STAT 601.29: Bayesian Infectious Disease Modelling Assignment 1 Winter 2023

Hao Nan Wang UCID: 30185958

2023-02-12

Contents

Question 1: SIR Model

```
# Import the data and look at the first three rows
q1cases = read.csv(file = 'Q1Cases.csv', header = TRUE)
head(q1cases,3)
```

```
##   week newcases
## 1    1        0
## 2    2        2
## 3    3        5
```

i)

```
# convert the new case data to prevalence data as the infectious period is two weeks long
result = c()
prev = function(arr,week){
  for (i in 1:length(arr)) {
  result[i] = sum(arr[i:(i+week-1)])
  #print(result[i])
  i = i+1
  }
  result = append(arr[1:week-1],na.omit(result))
  return(result)
}

V1 = prev(q1cases$newcases,1)
V2 = prev(q1cases$newcases,2)
V3 = prev(q1cases$newcases,3)

q1cases = as.data.frame(cbind(q1cases,V1,V2,V3))
q1cases
```

```
##    week newcases  V1  V2  V3
## 1     1        0   0   0   0
## 2     2        2   2   2   2
## 3     3        5   5   7   7
## 4     4        9   9  14  16
## 5     5        8   8  17  22
## 6     6       13  13  21  30
## 7     7       25  25  38  46
## 8     8       41  41  66  79
## 9     9       71  71 112 137
## 10   10      138 138 209 250
## 11   11      238 238 376 447
## 12   12      367 367 605 743
```

ii)

In frequency-dependent model, there are some assumptions lead us to a set of three ordinary differential equations for S_t, I_t and R_t:

$$\frac{dS_t}{dt} = -\beta\frac{S_tI_t}{N}; \frac{dI_t}{dt} = \beta\frac{S_tI_t}{N} - \gamma I_t; \frac{dS_t}{dt} = \gamma I_t$$

There is one unrecorded case who becomes infectious at the beginning of week 0, $S = N = 12000$, and $I(0) = 1, R(0) = 0$, so that $\frac{dI}{dt} = \beta\frac{SI}{N} - \gamma I = (\beta - \gamma)I$

Everyone should know that the general solution of ODE is $y = Ce^x$, and the question also gives $I(t = 0) = 1$ thus we could have

$$I(t) = Ce^{(\beta-\gamma)t}$$

where $C = 1$

Eventually, to estimate the transmision rate β, we have the objective function

$$\min\Sigma_t\left(e^{(\beta-\gamma)t} - \hat{I}(t)\right)^2$$

```
library(deSolve)
```

```
## Warning: package 'deSolve' was built under R version 4.2.2
```

```
## 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)))
  })
}
```

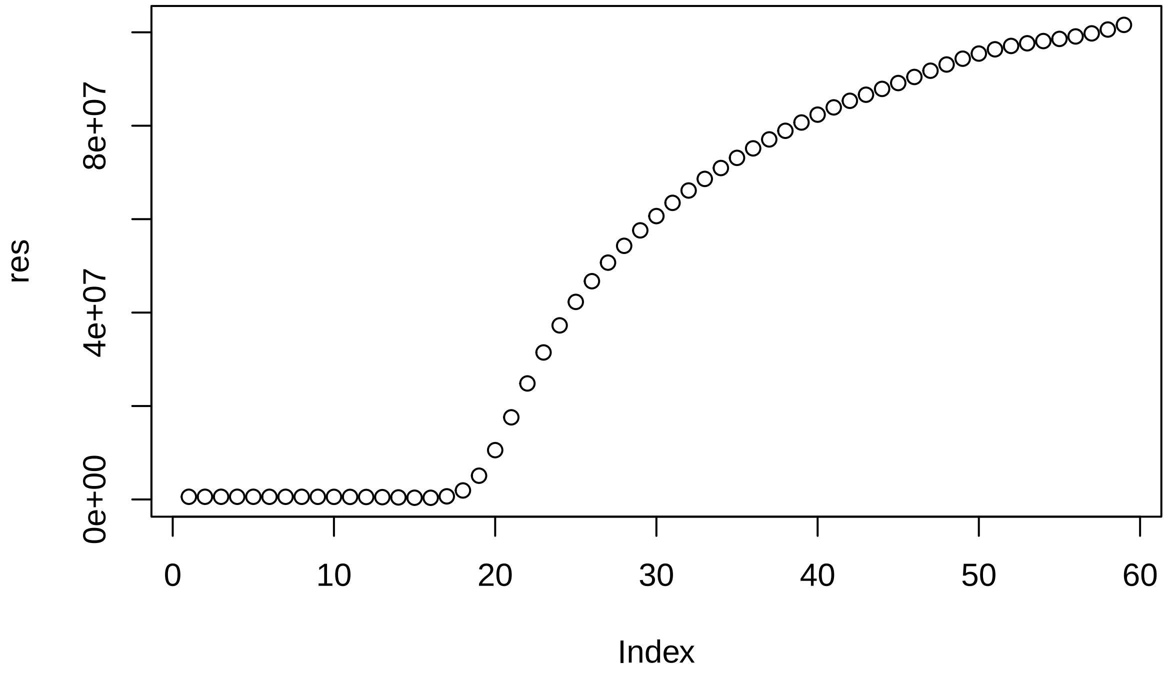
```
## Define the residuals for the optimization
residuals <- function(params, data) {
  init <- c(S = 12000, I = 1, R = 0)
  times <- seq(0, 12, by = 1)
  out = ode(y = init,times = times, func = SIR, parms = params)
  observed <- q1cases$V2
  predicted <- out[, "I"]
  return(sum((predicted - observed)^2))
}
```

```
# define a range of possible beta values
beta_start_values <- seq(0.00001, 0.0003, by = 0.000005)
# Initialize a vector to store the residual results
res = c()
# Loop over the starting values for beta
for (i in 1:length(beta_start_values)) {
  # Define the starting values for beta and gamma
  parameters <- c(beta = beta_start_values[i], gamma = 0.5)
  res[i] = residuals(parameters, data = q1cases$V2)
}
```

```
# Find the index of the minimum residuals
min_result_index <- which.min(res)
min_result <- res[[min_result_index]]
# Print the minimum residuals
min_result
```

```
## [1] 357053.9
```

```
# Plot the residuals under different values of beta
plot(res)
```



```
# Print the corresponding beta value
beta = beta_start_values[[min_result_index]]
beta
```

