Supplementary details on experiments

Olivia

2023-09-07

Experimental design

Synthetic reproduction numbers

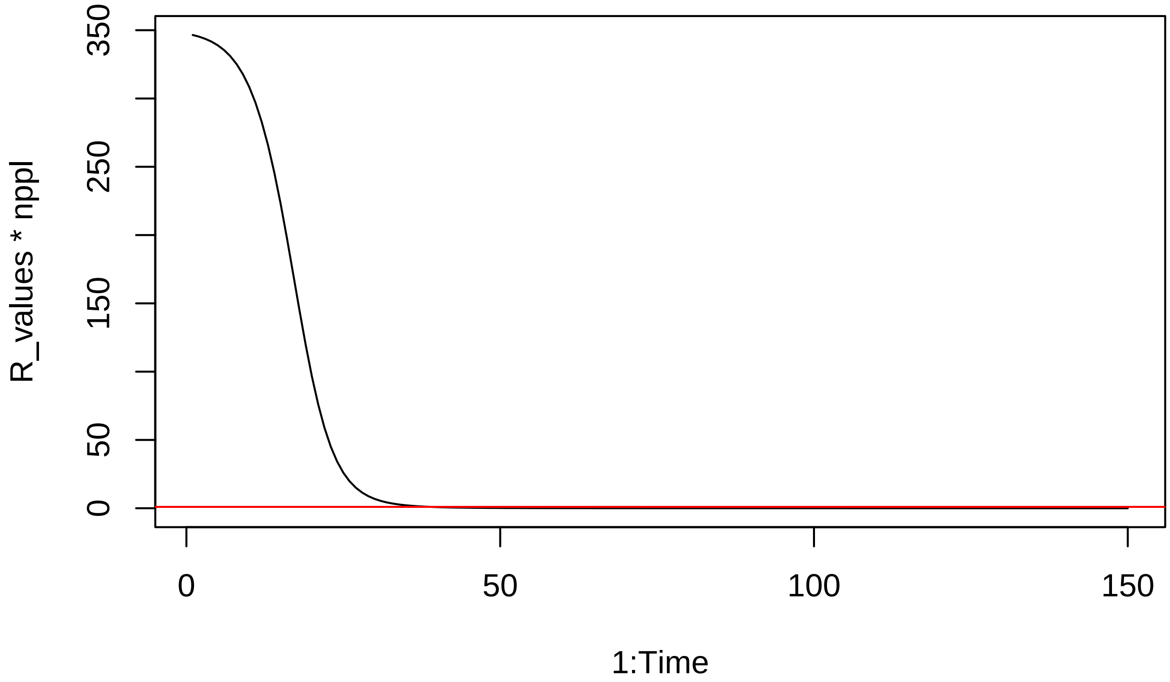
We generate the real-world transmission mechanism using a simple compartmental model, SIR, and compute the reproduction numbers by assuming fixed transmission parameters.

This model assumes a closed and well-mixed population with size N, a constant contact rate β and removal rate γ. The population is divided into three compartments, susceptible, infectious and removed (recovered) compartments, denoted by S, I, R respectively. We start with the initial proportions $S = 0.99, I = 0.01, R = 0.0$ with sum 1 and fix $\beta = 0.35, \gamma = 0.035$. Generate the compartmental proportions for the following $Time = 100$ time points and compute the reproduction number for each time point. The reproduction numbers are plotted in the figure below.

The basic reproduction number is widely used to be $\mathcal{R}_0 = \beta/\gamma$, but the effective reproduction number varies by time. By definition, the effective reproduction number is the number of secondary infection caused by a primary infection at a specific time. At time t, $S_{t-1} - S_t$ individuals are infected and moved from the susceptible compartment to the infectious compartment. There are I_{t-1} primary individuals at time $t-1$. Thus, the reproduction number at time t is $\mathcal{R}_t = (S_{t-1} - S_t)/I_{t-1}$.

Figure for one single set of parameters.

```r
# Parameters
beta = 0.35  # Transmission rate
gamma = 0.035 # Recovery rate

S0 = 0.99  # Initial proportion of susceptible individuals
I0 = 0.01  # Initial proportion of infectious individuals
R0 = 0.0  # Initial proportion of recovered individuals
Time = 150  # Total number of time points

# Store S, I, and R values
S = numeric(Time + 1); S[1] = S0
I = numeric(Time + 1); I[1] = I0
R = numeric(Time + 1); R[1] = R0
R_values = numeric(Time)

# Simulate the SIR model
for(t in 1:Time){
    dS = -beta * S[t] * I[t]
    dI = (beta * S[t] * I[t]) - (gamma * I[t])
    dR = gamma * I[t]

    S[t+1] = S[t] + dS
    I[t+1] = I[t] + dI
    R[t+1] = R[t] + dR

    # Calculate R for each time point
    R_values[t] = -dS / I[t]
}

# Set a number of population
nppl = 1000
# Plot R over time
plot(R_values*nppl, x=1:Time, type="l")
abline(h = 1, col="red")
```



```
plot((1-S)*nppl, col="red", type="l") # cumulative infection proportions
lines(R*nppl, col="orange")
lines(I*nppl, col="blue")
```

