .f = ~{
        if (.x == "S") {
          sample(c("I", "S"), size = 1, prob = c(step_prob, 1 - step_prob))
        } else{
          "R"
        }
      }
    )
    res <- cbind(res, data.frame(step_labs))
  }
  res %>%
    magrittr::set_colnames(
      stringr::str_c("step_", 0:r)
    ) %>%
    magrittr::set_rownames(
      stringr::str_c("person_", 1:length(init_labels))
    )
}
```

(ii) Assuming that initially the population of interest has 9 susceptible individuals, 1 infectious individual, and 0 removed individuals, print 5 realizations of label vectors at each generation using $p = 0.3$ and $r = 5$.

```
init <- c("S", "S", "S", "S", "S", "S", "I", "S", "S", "S")

for (i in 1:5) {
  print(stringr::str_c("Realization ", i))
  individual_discrete(init, p = 0.3, r = 5) %>%
    print()
}
```

```
## [1] "Realization 1"
##      step_0 step_1 step_2 step_3 step_4 step_5
## person_1      S      I      R      R      R      R
## person_2      S      S      S      I      R      R
## person_3      S      S      I      R      R      R
## person_4      S      S      S      I      R      R
## person_5      S      S      I      R      R      R
## person_6      S      I      R      R      R      R
## person_7      I      R      R      R      R      R
## person_8      S      S      S      I      R      R
## person_9      S      I      R      R      R      R
## person_10     S      S      I      R      R      R
## [1] "Realization 2"
##      step_0 step_1 step_2 step_3 step_4 step_5
## person_1      S      I      R      R      R      R
## person_2      S      I      R      R      R      R
## person_3      S      S      S      I      R      R
## person_4      S      S      I      R      R      R
## person_5      S      S      I      R      R      R
## person_6      S      S      I      R      R      R
## person_7      I      R      R      R      R      R
## person_8      S      I      R      R      R      R
## person_9      S      S      I      R      R      R
## person_10     S      S      I      R      R      R
## [1] "Realization 3"
##      step_0 step_1 step_2 step_3 step_4 step_5
## person_1      S      S      S      S      I      R
## person_2      S      S      S      S      I      R
## person_3      S      S      S      I      R      R
## person_4      S      S      I      R      R      R
## person_5      S      S      S      I      R      R
## person_6      S      I      R      R      R      R
## person_7      I      R      R      R      R      R
## person_8      S      S      I      R      R      R
## person_9      S      S      S      S      S      S
## person_10     S      I      R      R      R      R
## [1] "Realization 4"
##      step_0 step_1 step_2 step_3 step_4 step_5
## person_1      S      S      I      R      R      R
## person_2      S      I      R      R      R      R
## person_3      S      S      I      R      R      R
## person_4      S      I      R      R      R      R
## person_5      S      S      I      R      R      R
```

```
## person_6      S      S      S      I      R      R
## person_7      I      R      R      R      R      R
## person_8      S      I      R      R      R      R
## person_9      S      S      S      I      R      R
## person_10     S      S      S      I      R      R
## [1] "Realization 5"
##           step_0 step_1 step_2 step_3 step_4 step_5
## person_1      S      S      S      S      S      S
## person_2      S      I      R      R      R      R
## person_3      S      S      I      R      R      R
## person_4      S      I      R      R      R      R
## person_5      S      S      I      R      R      R
## person_6      S      I      R      R      R      R
## person_7      I      R      R      R      R      R
## person_8      S      S      I      R      R      R
## person_9      S      I      R      R      R      R
## person_10     S      S      S      S      S      S
```

(b) Continuous time individual-based modeling

(i) Write a function that simulates a fixed number of time units of a continuous time individual-based SIR model. Your function should have the following user specified arguments: initial vector of individual labels, infection rate β , removal rate γ , and the number of time units t .