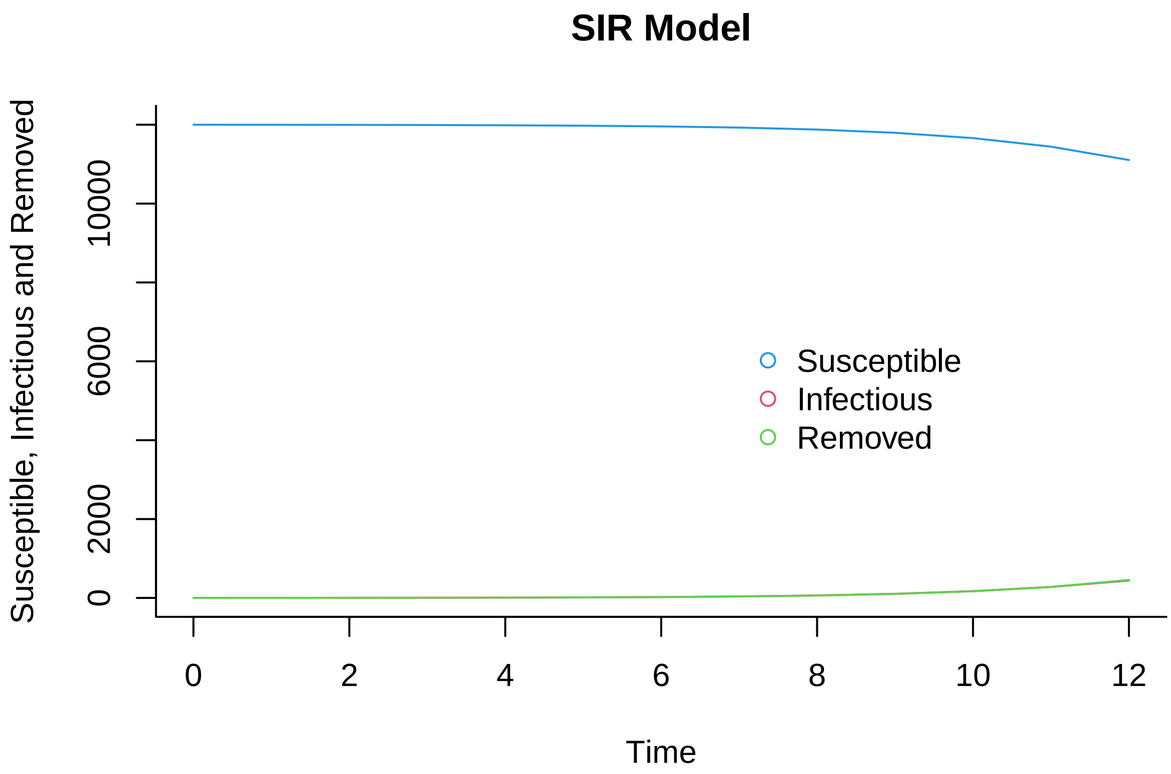
```
## [1] 8.5e-05
```

When infectious period is two weeks long, the corresponding transmission rate $\beta = 0.000085$

```
## plot the data
init <- c(S = 12000, I = 1, R = 0)
times <- seq(0, 12, by = 1)
parameters = c(beta = beta, gamma = 1/2)
out = ode(y = init,times = times, func = SIR, parms = parameters)
out = as.data.frame(out)
out$time = NULL
head(out,3)
```

```
##          S        I         R
## 1 12000.00 1.000000 0.0000000
## 2 11998.66 1.681941 0.6557845
## 3 11996.41 2.828504 1.7586927
```

```
matplot(x = times, y = out, type = "l", xlab = "Time",
        ylab = "Susceptible, Infectious and Removed", main = "SIR Model", lwd = 1, lty = 1,
        bty = "l", col = c(4,2,3))
legend(7, 7000, c("Susceptible", "Infectious", "Removed"), pch = 1, col = c(4,2,3), bty = "n")
```



iii)

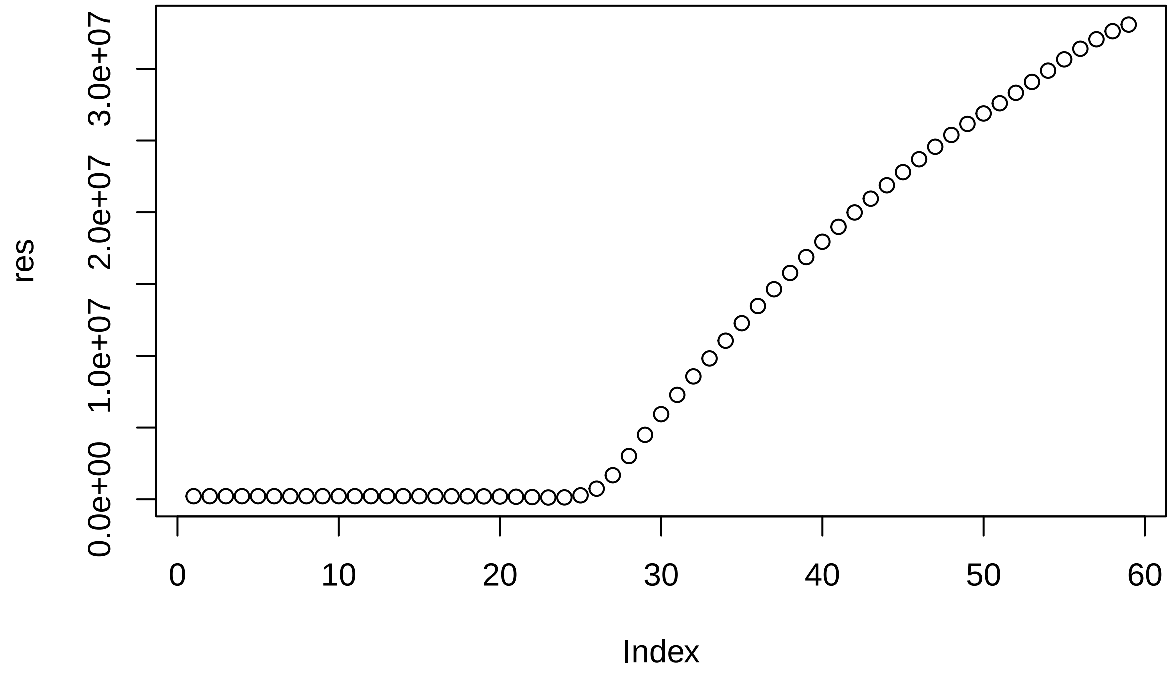
When infectious period is one week long

```
## Define the residuals for the optimization
residuals <- function(params, data) {
  init <- c(S = 12000, I = 1, R = 0)
  times <- seq(0, 12, by = 1)
  out = ode(y = init,times = times, func = SIR, parms = params)
  observed <- q1cases$V1
  predicted <- out[, "I"]
  return(sum((predicted - observed)^2))
}
```

```
# define a range of possible beta values
beta_start_values <- seq(0.00001, 0.0003, by = 0.000005)
# Initialize a vector to store the residual results
res = c()
# Loop over the starting values for beta
for (i in 1:length(beta_start_values)) {
  # Define the starting values for beta and gamma
  parameters <- c(beta = beta_start_values[i], gamma = 1)
  res[i] = residuals(parameters, data = q1cases$V1)
}

# Find the index of the minimum residuals
min_result_index <- which.min(res)
min_result <- res[[min_result_index]]
# Print the minimum residuals
min_result
```

```
## [1] 127848.4
```

```
# Plot the residuals under different values of beta
plot(res)
```



```
# Print the corresponding beta value
beta = beta_start_values[[min_result_index]]
beta
```