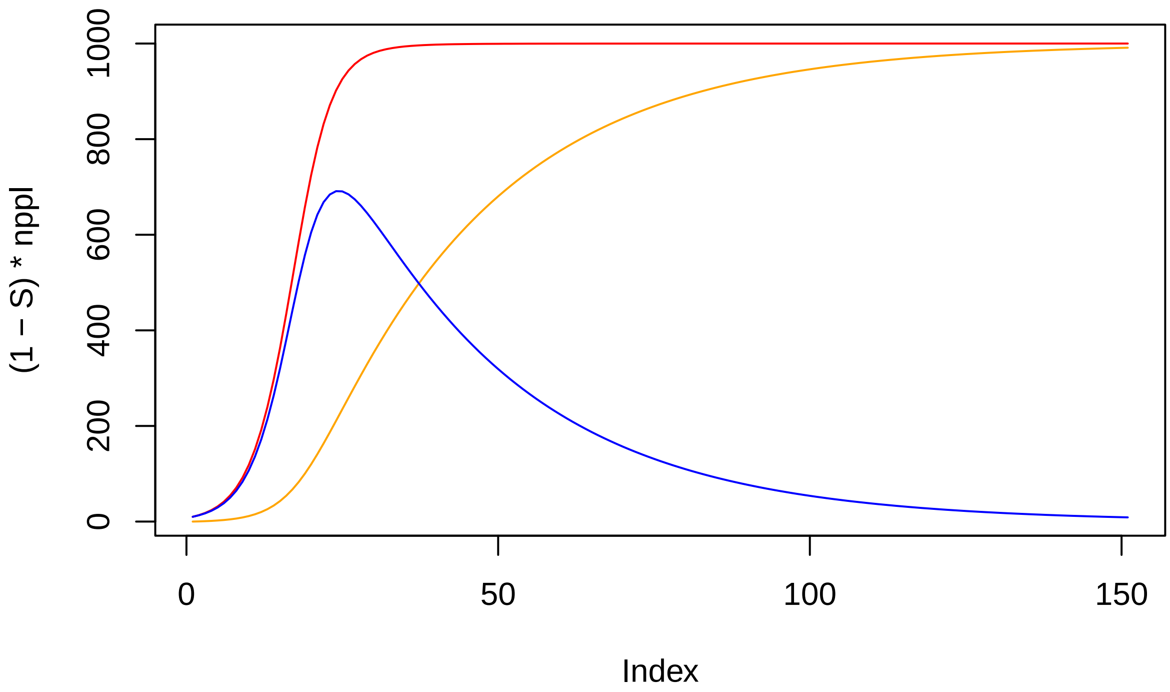


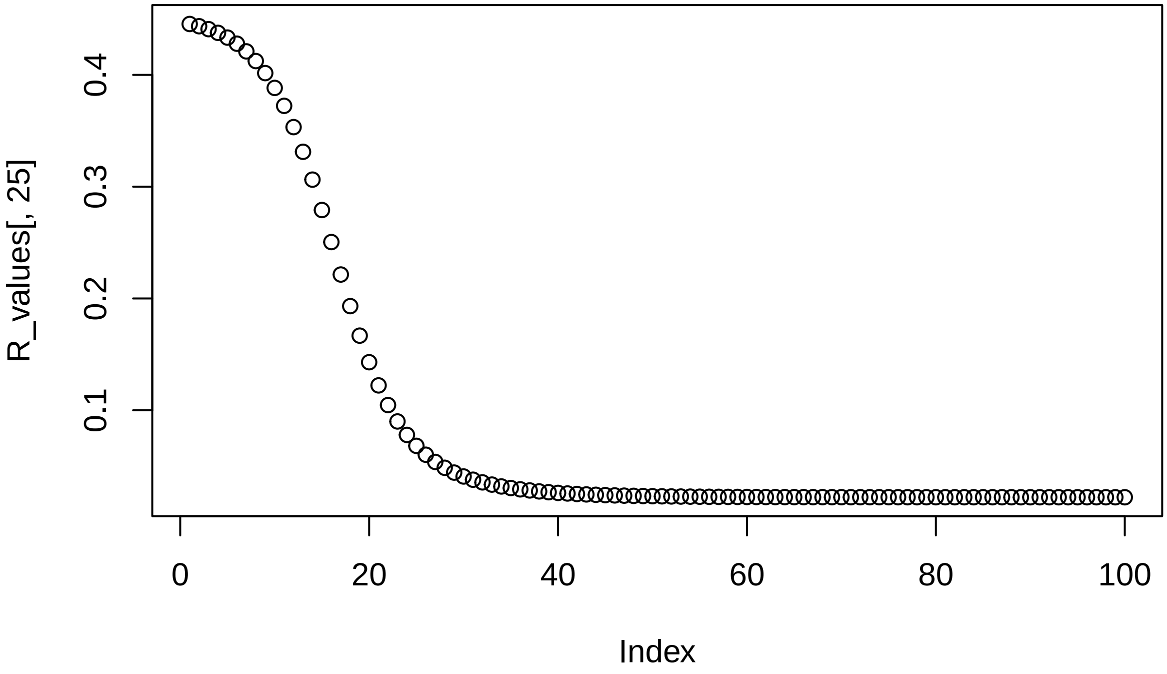
Figure for multiple sets of parameters.

```
# Parameters
beta = seq(0.35, 0.95, 0.05)  # Transmission rate
gamma = seq(0.05, 0.55, 0.05)  # Recovery rate
para_table <- data.table::CJ(beta = beta, gamma = gamma)
npar <- nrow(para_table)
print(para_table)
```

