# Problems 1.1 - An Introduction to Compartment Models

Amy Greer, University of Guelph for Math 6115,Bonne Bay, August 2023

2023-06-07

**Note:** Some of the examples and code provided in this laboratory have been extracted and/or modified from the textbook Epidemics, Models and Data using R by Ottar Bjornstad (https://link.springer.com/book/10.1007/978-3-319-97487-3).

## PART ONE - Setting Up and Interpreting Compartmental Models

The first part of this laboratory exercise will ask you to interpret and describe compartmental models that use differential equations. Having a solid understanding of model structure is an important first step towards moving these models into code that you can use for analysis.

The following diagram shows the general structure of a model for the transmission dynamics of a highly transmissible pathogen in a human population. This is a basic SEIR model. The letters next to each arrow represent the rate at which individuals move from one compartment to the next.

SEIR.png

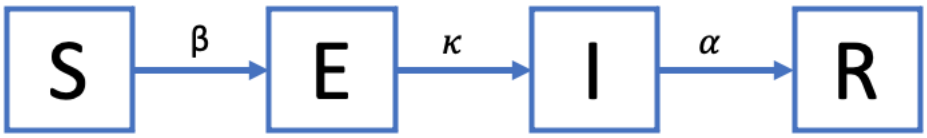



Figure 1: Simple SEIR Compartment model

## Question 1 (10 points)

**1a) Write down the differential equations for this model (6 points)**

**1b) What are the simplifying assumptions of this model (list at least 4)? (4 points)**

SEIR table.png

| Parameter | Description | Value (units) |
|---|---|---|
| $\beta$ | Transmission rate | 0.5 |
| $\kappa$ | Latent period | 1/3 days = 0.333 |
| $\alpha$ | Infectious period for symptomatic infections (I) | 1/7 days = 0.14 |

| Initial conditions | Value |
|---|---|
| $S(0)$ | 9950 |
| $E(0)$ | 0 |
| $I(0)$ | 50 |
| $R(0)$ | 0 |
| $N = S(0) + E(0) + I(0) + R(0) = 10{,}000$ | |

Figure 2: Parameter values and initial conditions for the SEIR model.

# PART TWO - An Example of a Simple Deterministic SEIR model

Let's start by generating the necessary code for a simple deterministic SEIR model based on the information provided in Part 1.

**Step 1:** We will first set up the model and then plot the model output. Step one is that we need to define the function (often called the gradient-functions) for the system of equations. We will use the deSolve-package to solve the equations after describing the different components.

## Question 2 (4 points)

### 2) Translate your equations into R code in lines 64-68 (4 points)

```
require(deSolve)

 seirmod=function(t, y, parms){
   #Pull state variables from y vector
   S=y[1]
   E=y[2]
   I=y[3]
   R=y[4]
   #Pull the required parameter values from the parms vector
   beta=parms["beta"]
   kappa=parms["kappa"]
   alpha=parms["alpha"]
   N=parms["N"]
   #Define the equations (use what you wrote out in question 1) - 6 points
   dS = -(beta*S*I/N)
   dE = +(beta*S*I/N) - (kappa*E)
   dI = (kappa*E) - (alpha*I)
   dR = (alpha*I)
   res=c(dS, dE, dI, dR)
```

```
  #Return list of gradients
  list(res)
}
```

**Steps 2-4:** we need to assign the simulation time (how long the simulation will run for and also what the time step is), the parameter values, and the initial conditions. Essentially we are specifying the timepoints at which we want the ode to record the states of the system (here we use 100 days with 1 time increment per day as specified in the vector times), the parameter values (in this case as specified in the vector parms) and the initial conditions as specified in the start vector.

```
times  = seq(0, 100, by=1)
parms  = c(beta=0.5, kappa=1/3, alpha=1/7, N=10000)
start = c(S=9950, E=0, I=50, R = 0)
```