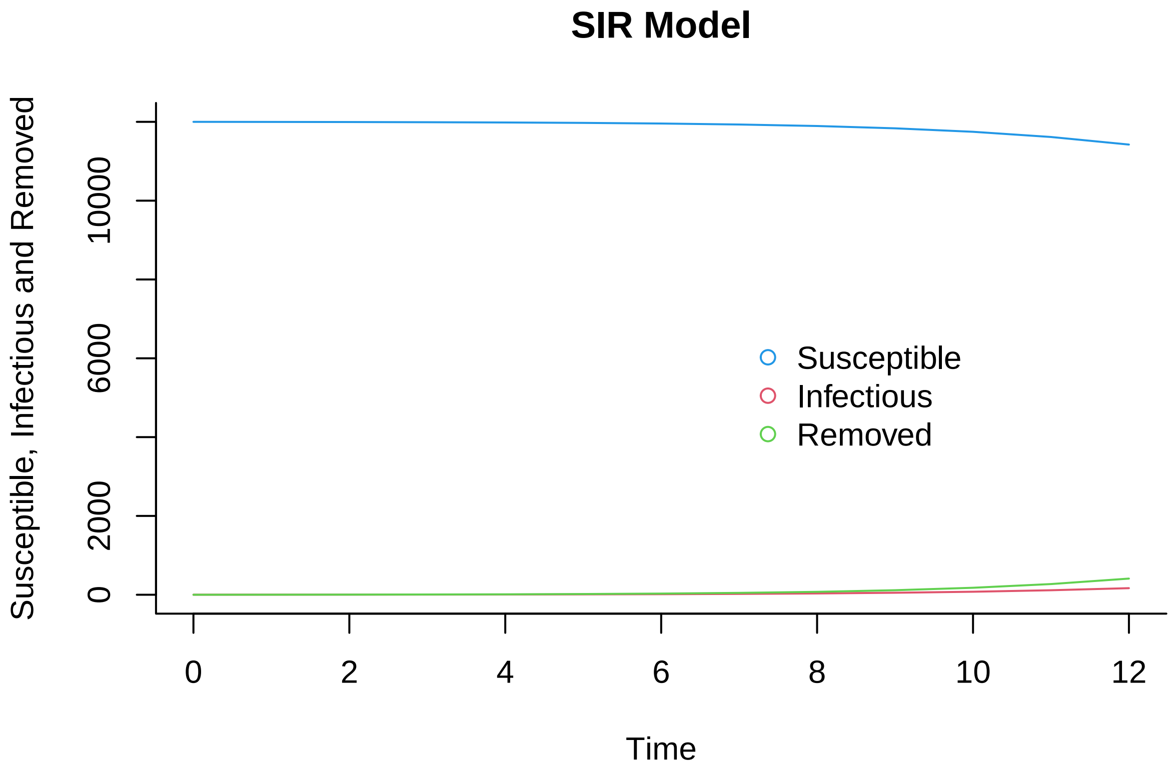
```
## [1] 0.00012
```

When infectious period is one week long, the corresponding transmission rate $\beta = 0.00012$

```
## plot the data
init <- c(S = 12000, I = 1, R = 0)
times <- seq(0, 12, by = 1)
parameters = c(beta = beta, gamma = 1)
out = ode(y = init,times = times, func = SIR, parms = parameters)
out = as.data.frame(out)
out$time = NULL
head(out,3)
```

```
##          S        I        R
## 1 12000.00 1.000000 0.000000
## 2 11998.19 1.552551 1.256108
## 3 11995.38 2.409759 3.206015
```

```
matplot(x = times, y = out, type = "l", xlab = "Time",
        ylab = "Susceptible, Infectious and Removed", main = "SIR Model", lwd = 1, lty = 1,
        bty = "l", col = c(4,2,3))
legend(7, 7000, c("Susceptible", "Infectious", "Removed"), pch = 1, col = c(4,2,3), bty = "n")
```



When infectious period is three weeks long

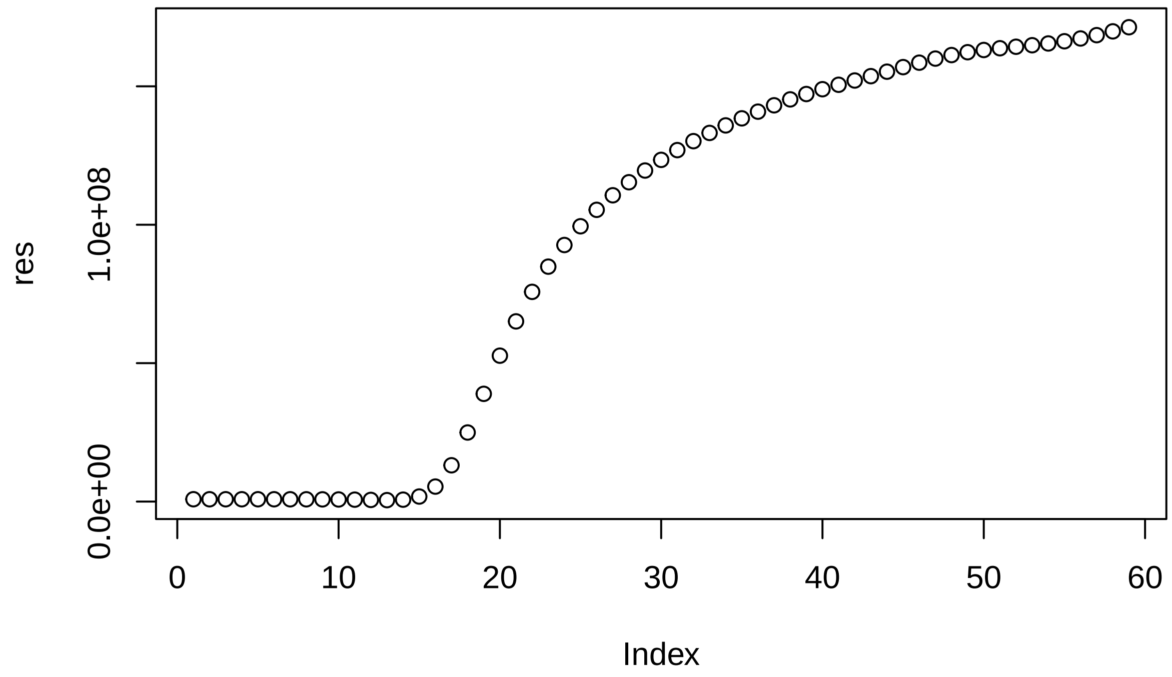
```
## Define the residuals for the optimization
residuals <- function(params, data) {
  init <- c(S = 12000, I = 1, R = 0)
  times <- seq(0, 12, by = 1)
  out = ode(y = init,times = times, func = SIR, parms = params)
  observed <- q1cases$V3
  predicted <- out[, "I"]
  return(sum((predicted - observed)^2))
}
```

```
# define a range of possible beta values
beta_start_values <- seq(0.00001, 0.0003, by = 0.000005)
# Initialize a vector to store the residual results
res = c()
# Loop over the starting values for beta
for (i in 1:length(beta_start_values)) {
  # Define the starting values for beta and gamma
  parameters <- c(beta = beta_start_values[i], gamma = 1/3)
  res[i] = residuals(parameters, data = q1cases$V3)
}

# Find the index of the minimum residuals
min_result_index <- which.min(res)
min_result <- res[[min_result_index]]
```

```
# Print the minimum residuals
min_result
```

```
## [1] 522852.6
```

```
# Plot the residuals under different values of beta
plot(res)
```



```
# Print the corresponding beta value
beta = beta_start_values[[min_result_index]]
beta
```

