Lab 7: More Practice with Functions in R and Returning Lists

STAT 244NF: Infectious Disease Modeling

Partial SOLUTIONS

02 November, 2021

In the last lab, we were introduced to writing functions in R, and we wrote a function to simulate data from the SIS model, which we called `SIS_simulation`. Today, we are going to write several other functions, building on what we have learned so far. One key difference is that we are going to work on returning a list (which could include a data frame, reproductive number, etc.) This will be useful if you want to return multiple pieces of information from your function.

Today, we will be working with the SIR and the SEIR model.

SIR model

The difference equations are given below:

$$S_t = S_{t-1} - \lambda_t S_{t-1}$$
$$I_t = I_{t-1} + \lambda_t S_{t-1} - \rho I_{t-1}$$
$$R_t = R_{t-1} + \rho I_{t-1}$$

Exercise 1 (Review)

Write a function to simulate from an SIR model. Call it `SIR_simulation`.

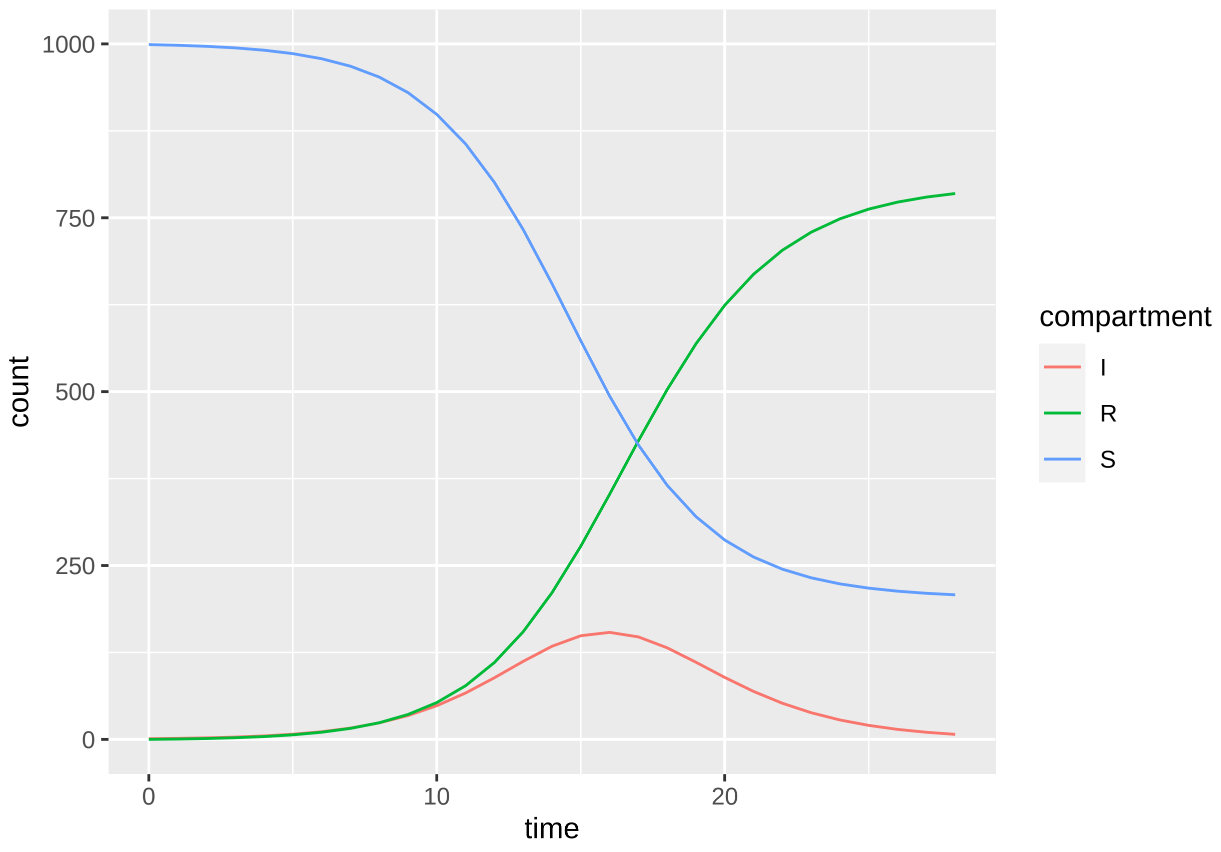
- It should take in the following variables: N (population size), I0 (initial number of infected individuals), R0 (initial number of recovered individuals), R_0 (the basic reproductive number), D (infectious duration), and time.
- It should return a data frame with three columns: time, compartment, and count.

```r
## function to calculate lambda_t
lambda_t_fcn <- function(R_0, D, I_i, N){
  c_e <- R_0/D
  return(1-exp(-c_e*I_i/N))
}
```

```r
SIR_simulation <- function(N, I0, R0, R_0, D, Time){
  SIR_df <- data.frame(time=0:Time,
                       S=rep(NA, Time+1),
                       I=rep(NA, Time+1),
                       R=rep(NA, Time+1),
                       lambda_t=rep(NA, Time+1))
  SIR_df$S[1] <- N-I0-R0
  SIR_df$I[1] <- I0
  SIR_df$R[1] <- R0

  for (i in 2:(Time+1)){
    SIR_df$lambda_t[i] <- lambda_t_fcn(R_0=R_0, D=D, I_i=SIR_df$I[i-1], N=N)
    SIR_df$S[i] <- SIR_df$S[i-1]-SIR_df$lambda_t[i]*SIR_df$S[i-1]
    SIR_df$I[i] <- SIR_df$I[i-1]+SIR_df$lambda_t[i]*SIR_df$S[i-1]-1/D*SIR_df$I[i-1]
    SIR_df$R[i] <- SIR_df$R[i-1]+1/D*SIR_df$I[i-1]
  }

  return(data.frame(time=rep(0:Time, 3),
                    compartment=rep(c("S","I", "R"), each=(Time+1)),
                    count=c(SIR_df$S, SIR_df$I, SIR_df$R)))
}
```