```
individual_continuous <- function(init_labels, beta, gamma, t) {
  step_labs <- init_labels
  res <- matrix(init_labels, nrow = 1) %>%
    as.data.frame() %>%
    magrittr::set_colnames(stringr::str_c("person_", 1:length(init_labels)))
  times <- c(0)
  repeat {
    i_t <- sum(step_labs == "I")
    tmp_df <- data.frame(state = step_labs) %>%
      dplyr::mutate(
        exp_race = purrr::map_dbl(
          .x = state,
          .f = ~{
            if (.x == "S") rexp(1, rate = beta * i_t)
            else if (.x == "I") rexp(1, rate = gamma)
            else if (.x == "R") NA_real_
          }
        ),
        state = dplyr::case_when(
          exp_race == min(exp_race, na.rm = TRUE) & state == "S" ~ "I",
          exp_race == min(exp_race, na.rm = TRUE) & state == "I" ~ "R",
          TRUE ~ state
        )
      )
    step_labs <- tmp_df$state
    res <- rbind(res, step_labs)
    times <- append(times, min(tmp_df$exp_race, na.rm = TRUE))

    if (!(sum(times, na.rm = TRUE) <= t & (!is.na(sum(times))))) break
  }
}
```

```

res %>%
  magrittr::set_rownames(
    stringr::str_c("step_", 0:(nrow(.)-1))
  ) %>%
  dplyr::mutate(time_of_step = cumsum(times)) %>%
  dplyr::select(time_of_step, dplyr::everything()) %>%
  dplyr::filter(time_of_step < t)
}

```

(ii) Assuming that initially the population of interest has 9 susceptible individuals, 1 infectious individual, and 0 removed individuals, print 5 realizations of label vectors, using $\beta = 0.8$, $\gamma = 1.5$, and $t = 2$. Print times of events and new label vectors at the event times.

Note that these print-outs are in the reverse orientation as the ones in the previous parts, so that we could include a time variable as a column.

```

init <- c("S", "S", "S", "S", "S", "S", "I", "S", "S", "S")

for (i in 1:5) {
  print(stringr::str_c("Realization ", i))
  individual_continuous(init, beta = 0.8, gamma = 1.5, t = 2) %>%
    print()
}

```

```

## [1] "Realization 1"
##      time_of_step person_1 person_2 person_3 person_4 person_5 person_6
## step_0      0.000000      S      S      S      S      S      S
## step_1      0.1018116      S      S      S      S      S      S
## step_2      0.1068633      S      S      S      S      S      S
## step_3      0.2013577      I      S      S      S      S      S
## step_4      0.2120748      I      S      S      S      S      I
## step_5      0.2269009      I      S      S      S      S      I
## step_6      0.2913914      I      S      S      S      I      I
## step_7      0.2988138      I      S      S      S      I      I
## step_8      0.4082589      I      S      S      I      I      I
## step_9      0.4105383      I      I      S      I      I      I
## step_10     0.4693244      I      I      S      I      I      R
## step_11     0.4866245      I      I      S      I      I      R
## step_12     0.5481570      I      I      S      I      I      R
## step_13     0.6281187      I      I      S      I      R      R
## step_14     0.6592227      I      I      S      R      R      R
## step_15     0.7823310      I      I      S      R      R      R
## step_16     1.0423026      I      R      S      R      R      R
## step_17     1.1515043      R      R      S      R      R      R
##      person_7 person_8 person_9 person_10
## step_0      I      S      S      S
## step_1      I      S      I      S
## step_2      I      I      I      S
## step_3      I      I      I      S
## step_4      I      I      I      S
## step_5      I      I      R      S
## step_6      I      I      R      S
## step_7      I      R      R      S

```