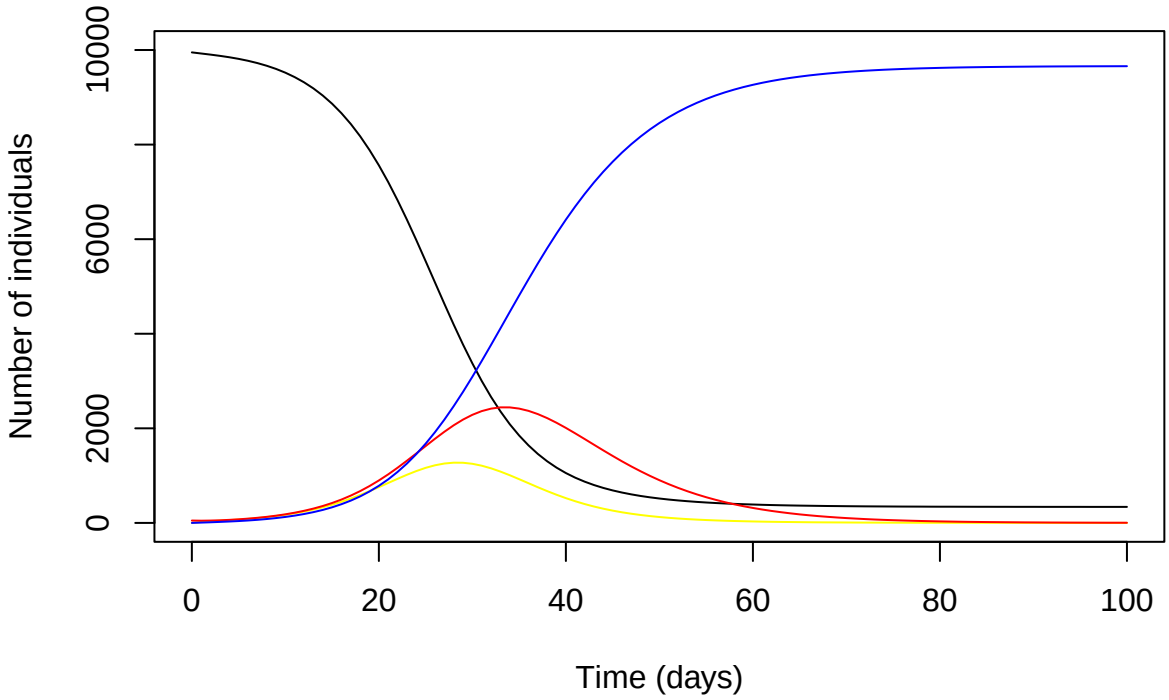
**Step 5:** Now we need to feed the start values, times, the function and parameter vector to the ode function. For convenience, we convert the output to a dataframe (ode returns a list). The head-function shows the first 5 rows of out, and round (,2) rounds the number to three decimals.

```
out = ode(y = start, times = times, func = seirmod,
     parms = parms)
out=as.data.frame(out)
head(round(out, 3))
```

```
##   time        S      E      I      R
## 1    0 9950.000  0.000 50.000  0.000
## 2    1 9926.254 20.155 46.764  6.827
## 3    2 9902.657 34.537 49.179 13.627
## 4    3 9876.950 46.678 55.318 21.054
## 5    4 9847.560 58.543 64.329 29.568
## 6    5 9813.193 71.306 75.945 39.557
```

We can plot the result to see if the model predicts an initial exponential growth of the epidemic that decelerates as susceptibles are depleted, and finally epidemic fade-out as susceptible numbers are too low to sustain the chain of transmission.

```
plot(x = out$time, y = out$S, col = "black", ylab = "Number of individuals",
      xlab = "Time (days)", type = "l", xlim = c(0, 100), ylim = c(0,10000))
 lines(x = out$time, y = out$E, col = "yellow")
 lines(x = out$time, y = out$I, col = "red")
 lines(x = out$time, y = out$R, col = "blue")
```




```
#note that this figure is generated using simple plotting and not ggplot.
#ggplot is a great R tool for plotting that you might want to look into.
```

**NOTE:** You can also start to become familiar with the ggplot package which gives much more flexibility to ensure that our model plots look the way we would like them to look. You can find a great ggplot "cheatsheet" for customizing your plots here: (https://statsandr.com/blog/files/ggplot2-cheatsheet.pdf)

The next thing we might want to investigate for a model like this is to calculate the basic reproductive number for the disease that we are examining. We have already discussed that $R_0 = \beta/(\alpha)$ and we have already defined the values for these parameters so we can calculate $R_0$ directly and then ask for it to be displayed.