Test your function for the following values: N=1000, I0=1, R0=0, R_0=2, D=2 days, time=28 days. Make sure that your function returns a data frame that seems reasonable. It is easiest to do this by plotting your results.

```
SIR_sim1 <- SIR_simulation(N=1000, I0=1, R0=0, R_0=2, D=2, Time=28)
ggplot(data=SIR_sim1, aes(x=time, y=count)) +
  geom_line(aes(color=compartment))
```



Exercise 2

It can be useful to return more than just a data frame from a function. For instance, we may want to return a data frame that contains information on our SIR model, the time point when we have the largest number of infectious individuals, and the herd immunity threshold (which is $1 - \frac{1}{R_0}$). The reason we want to do this is that we want to explore the relationship between the time point when we have the largest number of infectious individuals and the herd immunity threshold in the context of our simulation. We can achieve this by returning a list. In a moment we are going to modify our function from Exercise 1 to make it return a list (described below). Before we do that, it is useful to play around with lists a little bit. Run the following code to see how lists work:

```
## make a list with two arguments, x and y:
list(x=5, y=6)
```

```
## $x
## [1] 5
##
## $y
## [1] 6
```

```
## save our list so we can play with it
list1 <- list(x=5, y=6)
list1
```

```
## $x
## [1] 5
##
## $y
## [1] 6
```

```
##access the element named x in the list
list1$x
```

```
## [1] 5
```

```
##access the element named y in the list
list1$y
```

```
## [1] 6
```

```
test_df <- data.frame(x=0:3, y=1:4)
list2 <- list(df=test_df, time=5)
list2
```

```
## $df
##   x y
## 1 0 1
## 2 1 2
## 3 2 3
## 4 3 4
##
## $time
## [1] 5
```

```r
## access the element named df in the list
list2$df
```

```
##   x y
## 1 0 1
## 2 1 2
## 3 2 3
## 4 3 4
```

```r
## access the element named time in the list
list2$time
```