```

## step_8      I      R      R      S
## step_9      I      R      R      S
## step_10     I      R      R      S
## step_11     I      R      R      I
## step_12     I      R      R      R
## step_13     I      R      R      R
## step_14     I      R      R      R
## step_15     R      R      R      R
## step_16     R      R      R      R
## step_17     R      R      R      R
## [1] "Realization 2"
##      time_of_step person_1 person_2 person_3 person_4 person_5 person_6
## step_0  0.0000000      S      S      S      S      S      S
## step_1  0.1158235      S      S      S      S      S      S
##      person_7 person_8 person_9 person_10
## step_0      I      S      S      S
## step_1      R      S      S      S
## [1] "Realization 3"
##      time_of_step person_1 person_2 person_3 person_4 person_5 person_6
## step_0  0.0000000      S      S      S      S      S      S
## step_1  0.04314463      S      S      S      S      S      S
## step_2  0.09399832      S      S      S      S      I      S
## step_3  0.15029810      S      S      I      S      I      S
## step_4  0.17119009      S      I      I      S      I      S
## step_5  0.22120817      S      I      R      S      I      S
## step_6  0.29251809      S      I      R      S      I      I
## step_7  0.33108077      S      I      R      S      I      I
## step_8  0.38147594      S      I      R      S      I      I
## step_9  0.42266746      I      I      R      S      I      I
## step_10 0.48759432      I      I      R      I      I      I
## step_11 0.51994708      I      I      R      R      I      I
## step_12 0.54263899      I      I      R      R      I      I
## step_13 0.55617411      I      I      R      R      I      R
## step_14 0.79248469      I      R      R      R      I      R
## step_15 0.83471229      I      R      R      R      I      R
## step_16 1.43276566      I      R      R      R      I      R
## step_17 1.70309720      I      R      R      R      R      R
## step_18 1.95652079      I      R      R      R      R      R
##      person_7 person_8 person_9 person_10
## step_0      I      S      S      S
## step_1      I      I      S      S
## step_2      I      I      S      S
## step_3      I      I      S      S
## step_4      I      I      S      S
## step_5      I      I      S      S
## step_6      I      I      S      S
## step_7      I      I      S      I
## step_8      I      I      I      I
## step_9      I      I      I      I
## step_10     I      I      I      I
## step_11     I      I      I      I
## step_12     I      R      I      I
## step_13     I      R      I      I
## step_14     I      R      I      I

```

```

## step_15      I      R      I      R
## step_16      I      R      R      R
## step_17      I      R      R      R
## step_18      R      R      R      R
## [1] "Realization 4"
##      time_of_step person_1 person_2 person_3 person_4 person_5 person_6
## step_0  0.00000000      S      S      S      S      S      S
## step_1  0.09066563      S      S      S      S      S      S
##      person_7 person_8 person_9 person_10
## step_0      I      S      S      S
## step_1      R      S      S      S
## [1] "Realization 5"
##      time_of_step person_1 person_2 person_3 person_4 person_5 person_6
## step_0  0.00000000      S      S      S      S      S      S
## step_1  0.003515907      S      S      S      S      S      I
## step_2  0.068351228      S      S      S      S      S      I
## step_3  0.091865981      I      S      S      S      S      I
## step_4  0.129989929      I      I      S      S      S      I
## step_5  0.155579897      I      I      S      S      S      I
## step_6  0.187606567      I      I      I      S      S      I
## step_7  0.196011095      I      I      R      S      S      I
## step_8  0.208681309      I      I      R      S      S      R
## step_9  0.304004775      I      I      R      I      S      R
## step_10 0.412135333      I      I      R      I      S      R
## step_11 0.440409175      I      I      R      I      S      R
## step_12 0.458189887      I      R      R      I      S      R
## step_13 0.604814121      I      R      R      I      I      R
## step_14 0.676312642      R      R      R      I      I      R
## step_15 0.703244270      R      R      R      I      I      R
## step_16 0.784671992      R      R      R      I      I      R
## step_17 0.807670084      R      R      R      I      R      R
## step_18 1.349280346      R      R      R      R      R      R
##      person_7 person_8 person_9 person_10
## step_0      I      S      S      S
## step_1      I      S      S      S
## step_2      I      S      S      I
## step_3      I      S      S      I
## step_4      I      S      S      I
## step_5      I      S      I      I
## step_6      I      S      I      I
## step_7      I      S      I      I
## step_8      I      S      I      I
## step_9      I      S      I      I
## step_10     I      I      I      I
## step_11     I      I      I      R
## step_12     I      I      I      R
## step_13     I      I      I      R
## step_14     I      I      I      R
## step_15     I      R      I      R
## step_16     I      R      R      R
## step_17     I      R      R      R
## step_18     I      R      R      R

```

(c) Continuous time compartmental modeling

(i) Write a function that simulates a fixed number of time units of a continuous time compartmental SIR model. Your function should have the following user specified arguments: initial vector of S , I , and R counts, infection rate β , removal rate γ , and the number of time units t .