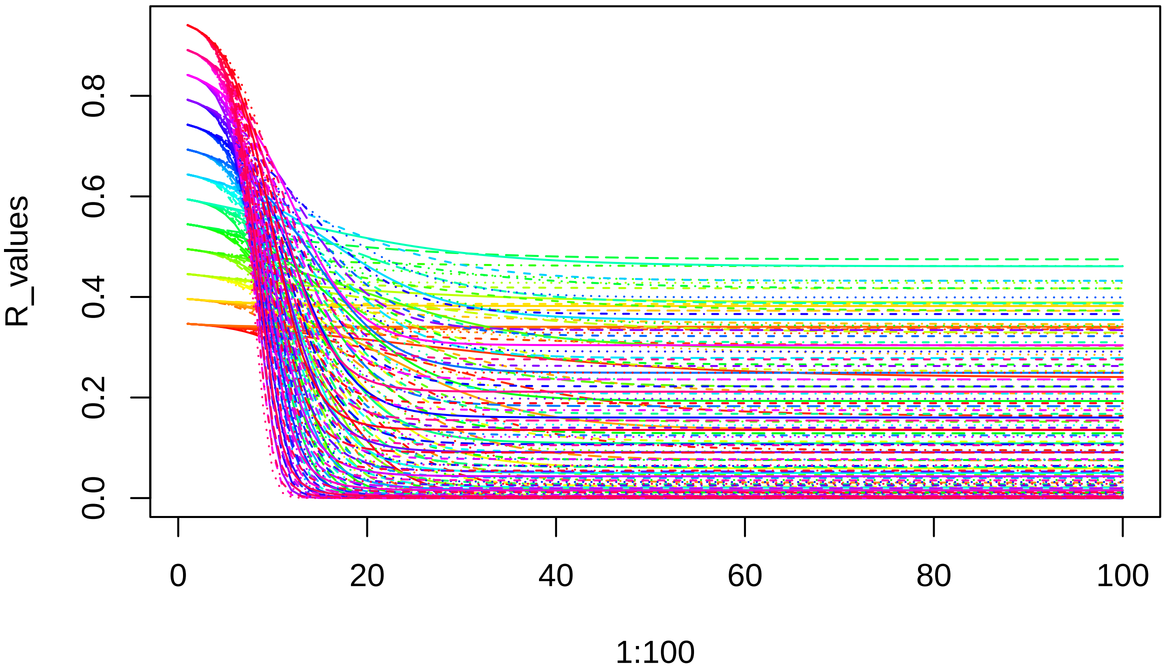
```
##      beta gamma
##   1: 0.35  0.05
##   2: 0.35  0.10
##   3: 0.35  0.15
##   4: 0.35  0.20
##   5: 0.35  0.25
##  ---
## 139: 0.95  0.35
## 140: 0.95  0.40
## 141: 0.95  0.45
## 142: 0.95  0.50
## 143: 0.95  0.55
```

```
S0 = 0.99  # Initial proportion of susceptible individuals
I0 = 0.01  # Initial proportion of infectious individuals
R0 = 0.0  # Initial proportion of recovered individuals
Time = 100  # Total number of time points

# Initialize arrays to store S, I, and R values
S = numeric(Time + 1); S[1] = S0
I = numeric(Time + 1); I[1] = I0
R = numeric(Time + 1); R[1] = R0
R_values = matrix(nrow = Time, ncol = npar)
```

```r
# Simulate the SIR model
for(l in 1:npar){
  for(t in 1:Time){
    dS = -para_table$beta[l] * S[t] * I[t]
    dI = (para_table$beta[l] * S[t] * I[t]) - (para_table$gamma[l] * I[t])
    dR = para_table$gamma[l] * I[t]

    S[t+1] = S[t] + dS
    I[t+1] = I[t] + dI
    R[t+1] = R[t] + dR

    # Calculate R for each time point
    R_values[t,l] = -dS / I[t]
  }
}
colnames(R_values) <- paste("Para",1:npar,sep="")

plot(R_values[,25])
```



```r
library(RColorBrewer)
palette <- rainbow(ncol(R_values))
matplot(1:100, R_values, type="l", col = palette)
```



```
#legend("topright", legend = colnames(R_values), fill = palette, title = "Legend", cex=.3)
```