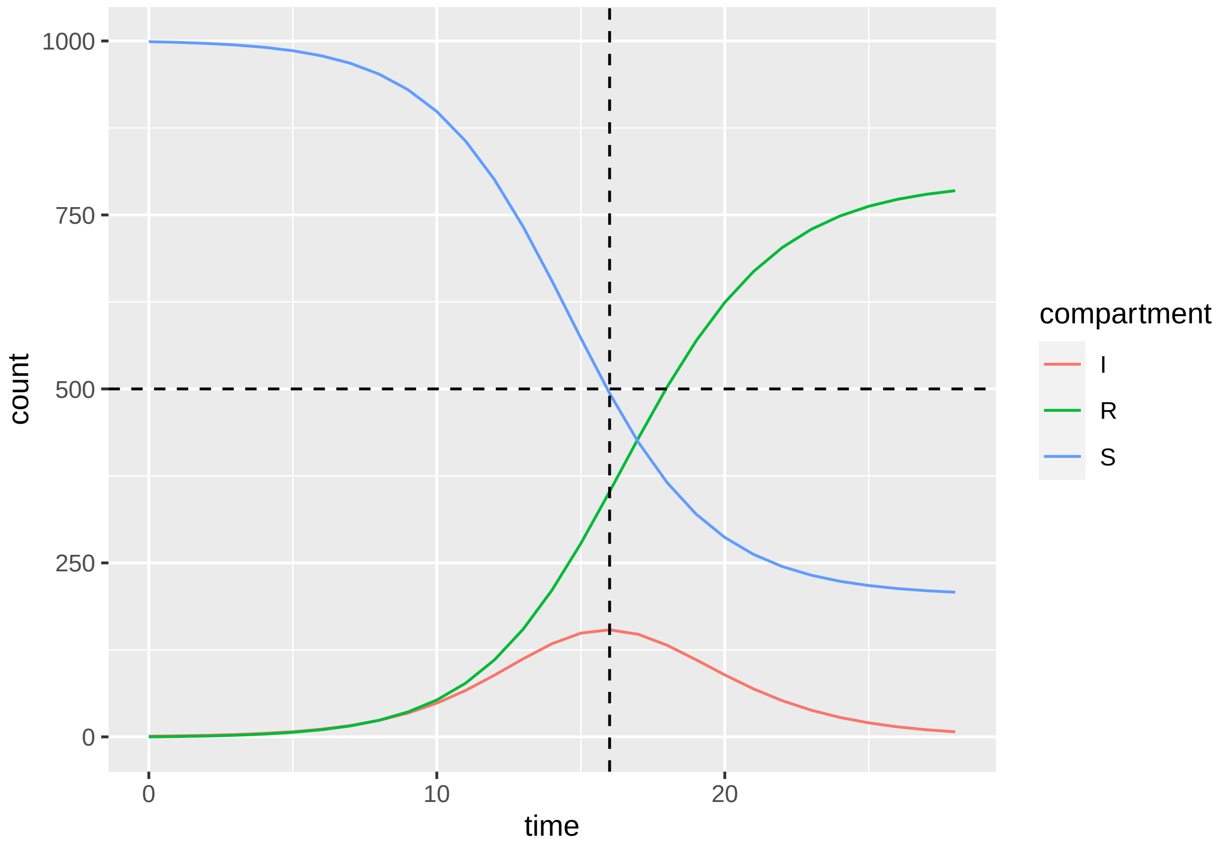
```
## [1] 5
```

Now, modify your SIR_simulation function so that it (a) finds the time point when we have the largest number of infectious individuals, and calculates the herd immunity threshold and (b) returns a list that contains your data frame (what you already returned), the time point when we have the largest number of infectious individuals, and the herd immunity threshold. In your list, you can name these pieces comp_df, max_time, and HIT, respectively.

```r
SIR_simulation <- function(N, I0, R0, R_0, D, Time){
  SIR_df <- data.frame(time=0:Time,
                       S=rep(NA, Time+1),
                       I=rep(NA, Time+1),
                       R=rep(NA, Time+1),
                       lambda_t=rep(NA, Time+1))

  SIR_df$S[1] <- N-I0-R0
  SIR_df$I[1] <- I0
  SIR_df$R[1] <- R0

  for (i in 2:(Time+1)){
    SIR_df$lambda_t[i] <- lambda_t_fcn(R_0=R_0, D=D, I_i=SIR_df$I[i-1], N=N)
    SIR_df$S[i] <- SIR_df$S[i-1]-SIR_df$lambda_t[i]*SIR_df$S[i-1]
    SIR_df$I[i] <- SIR_df$I[i-1]+SIR_df$lambda_t[i]*SIR_df$S[i-1]-1/D*SIR_df$I[i-1]
    SIR_df$R[i] <- SIR_df$R[i-1]+1/D*SIR_df$I[i-1]
  }

  ## Calculate time point at which we have maximum number of infectious individuals
  max_time <- SIR_df$time[which.max(SIR_df$I)]

  ## Calculate herd immunity threshold
  HIT <- 1-1/R_0

  ## Save data frame with appropriate format to plot in ggplot2
  df <- data.frame(time=rep(0:Time, 3),
                   compartment=rep(c("S","I", "R"), each=(Time+1)),
                   count=c(SIR_df$S, SIR_df$I, SIR_df$R))

  return(list(df=df, max_time=max_time, HIT=HIT))
}
```