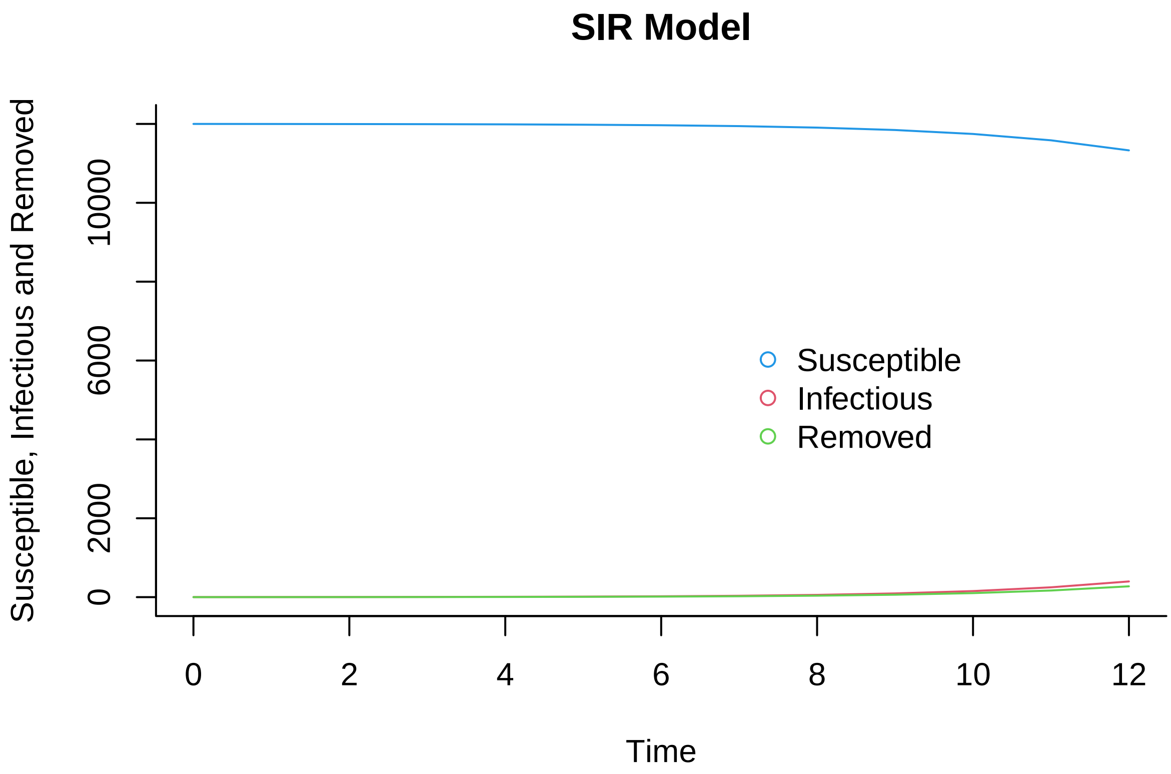
```
## [1] 7e-05
```

When infectious period is one week long, the corresponding transmission rate $\beta = 0.00007$

```
## plot the data
init <- c(S = 12000, I = 1, R = 0)
times <- seq(0, 12, by = 1)
parameters = c(beta = beta, gamma = 1/3)
out = ode(y = init,times = times, func = SIR, parms = parameters)
out = as.data.frame(out)
out$time = NULL
head(out,3)
```

```
##          S        I         R
## 1 12000.00 1.000000 0.0000000
## 2 11998.91 1.659693 0.4340411
## 3 11997.09 2.754305 1.1543799
```

```
matplot(x = times, y = out, type = "l", xlab = "Time",
        ylab = "Susceptible, Infectious and Removed", main = "SIR Model", lwd = 1, lty = 1,
        bty = "l", col = c(4,2,3))
legend(7, 7000, c("Susceptible", "Infectious", "Removed"), pch = 1, col = c(4,2,3), bty = "n")
```



iv

From part ii) and iii), we determine $\xi = \Sigma_t \left(e^{(\beta-\gamma)t} - \hat{I}(t)\right)^2$, the sum of residual squared between the estimate and the real value. When infectious period is 1 week long, ξ is 127848; When infectious period is 2 weeks long, ξ is 357054; When infectious period is 3 weeks long, ξ is 522853. Infectious period of one week produces the smallest ξ among them, so I believe that the infectious period of one week is most likely correct; However, we still can see a huge ξ over there. This method is not the optimal way to solve this particular question because it determines one parameter with the assumption of another parameter is fixed. So the region of parameter candidates is not correct / smaller than it should be, thus, it is hard to find the correct parameter. The correct way to determine a vector of parameters is to set their COMBINATION or use nested loop to try all possibilities.