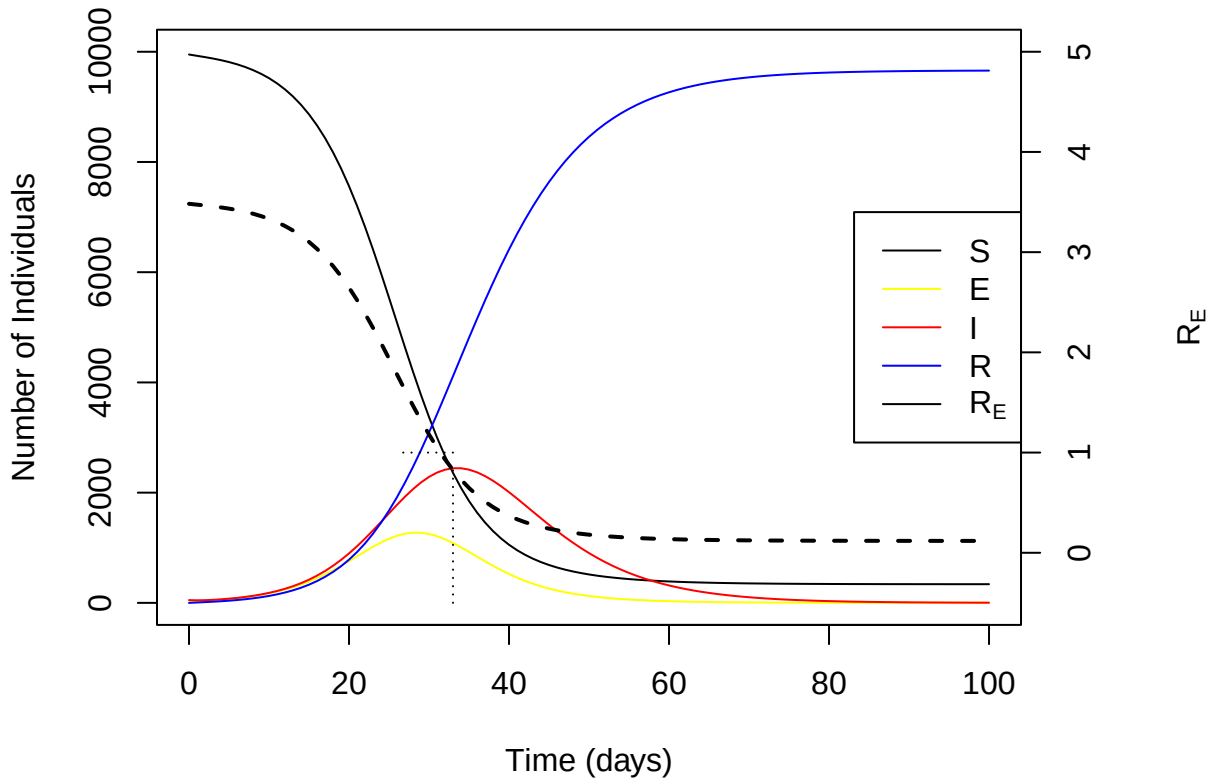
```
#Calculate R0
R0 = parms["beta"]/(parms["alpha"])
R0
```

```
## beta
##  3.5
```

So we now know that the basic reproductive number ($R_0$) for this outbreak is ~3.5 (which is pretty high). We are going to add a right hand axis to our previous figure for the *effective* reproductive number ($R_e$) - the expected number of new cases per infected individual in a *not* completely susceptible population - and a legend so that we can confirm that the turnover of the epidemic happens exactly when $R_e = R_0 s$=1, where $s$ is the fraction of remaining susceptible hosts. The threshold $R_0 s = 1$ is the same as $s = 1/R\ 0$ results in an important rule of thumb for interventions like vaccine induced eradication and herd immunity: If we can keep the susceptible population below a critical fraction, $p_c = 1 - 1/R\ 0$ , then pathogen spread will no longer be possible and the pathogen will not be able to reinvade the population. In order to update our figure accordingly, we add the following additional code.

```
#Adjust margins to accommodate a second right axis
par(mar = c(5,5,2,5))
#Plot state variables
plot(x = out$time, y = out$S, ylab = "Number of Individuals",
     xlab = "Time (days)",  type = "l", xlim=c(0,100), ylim=c(0,10000))
lines(x = out$time, y = out$E, col = "yellow")
```

```
lines(x = out$time, y = out$I, col = "red")
lines(x = out$time, y = out$R, col = "blue")

#Add vertical line at turnover point
xx = out$time[which.max(out$I)]
lines(c(xx,xx), c(1/R0,max(out$I)), lty = 3)

#prepare to superimpose 2nd plot
par(new = TRUE)
#plot effective reproductive ratio (w/o axes)
plot(x = out$time, y = R0*out$S/10000, type = "l", lty = 2,
     lwd = 2, col = "black", axes = FALSE, xlab = NA,
     ylab = NA, ylim = c(-.5, 5.0))
lines(c(xx, 26), c(1,1), lty = 3)
#Add right-hand axis for RE
axis(side = 4)
mtext(side = 4, line = 4, expression(R[E]))
#Add legend
legend("right", legend = c("S", "E", "I", "R",expression(R[E])), lty = c(1,1,1,1,1,2),
     col = c("black", "yellow", "red", "blue", "black"))
```