You can add horizontal and vertical lines to a plot by adding the layers `geom_hline` and `geom_vline`, respectively. Add a horizontal line at the herd immunity threshold and a vertical line at the max_time. What do you notice?

```r
SIR_sim2 <- SIR_simulation(N=1000, I0=1, R0=0, R_0=2, D=2, Time=28)
ggplot(data=SIR_sim2$df, aes(x=time, y=count)) +
  geom_line(aes(color=compartment)) +
  geom_vline(xintercept=SIR_sim2$max_time, linetype="dashed") +
  geom_hline(yintercept=SIR_sim2$HIT*1000, linetype="dashed")
```



SEIR model

The difference equations are given below:

$$
\begin{aligned}
S_t &= S_{t-1} - \lambda_t S_{t-1} \\
E_t &= E_{t-1} + \lambda_t S_{t-1} - \pi E_{t-1} \\
I_t &= I_{t-1} + \pi E_{t-1} - \rho I_{t-1} \\
R_t &= R_{t-1} + \rho I_{t-1}
\end{aligned}
$$

E_t is the number of people in the pre-infectious compartment at time t; π is the proportion of individuals that become infectious by time t, so πE_{t-1} people leave the pre-infectious compartment and enter the infectious compartment. The parameter π is related to the latent (pre-infectious) period: $\pi = \frac{1}{D'}$.

The formulation of λ_t is the same as in the SIR model – it still depends on the proportion of infectious individuals in the population.

Exercise 3 (Review)