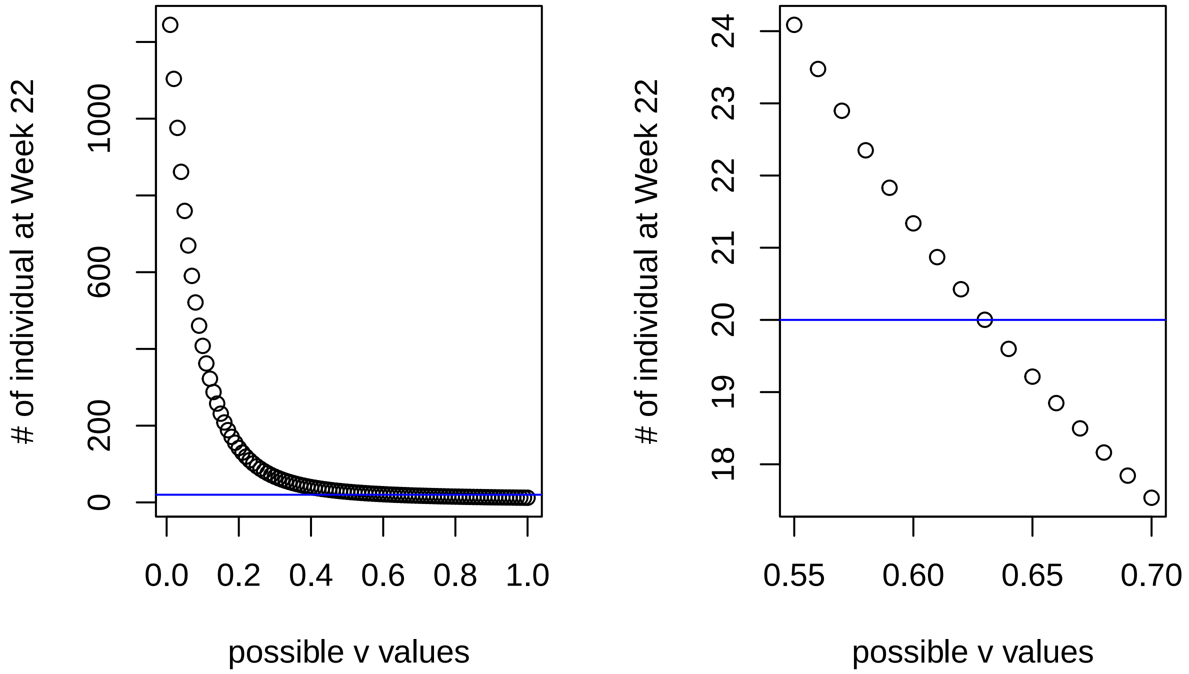
Question 2: SIR Model with Vaccination

```
## we modify our SIR model, called SIRV, as now there are some susceptible are now
## being removed rather than being infected, via the vaccination program
library(deSolve)
## SIR Function
SIRV = function(time, state, parameters){
  with(as.list(c(state,parameters)),{
    dS = (-beta*S*I) - V*S
    dI = (beta*S*I)-(gamma*I)
    dR = gamma*I + V*S
    return(list(c(dS,dI,dR)))
  })
}
## Initialization
## at the beginning of week 13 obtain from ii)
init = c(S=11104.88, I=440.107213, R=456.0110071)
## define a range of possible v values
v_start_values <- seq(0.01, 1, by = 0.01)
## define an empty set to store v values that corresponding to the 10th week of vaccination
v = c()
## loop
for (i in 1:length(v_start_values)) {
  parameters = c(beta=8.5e-05, gamma=1/2, V=v_start_values[i])
  times = seq(13,22,by=1)
  ## Solve using ODE
  v[i] = ode(y=init, times=times, func=SIRV, parms=parameters)[10,3]
}
v_start_values[Position(function(x) x < 20, v)]
```

```
## [1] 0.64
```

At 10 weeks after the policy is introduced (Week 22), we test different v values (from 0.01 to 1.00), we see when $v = 0.63$, the number of infectious individual meets the bar/threshold of "under control" and starting $v = 0.64$, the number of infectious individual is totally below 20. See the two graphs illustrated on next page.

```
par(mfrow=c(1,2))
plot(x=v_start_values, y=v, xlab = "possible v values", ylab = "# of individual at Week 22")
abline(h=20, col="blue")
plot(x=seq(0.55,0.70,by=0.01), y=v[55:70],xlab = "possible v values",
     ylab = "# of individual at Week 22")
abline(h=20, col="blue")
```



# of individual at Week 22
0
200
600
1000
0.0
0.2
0.4
0.6
0.8
1.0
possible v values
# of individual at Week 22
18
19
20
21
22
23
24
0.55
0.60
0.65
0.70
possible v values

Question 3: Two Populations (Parasite and Host)

Some important assumptions given in the question:

1. The disease follows an SIR and an SI compartmental model for Human and Mosquito population,respectively.
2. Mosquitoes can infect humans but not other mosquitoes; vice versa. Also the infection rates between the two species to differ.
3. The infectious period for human is 10 time units and at time zero, we have $N_h(0) = 1000$, $S_h(0) = 999$ and $I_h(0) = 1$ and $N_m(0) = S_m(0) = 1000$
4. Assume a density-dependent model
5. Assume constant birthrate and constant death rate for human population, but these two rates are different
6. Assume constant birthrate and constant death rate for mosquito population, but these two rates are different

Assuming that the total human population size is N=1000, the total mosquito population size is M=1000, and the infectious period for humans is T=10, the equations for the population dynamics of the two species can be described as three ODE as follows:

Human population dynamics:

$$dS/dt = \nu(S + I + R) - \beta S I_m - \mu S$$
$$dI/dt = \beta S I_m - \gamma I - \mu I$$
$$dR/dt = \gamma I - \mu R$$

Mosquito population dynamics:

$$dS_m/dt = \phi(S_m + I_m) - \alpha S_m I - \psi S_m$$
$$dI_m/dt = \alpha S_m I - \psi I$$

where β is the transmission rate from humans to mosquitoes, α is the transmission rate from mosquitoes to humans, S is the number of susceptible humans, I is the number of infected humans, R is the number of recovered humans, S_m is the number of susceptible mosquitoes, and I_h is the number of infected humans. Also, ν is the birth rate and μ is the natural death rate for human and ϕ is the birth rate and ψ is the natural death rate for mosquito .

```
library(deSolve)
## Define the ODE model
ode_model <- function(t, y, parameters) {
  with(as.list(c(y, parameters)), {
    dS <- nu*(S+I+R) -beta * S * I_m - mu * S
    dI <- beta * S * I_m - gamma * I - mu* I
    dR <- gamma * I - mu * R
    dS_m <- phi*(S+I+R) -alpha * S_m * I - psi * S_m
    dI_m <- alpha * S_m * I - psi * I_m
    return(list(c(dS, dI, dR, dS_m, dI_m)))
  })
}
```

i)