```
compartiment_continuous <- function(init_counts, beta, gamma, t) {
  times <- c(0)
  res <- data.frame(
    time = 0,
    S = init_counts[1],
    I = init_counts[2],
    R = init_counts[3]
  )
  step <- 1
  while (sum(times) < t) {
    i <- rexp(1, rate = beta * res$S[step] * res$I[step])
    r <- rexp(1, rate = gamma * res$I[step])
    if (!is.na(i) & !is.na(r) & i < r) {
      new_res <- data.frame(
        time = i + sum(times),
        S = res$S[step] - 1,
        I = res$I[step] + 1,
        R = res$R[step]
      )
      times <- append(times, i)
    } else if (!is.na(r)) {
      new_res <- data.frame(
        time = r + sum(times),
        S = res$S[step],
        I = res$I[step] - 1,
        R = res$R[step] + 1
      )
      times <- append(times, r)
    } else {
      break
    }
    step <- step + 1
    res <- rbind(res, new_res)
  }
  res %>%
    rbind(
      data.frame(
        time = t,
        S = res$S[nrow(res)],
        I = res$I[nrow(res)],
        R = res$R[nrow(res)]
      )
    )
}
```

(ii) Assuming that initially the population of interest has 990 susceptible individuals, 10 infectious individuals, and 0 removed individuals, plot 5 realizations of SIR trajectories using $\beta = 0.008$, $\gamma = 3.5$, and $t = 4$.

```

library(ggplot2)
init_labels <- c(990, 10, 0)

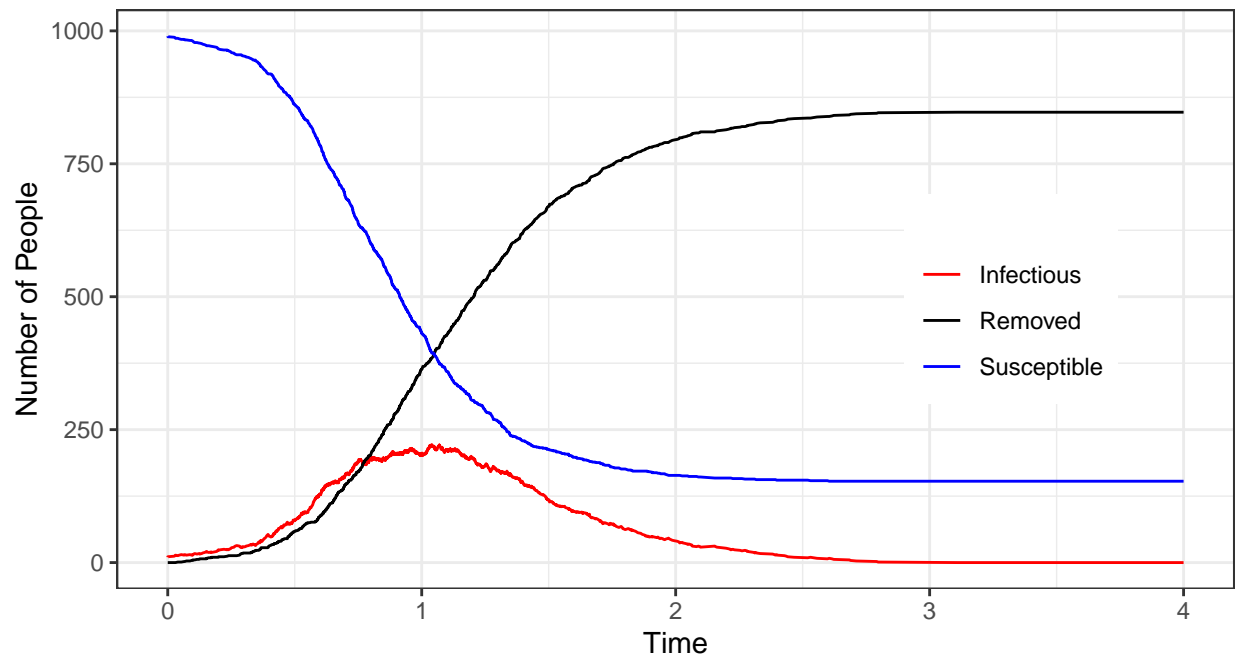
for (i in 1:5) {
  p <- compartment_continuous(init_labels, beta = 0.008, gamma = 3.5, t = 4) %>%
    tidyr::pivot_longer(
      cols = c("S", "I", "R"),
      names_to = "label",
      values_to = "value"
    ) %>%
    dplyr::mutate(
      label = dplyr::case_when(
        label == "S" ~ "Susceptible",
        label == "I" ~ "Infectious",
        label == "R" ~ "Removed"
      )
    ) %>%
    ggplot() +
    aes(x = time, y = value, color = label) +
    geom_line() +
    scale_color_manual(values = c("red", "black", "blue")) +
    xlab("Time") +
    ylab("Number of People") +
    ggtitle(
      stringr::str_c("Realization ", i, " of Compartmental SIR"),
      "Beta = 0.008, Gamma = 3.5, t = 4"
    ) +
    labs(color = "") +
    theme_bw() +
    theme(legend.position = c(0.8, 0.5))