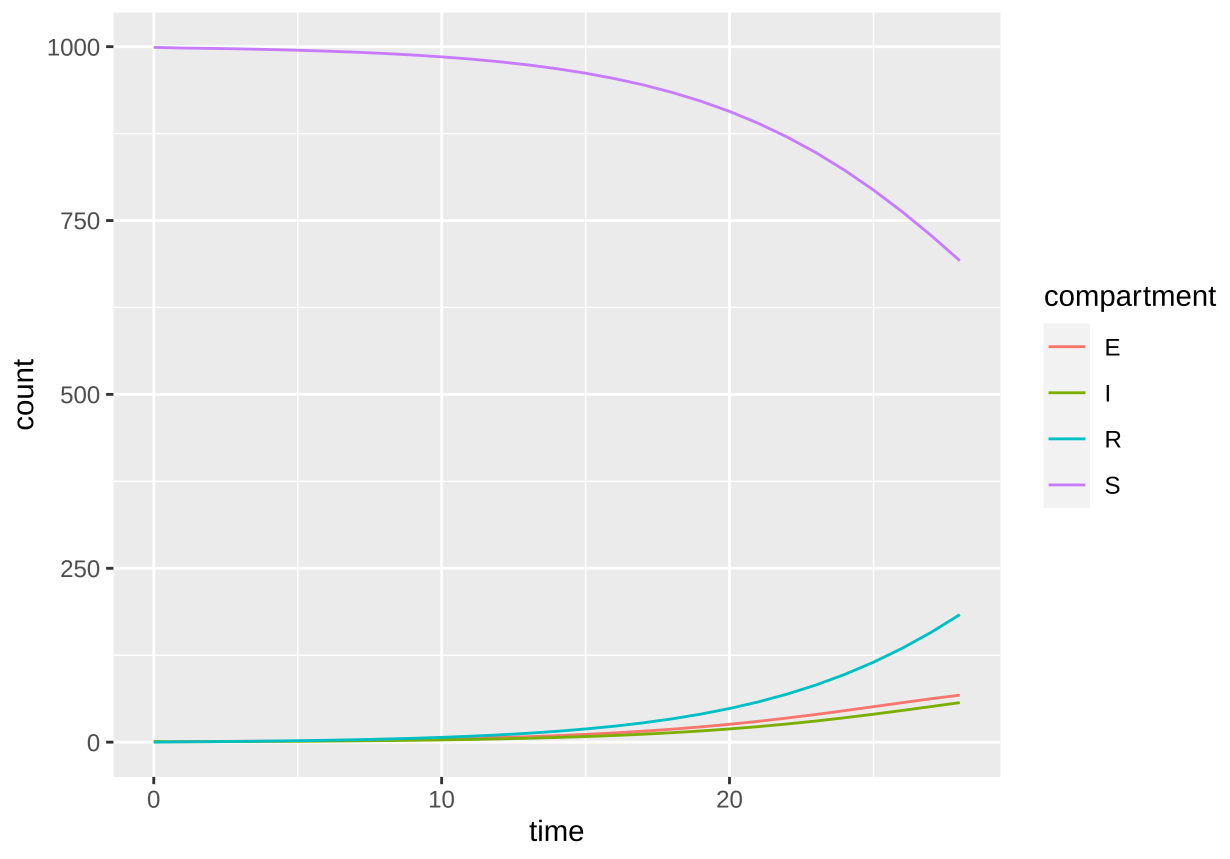
Write a function to simulate from an SEIR model. Call it `SEIR_simulation`.

- It should take in the following variables: N (population size), E0 (initial number of exposed individuals), I0 (initial number of infected individuals), R0 (initial number of recovered individuals), R_0 (the basic reproductive number), pD (pre-infectious duration), D (infectious duration), and time.
- It should return a data frame with three columns: time, compartment, and count.