Something else that we often want to know is about the *final epidemic size*. $S$ is the fraction of susceptibles that escape the infection all together, and $R$ is the *final epidemic size* otherwise called the fraction of susceptibles that will be infected before the epidemic self-extinguishes. For a "closed" epidemic (one where there are no demographics included) there is an exact mathematical solution to the final epidemic size (there are other ways to do this that we will not go into in this introductory class). For our purposes here, the rootSolve-package will attempt to find the equilibria of the system through numerical integration. The function runsteady is used to integrate until the system settles to some steady state (if one exists).

```
require(rootSolve)
```

```
## Loading required package: rootSolve
```

```
equil=runsteady(y=c(S=9950, E=0, I=50, R = 0),
times=c(0,100), func=seirmod, parms=parms)
round(equil$y, 3)
```

```
##        S        E        I        R
##  338.644    0.227    2.898 9658.231
```

So for the parameters that we have used for this model, ~3.4% of susceptibles (~339/10,000) are expected to escape infection altogether and ~96.6% (~9661/10,000) - the final epidemic size - are expected to be infected during the course of the epidemic.

## PART TWO - Adapt the Model to SEIAR and Interpret the Outcomes Compared to the SEIR model

One of the assumptions of this SEIR model as described, is that all infected individuals are symptomatic. For some pathogens, infected individuals can be either symptomatic or asymptomatic (e.g. influenza). Lets assume that our current SEIR model is an oversimplification of the biology for this pathogen. Let's assume that ~ 1/3 of infected individuals experience an asymptomatic infection and that these individuals are equally infectious as symptomatic infections but that asymptomatic individuals recover more quickly (in 5 days instead of 7 days) (Figure 3).

SEIAR.png

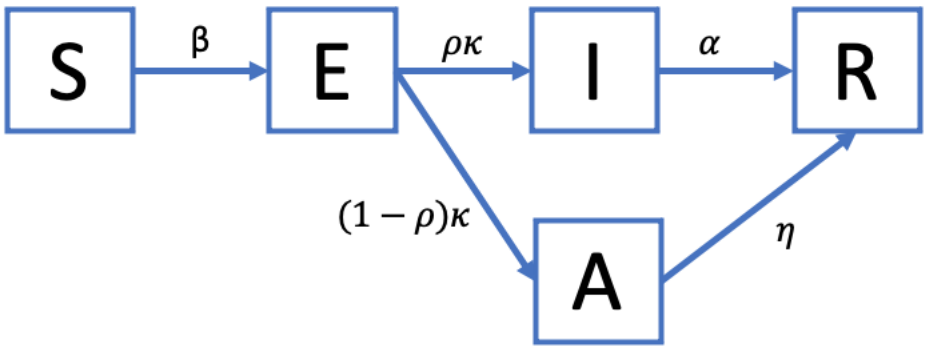



Figure 3: Simple SEIAR Compartment model

## Question 3 (10 points)

**3a) Using the model code above (in Part 1) as a guide, generate a new model that adds in an asymptomatic compartment similar to Figure 3 and using the updated parameter values and initial conditions from Table 2. This model will now assume that 1/3 of the infections are asymptomatic but that asymptomatic individuals are still equally infectious to symptomatic**

SEIAR table.png

| Parameter | Description | Value (units) |
|---|---|---|
| $\beta$ | Transmission rate | 0.5 |
| $\rho$ | Proportion of infections that are symptomatic | 2/3 = 0.67 |
| $\kappa$ | Latent period | 1/3 days = 0.333 |
| $\alpha$ | Infectious period for symptomatic infections (I) | 1/7 days = 0.143 |
| $\eta$ | Infectious period for asymptomatic infections (A) | 1/5 days = 0.2 |

| Initial conditions | Value |
|---|---|
| $S(0)$ | 9950 |
| $E(0)$ | 0 |
| $I(0)$ | 33 |
| $A(0)$ | 17 |
| $R(0)$ | 0 |
| $N = S(0) + E(0) + I(0) + A(0) + R(0) = 10{,}000$ | |

Figure 4: Updated parameter values and initial conditions for the SEIAR model.