  print(p)
}

```

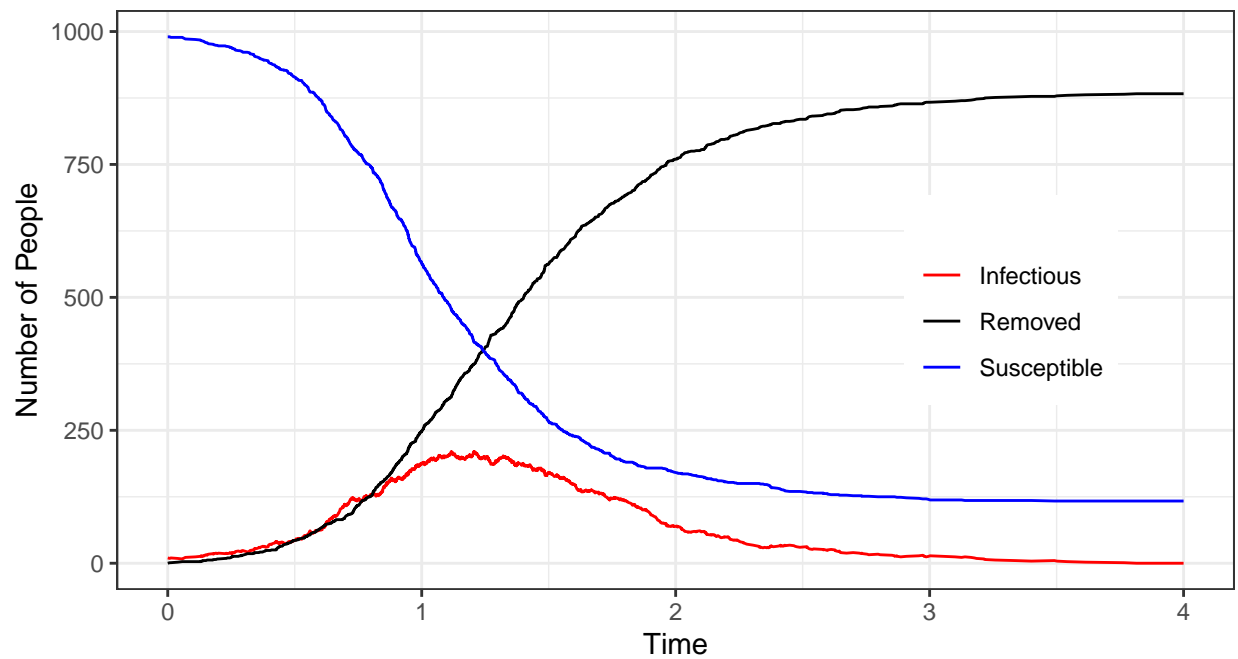

Realization 1 of Compartmental SIR

Beta = 0.008, Gamma = 3.5, t = 4



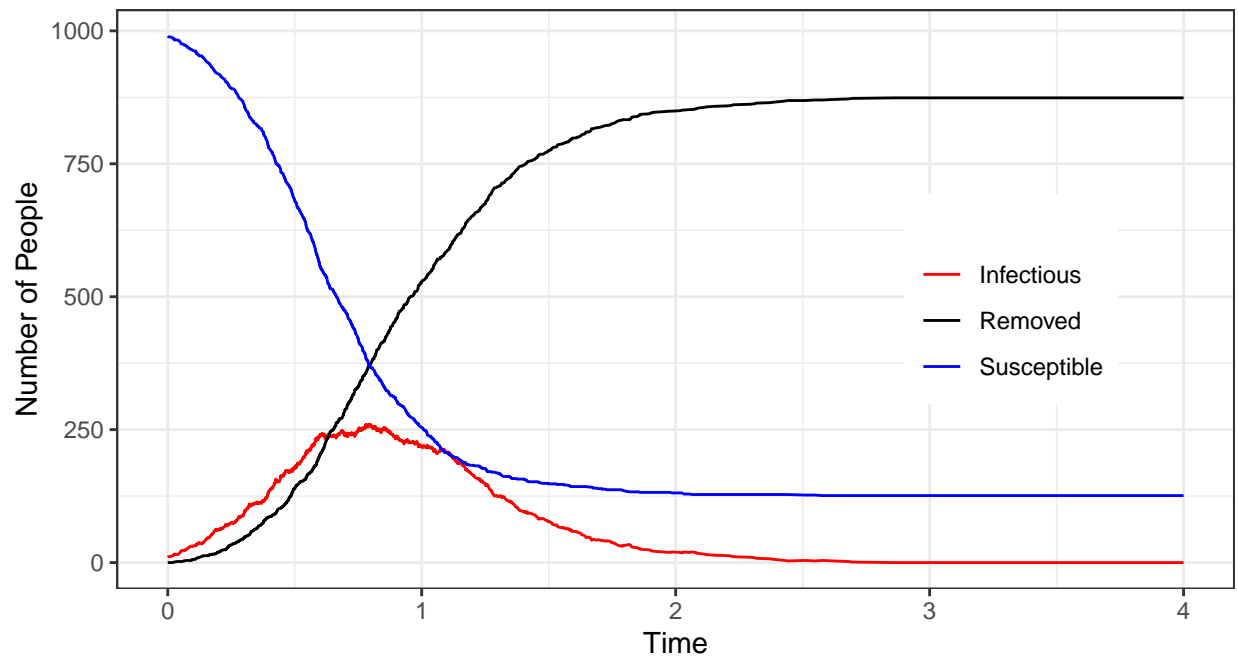
Realization 2 of Compartmental SIR

Beta = 0.008, Gamma = 3.5, t = 4



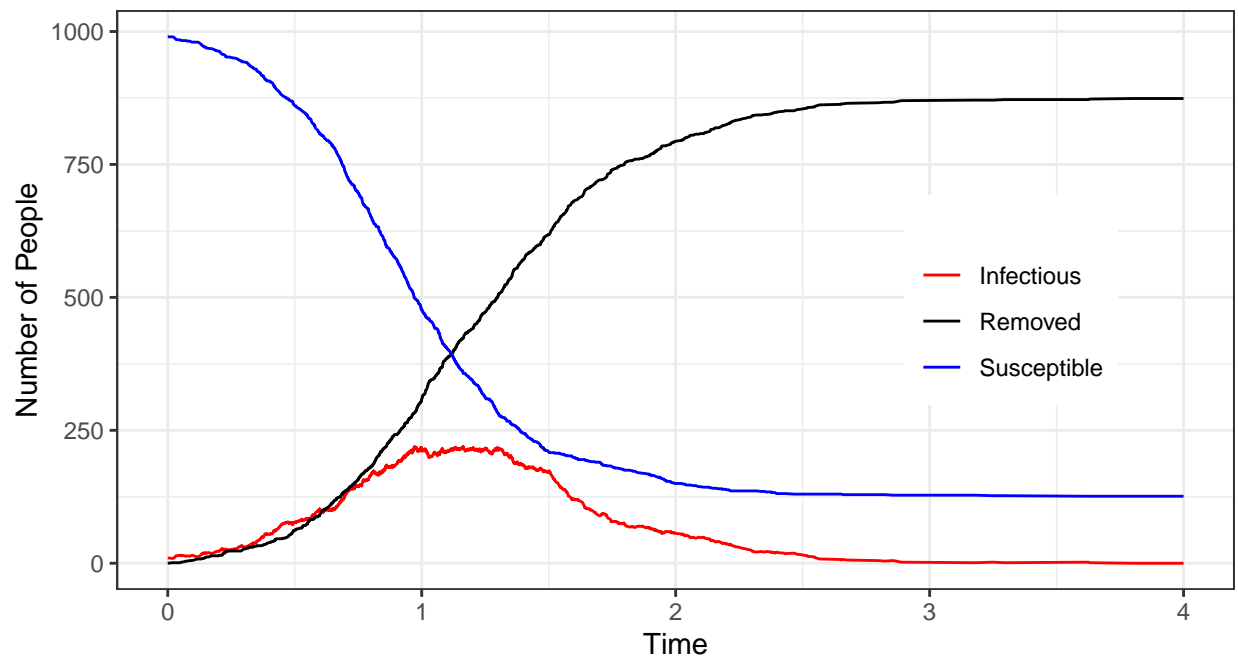
Realization 3 of Compartmental SIR

Beta = 0.008, Gamma = 3.5, t = 4



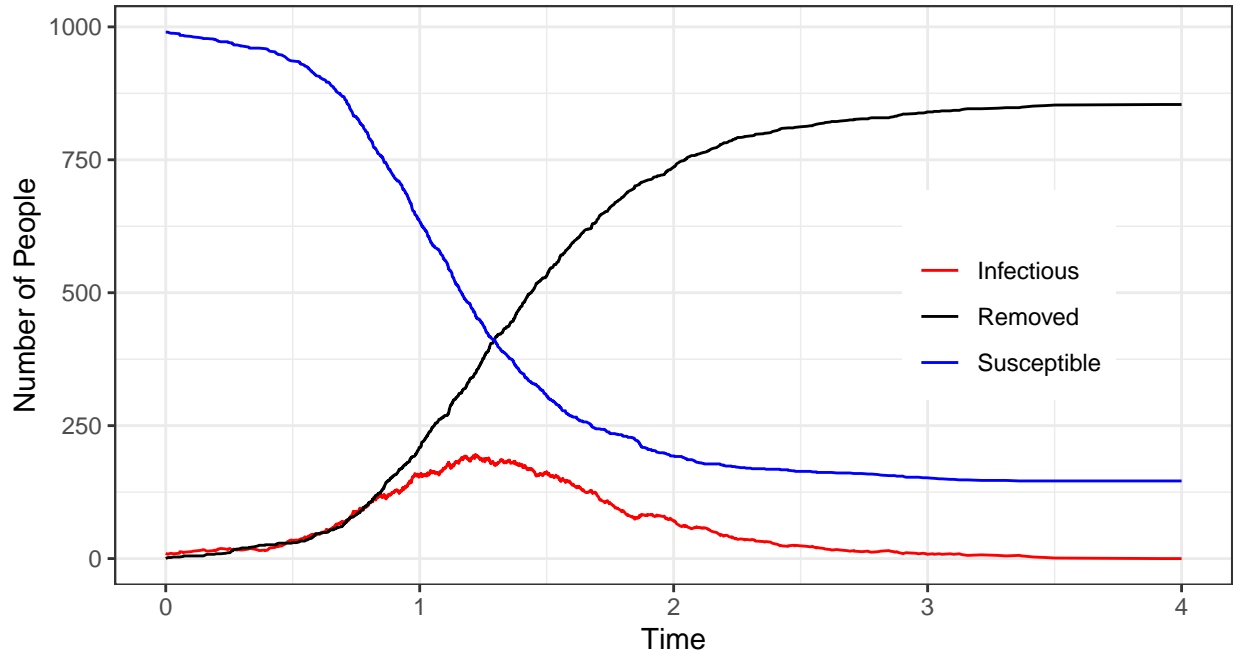
Realization 4 of Compartmental SIR

Beta = 0.008, Gamma = 3.5, t = 4



Realization 5 of Compartmental SIR

Beta = 0.008, Gamma = 3.5, t = 4



Problem 2

Formulate an extension of an SIR ODE model with vaccinated and unvaccinated compartments. Vaccinated individuals should be able to get infected and to transmit the disease.

(a) Write down the differential equations

First, we will define several functions based on the vaccinated vs. unvaccinated compartments:

- $V_S(t)$ is the number of susceptibles who are vaccinated