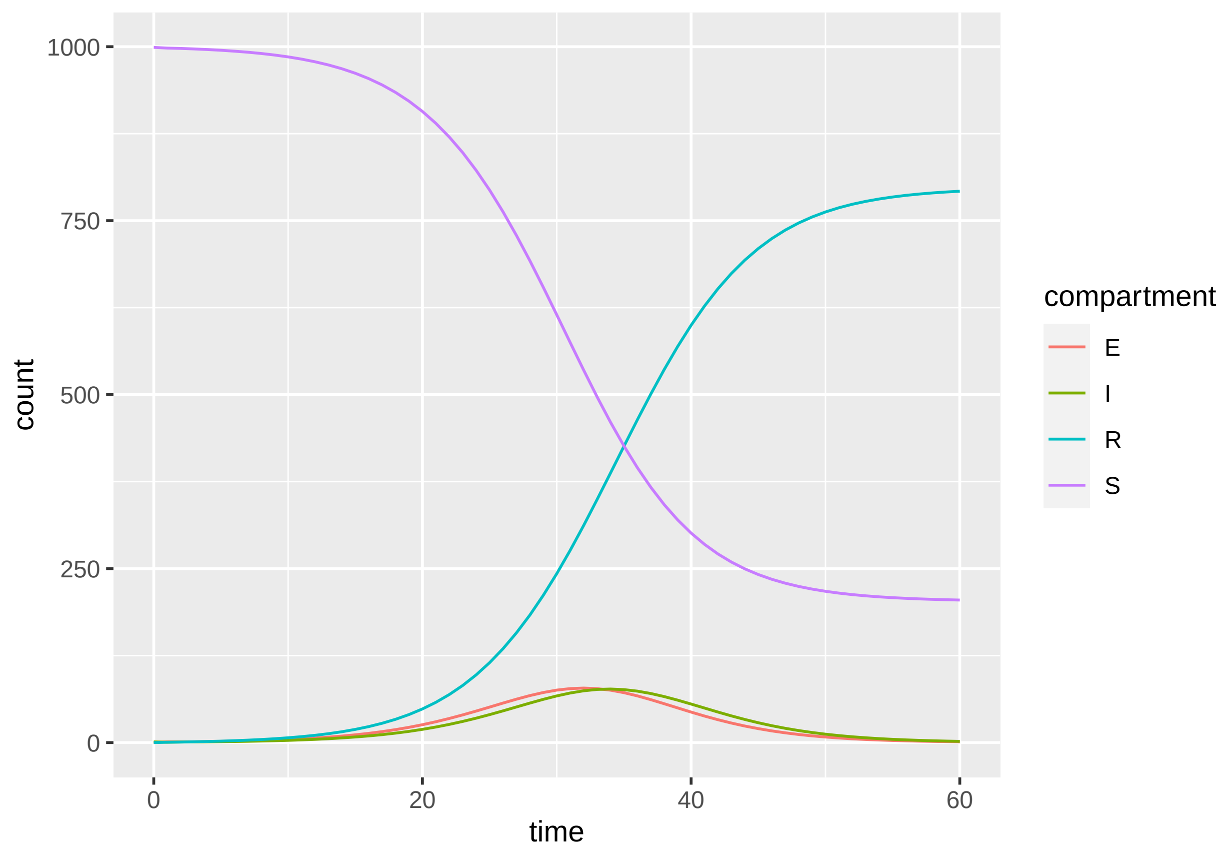
```r
SEIR_simulation <- function(N, E0, I0, R0, R_0, pD, D, Time){
  SEIR_df <- data.frame(time=0:Time,
                        S=rep(NA, Time+1),
                        I=rep(NA, Time+1),
                        R=rep(NA, Time+1),
                        lambda_t=rep(NA, Time+1))

  SEIR_df$S[1] <- N-E0-I0-R0
  SEIR_df$E[1] <- E0
  SEIR_df$I[1] <- I0
  SEIR_df$R[1] <- R0

  for (i in 2:(Time+1)){
    SEIR_df$lambda_t[i] <- lambda_t_fcn(R_0=R_0, D=D, I_i=SEIR_df$I[i-1], N=N)
    SEIR_df$S[i] <- SEIR_df$S[i-1]-SEIR_df$lambda_t[i]*SEIR_df$S[i-1]
    SEIR_df$E[i] <- SEIR_df$E[i-1]+SEIR_df$lambda_t[i]*SEIR_df$S[i-1]-1/pD*SEIR_df$E[i-1]
    SEIR_df$I[i] <- SEIR_df$I[i-1]+1/pD*SEIR_df$E[i-1]-1/D*SEIR_df$I[i-1]
    SEIR_df$R[i] <- SEIR_df$R[i-1]+1/D*SEIR_df$I[i-1]
  }

  return(data.frame(time=rep(0:Time, 4),
                    compartment=rep(c("S", "E", "I", "R"), each=(Time+1)),
                    count=c(SEIR_df$S, SEIR_df$E, SEIR_df$I, SEIR_df$R)))
}
```

Test your function for the following values: N=1000, E0=0, I0=1, R0=0, R_0=2, pD=2 days, D=2 days, time=28 days. Make sure that your function returns a data frame that seems reasonable. It is easiest to do this by plotting your results.

```r
SEIR_sim <- SEIR_simulation(N=1000, E0=0, I0=1, R0=0, R_0=2, pD=2, D=2, Time=28)
ggplot(data=SEIR_sim, aes(x=time, y=count)) +
  geom_line(aes(color=compartment))
```



```r
## This makes more sense for more days - try 60
SEIR_sim2 <- SEIR_simulation(N=1000, E0=0, I0=1, R0=0, R_0=2, pD=2, D=2, Time=60)
ggplot(data=SEIR_sim2, aes(x=time, y=count)) +
  geom_line(aes(color=compartment))
```