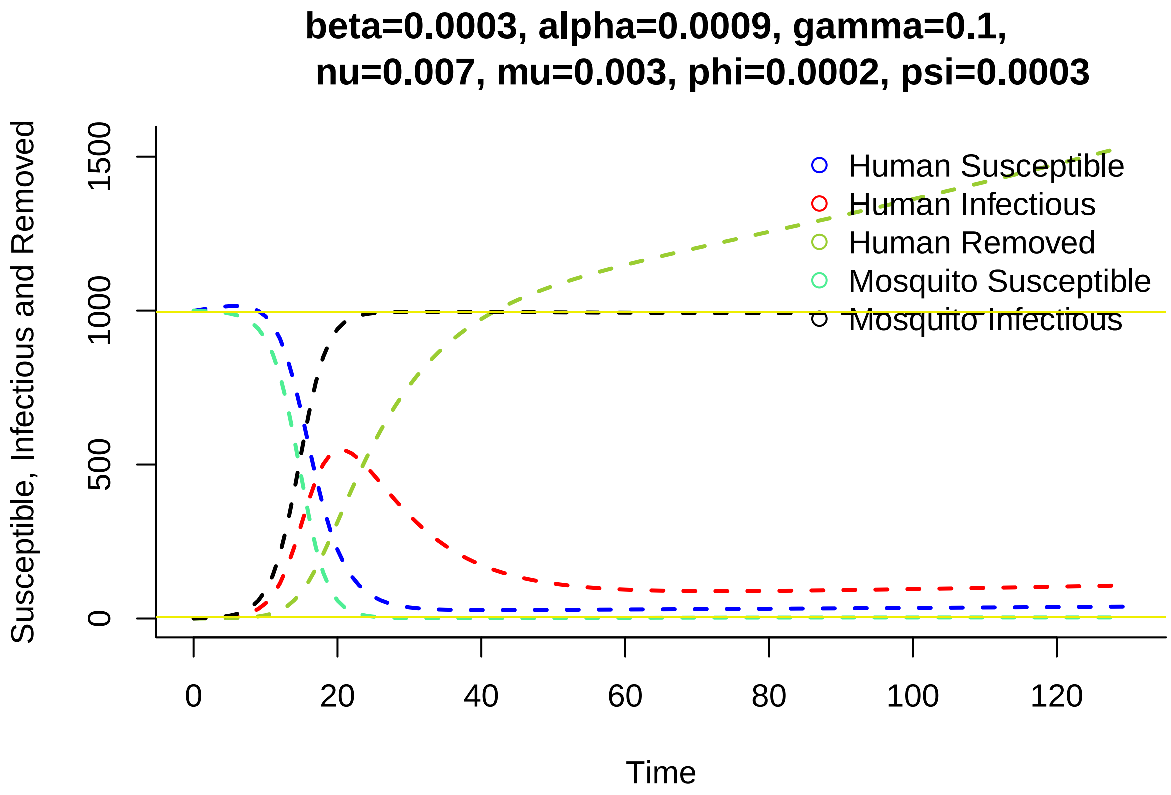
```r
## Initialization
initial_conditions <- c(S = 999, I = 1, R = 0, S_m = 1000, I_m = 0)
parameters <- c(beta = 0.0003, alpha = 0.0009, N = 1000, M = 1000, gamma = 0.1,
                nu = 0.007, mu = 0.003, phi = 0.0002, psi = 0.0003)
times <- seq(from = 0, to = 130, by = 1)

# result in an equilibrium state with higher prevalence in mosquitoes
out <- ode(y = initial_conditions, times = times, func = ode_model, parms = parameters)
head(out,2)
```

```
##      time        S        I          R       S_m       I_m
## [1,]    0  999.000 1.000000 0.00000000 1000.0000 0.0000000
## [2,]    1 1002.878 1.031243 0.09921042  999.0068 0.8936488
```

```r
## plot
out = as.data.frame(out)
out$time =NULL
tail(out,2)
```

```
##            S        I        R      S_m      I_m
## 130 38.51345 107.0966 1529.703 3.458735 991.7561
## 131 38.66659 107.5253 1535.836 3.459356 991.7926
```

```r
matplot(x = times, y = out, type = "l", main = "beta=0.0003, alpha=0.0009, gamma=0.1,
        nu=0.007, mu=0.003, phi=0.0002, psi=0.0003", xlab = "Time",
        ylab = "Susceptible, Infectious and Removed", lwd = 2, lty = 2,
        bty = "l", col = c("blue", "red", "yellowgreen", "seagreen2", "black"))
legend("topright", c("Human Susceptible", "Human Infectious", "Human Removed",
                  "Mosquito Susceptible", "Mosquito Infectious"), pch = 1,
       col = c("blue", "red", "yellowgreen", "seagreen2", "black"), bty = "n")
abline(h=5, col="yellow2")
abline(h=995, col="yellow2")
```



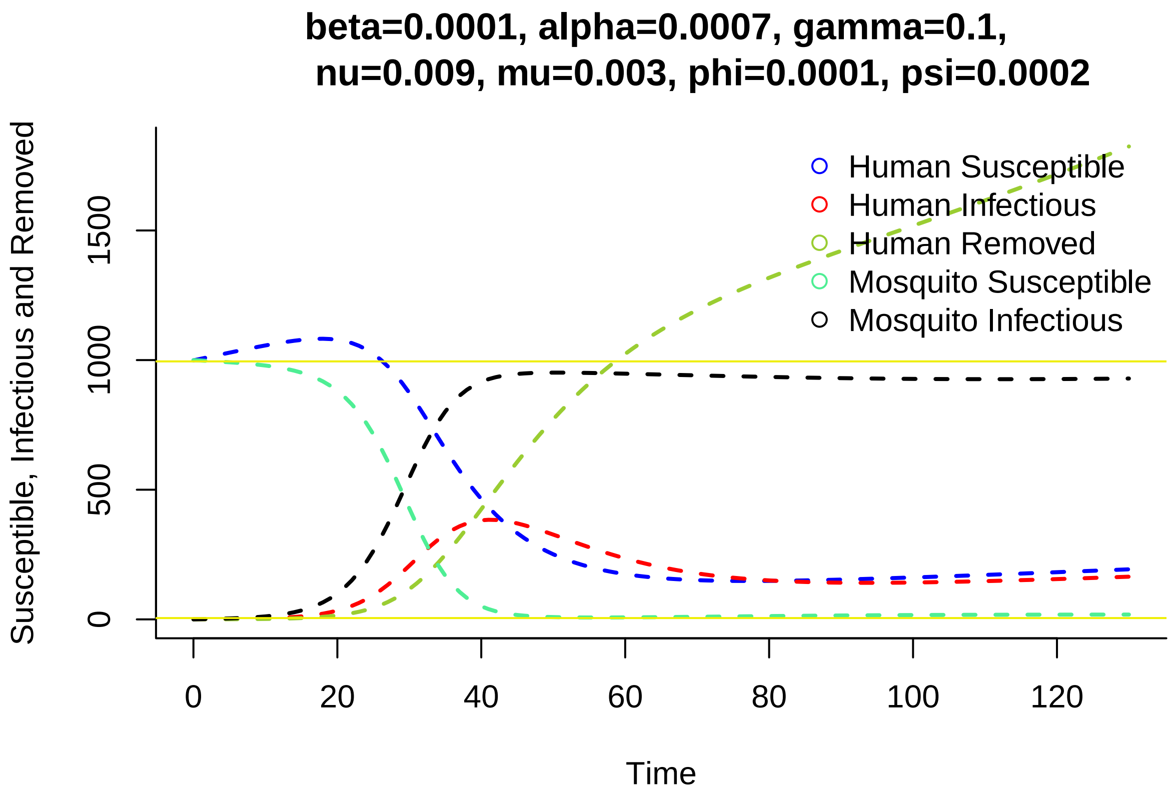
```r
## Initialization
initial_conditions <- c(S = 999, I = 1, R = 0, S_m = 1000, I_m = 0)
parameters <- c(beta = 0.0001, alpha = 0.0007, N = 1000, M = 1000, gamma = 0.1,
                nu = 0.009, mu = 0.003, phi = 0.001, psi = 0.002)
times <- seq(from = 0, to = 130, by = 1)

# result in an equilibrium state with higher prevalence in mosquitoes
out <- ode(y = initial_conditions, times = times, func = ode_model, parms = parameters)
head(out,2)
```

```
##      time        S         I          R       S_m       I_m
## [1,]    0  999.000 1.0000000 0.00000000 1000.0000 0.0000000
## [2,]    1 1004.987 0.9350653 0.09599007  998.3323 0.6717126
```

```r
## plot
out = as.data.frame(out)
out$time =NULL
tail(out,2)
```

```
##            S        I        R      S_m      I_m
## 130 191.8607 163.5999 1812.962 18.45507 928.6186
## 131 192.9866 164.5705 1823.915 18.47234 928.8818
```

```
matplot(x = times, y = out, type = "l", main = "beta=0.0001, alpha=0.0007, gamma=0.1,
        nu=0.009, mu=0.003, phi=0.0001, psi=0.0002", xlab = "Time",
        ylab = "Susceptible, Infectious and Removed", lwd = 2, lty = 2,
        bty = "l", col = c("blue", "red", "yellowgreen", "seagreen2", "black"))
legend("topright", c("Human Susceptible", "Human Infectious", "Human Removed",
                  "Mosquito Susceptible", "Mosquito Infectious"), pch = 1,
       col = c("blue", "red", "yellowgreen", "seagreen2", "black"), bty = "n")
abline(h=5, col="yellow2")
abline(h=995, col="yellow2")
```