- $U_S(t)$ is the number of susceptibles who are unvaccinated
- $V_I(t)$ is the number of infectious who are vaccinated
- $U_I(t)$ is the number of infectious who are unvaccinated
- $V_R(t)$ is the number of removed who are vaccinated
- $U_R(t)$ is the number of removed who are unvaccinated

This leads to the overall functions:

- $S(t) = V_S(t) + U_S(t)$ is the total number of susceptibles in the population
- $I(t) = V_I(t) + U_I(t)$ is the total number of infectious in the population
- $R(t) = V_R(t) + U_R(t)$ is the total number of removed in the population

We can write the following differential equations for the vaccinated/unvaccinated compartments:

$$\begin{aligned}
\frac{dV_S(t)}{dt} &= -\beta_1 V_S(t)U_I(t) - \beta_2 V_S(t)V_I(t) \\
\frac{dU_S(t)}{dt} &= -\beta_3 U_S(t)U_I(t) - \beta_4 U_S(t)V_I(t) \\
\frac{dV_I(t)}{dt} &= \beta_1 V_S(t)U_I(t) + \beta_2 V_S(t)V_I(t) - \gamma_1 V_I(t) \\
\frac{dU_I(t)}{dt} &= \beta_3 U_S(t)U_I(t) + \beta_4 U_S(t)V_I(t) - \gamma_2 U_I(t) \\