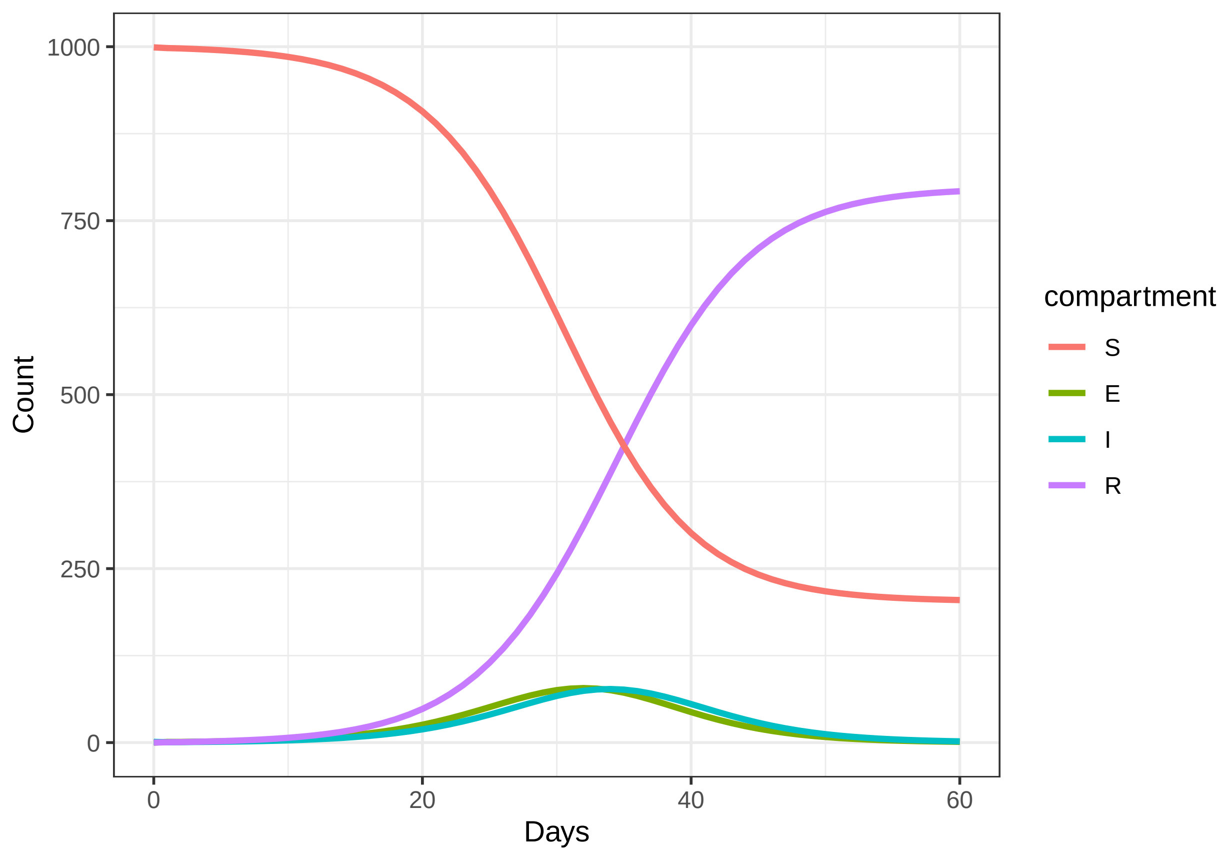
If we compare these results those from the SIR model, the only difference here is that we have added a pre-infectious period of 2 days. How does adding the E compartment influence the I curve?

We still get the same total number of infectious people, but it takes about twice as long because of the pre-infectious period (2 days here). The maximum number of infectious individuals in smaller.

Exercise 4 (time-permitting)

R (more specifically ggplot2) chooses a default ordering and placement for the plot legend(s). However, you can control pieces of this. Read this resource from Cookbook for R and modify your legend. You can start with changing the order of the compartments so they show up in the same ordering as they do in the model (rather than in alphabetical order). If you have more time, focus on making your plot as presentation ready as possible (readable, well organized, etc.)

```
## Example
ggplot(data=SEIR_sim2, aes(x=time, y=count)) +
  geom_line(aes(color=compartment), size=1.1) +
  theme_bw() +
  scale_colour_discrete(limits = c("S", "E", "I", "R")) +
  xlab("Days") +
  ylab("Count")
```



1000
750
500
250
0
Count
0
20
40
60
Days
compartment
S
E
I
R