ii)

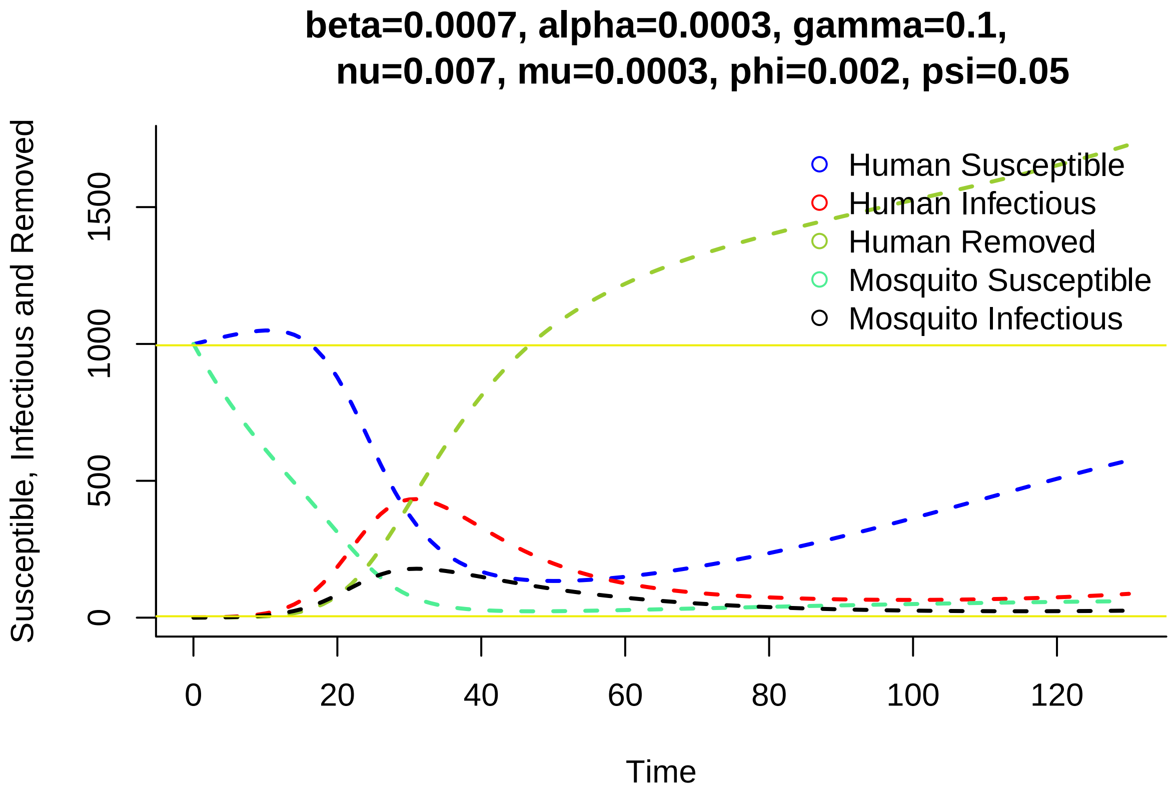
```
## Initialization
initial_conditions <- c(S = 999, I = 1, R = 0, S_m = 1000, I_m = 0)
parameters <- c(beta = 0.0007, alpha = 0.0003, N = 1000, M = 1000, gamma = 0.1,
                nu = 0.007, mu = 0.0003, phi = 0.002, psi = 0.05)
times <- seq(from = 0, to = 130, by = 1)

# result in an equilibrium state with higher prevalence in mosquitoes
out <- ode(y = initial_conditions, times = times, func = ode_model, parms = parameters)
head(out,2)
```

```
##      time        S        I          R       S_m       I_m
## [1,]    0  999.000 1.000000 0.00000000 1000.0000 0.0000000
## [2,]    1 1005.622 1.001609 0.09842385  952.9057 0.2811578
```

```
## plot
out = as.data.frame(out)
out$time =NULL
tail(out,2)
```

```
##            S        I        R      S_m      I_m
## 130 568.2520 85.47541 1719.617 59.88204 25.35857
## 131 574.5517 87.02321 1727.724 60.09314 25.63589
```

```
matplot(x = times, y = out, type = "l", main = "beta=0.0007, alpha=0.0003, gamma=0.1,
        nu=0.007, mu=0.0003, phi=0.002, psi=0.05", xlab = "Time",
        ylab = "Susceptible, Infectious and Removed", lwd = 2, lty = 2,
        bty = "l", col = c("blue", "red", "yellowgreen", "seagreen2", "black"))
legend("topright", c("Human Susceptible", "Human Infectious", "Human Removed",
                  "Mosquito Susceptible", "Mosquito Infectious"), pch = 1,
       col = c("blue", "red", "yellowgreen", "seagreen2", "black"), bty = "n")
abline(h=5, col="yellow2")
abline(h=995, col="yellow2")
```