\frac{dV_R(t)}{dt} &= \gamma_1 V_I(t) \\
\frac{dU_R(t)}{dt} &= \gamma_2 U_I(t)
\end{aligned}$$

And for the overall counts, we see that since derivatives are additive, we have

$$\begin{aligned}
\frac{dS(t)}{dt} &= \frac{dV_S(t)}{dt} + \frac{dU_S(t)}{dt} = -\beta_1 V_S(t)U_I(t) - \beta_2 V_S(t)V_I(t) - \beta_3 U_S(t)U_I(t) - \beta_4 U_S(t)V_I(t) \\
\frac{dI(t)}{dt} &= \frac{dV_I(t)}{dt} + \frac{dU_I(t)}{dt} \\
&= \beta_1 V_S(t)U_I(t) + \beta_2 V_S(t)V_I(t) - \gamma_1 V_I(t) + \beta_3 U_S(t)U_I(t) + \beta_4 U_S(t)V_I(t) - \gamma_2 U_I(t) \\
\frac{dR(t)}{dt} &= \frac{dV_R(t)}{dt} + \frac{dU_R(t)}{dt} = \gamma_1 V_I(t) + \gamma_2 U_I(t)
\end{aligned}$$

(b) Provide interpretations of all parameters

We can interpret the parameters as follows:

- β_1 is the transmission rate from unvaccinated people to vaccinated people
- β_2 is the transmission rate from vaccinated people to vaccinated people
- β_3 is the transmission rate from unvaccinated people to unvaccinated people
- β_4 is the transmission rate from vaccinated people to unvaccinated people
- γ_1 is the recovery rate of vaccinated individuals
- γ_2 is the recovery rate of unvaccinated individuals

(c) What parameters would allow you to measure vaccine efficacy against infection and transmission if you were able to estimate them from the data?

Estimating $\beta_1, \beta_2, \beta_3, \beta_4$ would allow you to measure vaccine efficacy against infection and transmission. For instance, β_1 and β_2 give you information on how likely it is for vaccinated people to become infected, while β_3 and β_4 give you information on how likely it is for unvaccinated people to become infected. Looking at it from the other way, β_1 and β_3 give you information on how likely it is that unvaccinated people transmit the disease, while β_2 and β_4 give you information on how likely it is that vaccinated people transmit the disease.