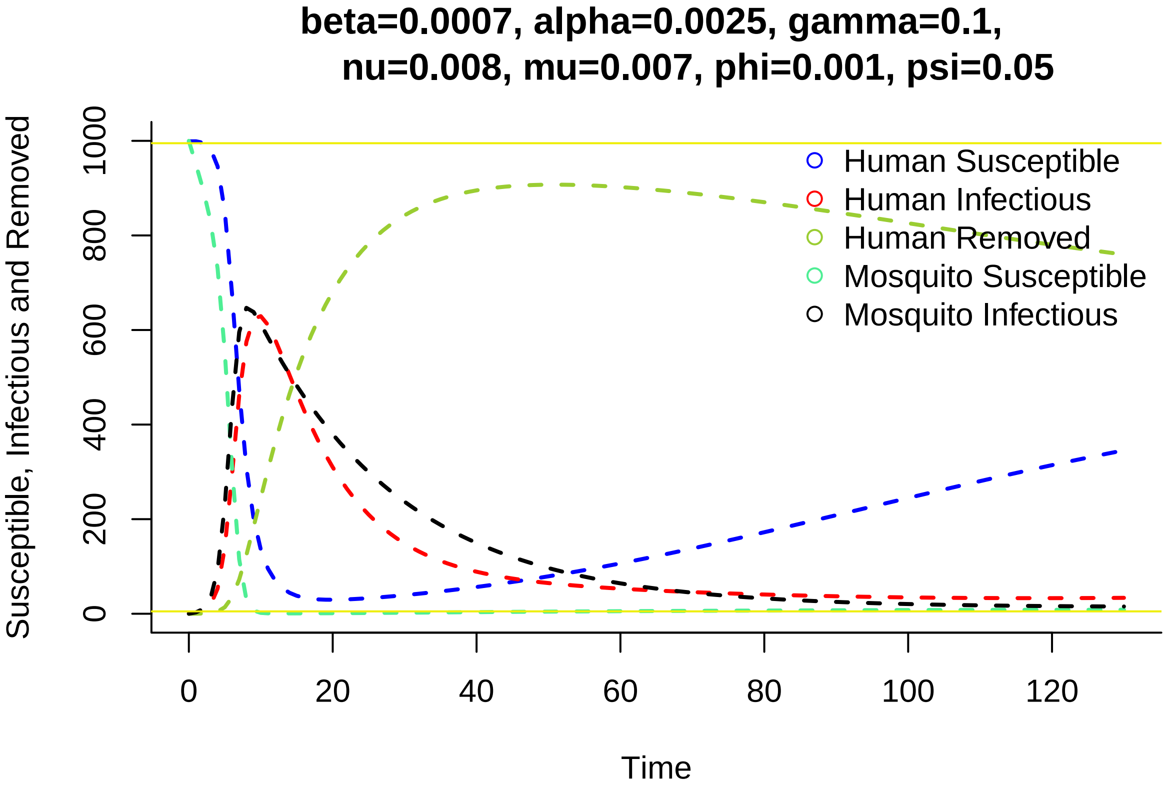
```
## Initialization
initial_conditions <- c(S = 999, I = 1, R = 0, S_m = 1000, I_m = 0)
parameters <- c(beta = 0.0007, alpha = 0.0025, N = 1000, M = 1000, gamma = 0.1,
                nu = 0.008, mu = 0.007, phi = 0.001, psi = 0.05)
times <- seq(from = 0, to = 130, by = 1)

# result in an equilibrium state with higher prevalence in mosquitoes
out <- ode(y = initial_conditions, times = times, func = ode_model, parms = parameters)
head(out,2)
```

```
##      time       S        I         R       S_m      I_m
## [1,]    0 999.000 1.000000 0.0000000 1000.0000 0.000000
## [2,]    1 999.072 1.804718 0.1237824  949.2554 2.949938
```

```
## plot
out = as.data.frame(out)
out$time =NULL
tail(out,2)
```

```
##            S        I        R      S_m      I_m
## 130 342.5560 33.54169 761.5924 8.476326 15.38085
## 131 345.5562 33.64481 759.6274 8.478738 15.32519
```

```
matplot(x = times, y = out, type = "l", main = "beta=0.0007, alpha=0.0025, gamma=0.1,
        nu=0.008, mu=0.007, phi=0.001, psi=0.05", xlab = "Time",
        ylab = "Susceptible, Infectious and Removed", lwd = 2, lty = 2,
        bty = "l", col = c("blue", "red", "yellowgreen", "seagreen2", "black"))
legend("topright", c("Human Susceptible", "Human Infectious", "Human Removed",
                  "Mosquito Susceptible", "Mosquito Infectious"), pch = 1,
       col = c("blue", "red", "yellowgreen", "seagreen2", "black"), bty = "n")
abline(h=5, col="yellow2")
abline(h=995, col="yellow2")
```



We present two plots for each case, which demonstrate the relationship among the number of susceptible, the number of infected and removed individuals. The intersection of the infected and removed curves in the plot represents the equilibrium state, where the rate of new infections is equal to the rate of recoveries. The growth or decline of the infected population is dependent on the relative size of the birth rate, natural death rate, transmission rate, and removal rate. For example, a larger birth rate compared to the natural death rate leads to a larger susceptible population, which could result in a larger outbreak. A higher transmission rate compared to the removal rate results in a larger infected population, whereas a higher removal rate compared to the transmission rate leads to a smaller outbreak.

The relative heights of the red and black lines in the SIR plot are a result of the interplay between the transmission rate between humans and mosquitoes, as well as the birth rate and natural death rate of both human and mosquito populations. If the transmission rate from humans to mosquitoes is greater, the number of infectious mosquitoes increases and the black line rises. Conversely, if the transmission rate from mosquitoes to humans is higher, the number of infectious humans increases and the red line rises. Similarly, a higher birth rate for humans or mosquitoes leads to a corresponding increase in the number of infectious individuals of that species. The removal rate also plays a role in determining the height of the red and black lines. A higher removal rate for humans or mosquitoes results in a quicker decline in the number of infectious individuals of that species. But in this case, the removal rate is fixed at 0.1.

In conclusion, the specific combination of the birth rate, natural death rate, transmission rate, and removal rate determines the trajectory of the epidemic and the relative heights of the red and black lines in the SIR plot.