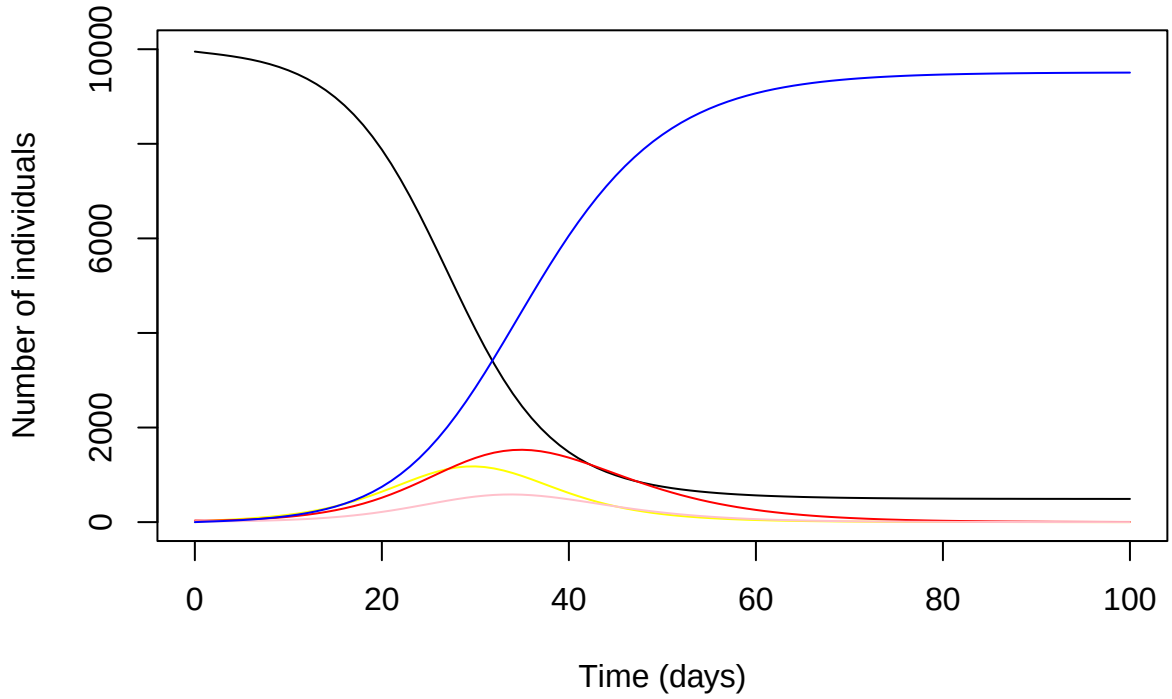
**individuals. However, we are going to also assume that asymptomatic individuals recover more quickly (5 days instead of 7 days). Update the code below to run this new SEIAR model. Annotate the steps as you go using comments (10 points).**

```
require(deSolve)

 seiarmod=function(t, y, parms){
   #Pull state variables from y vector
   S=y[1]
   E=y[2]
   I=y[3]
   A=y[4]
   R=y[5]
   #Pull the required parameter values from the parms vector
   beta=parms["beta"]
   kappa=parms["kappa"]
   alpha=parms["alpha"]
   rho=parms["rho"]
   eta=parms["eta"]
   N=parms["N"]
   #Define the equations (use what you wrote out in question 1) - 6 points
   dS = -(beta*S*(I+A)/N)
   dE = +(beta*S*(I+A)/N) - (rho*kappa*E) - ((1-rho)*kappa*E)
   dI = (rho*kappa*E) - (alpha*I)
   dA = ((1-rho)*kappa*E) - (eta*A)
   dR = (alpha*I) + (eta*A)
   res=c(dS, dE, dI, dA, dR)
   #Return list of gradients
   list(res)
 }
```

```r
times  = seq(0, 100, by=1)
parms  = c(beta=0.5, kappa=1/3, alpha=1/7, N=10000, rho=2/3, eta=1/5 )
start = c(S=9950, E=0, I=33, A=17, R = 0)

out = ode(y = start, times = times, func = seiarmod,
     parms = parms)
out=as.data.frame(out)
head(round(out, 3))
```

```
##   time        S      E      I      A      R
## 1    0 9950.000  0.000 33.000 17.000  0.000
## 2    1 9926.475 19.958 30.872 15.029  7.666
## 3    2 9903.503 33.854 32.434 15.070 15.138
## 4    3 9878.844 45.287 36.361 16.353 23.154
## 5    4 9851.003 56.215 42.054 18.517 32.210
## 6    5 9818.815 67.769 49.304 21.414 42.699
```

```r
plot(x = out$time, y = out$S, col = "black", ylab = "Number of individuals",
      xlab = "Time (days)", type = "l", xlim = c(0, 100), ylim = c(0,10000))
 lines(x = out$time, y = out$E, col = "yellow")
 lines(x = out$time, y = out$I, col = "red")
 lines(x = out$time, y = out$A, col = "pink")
 lines(x = out$time, y = out$R, col = "blue")
```

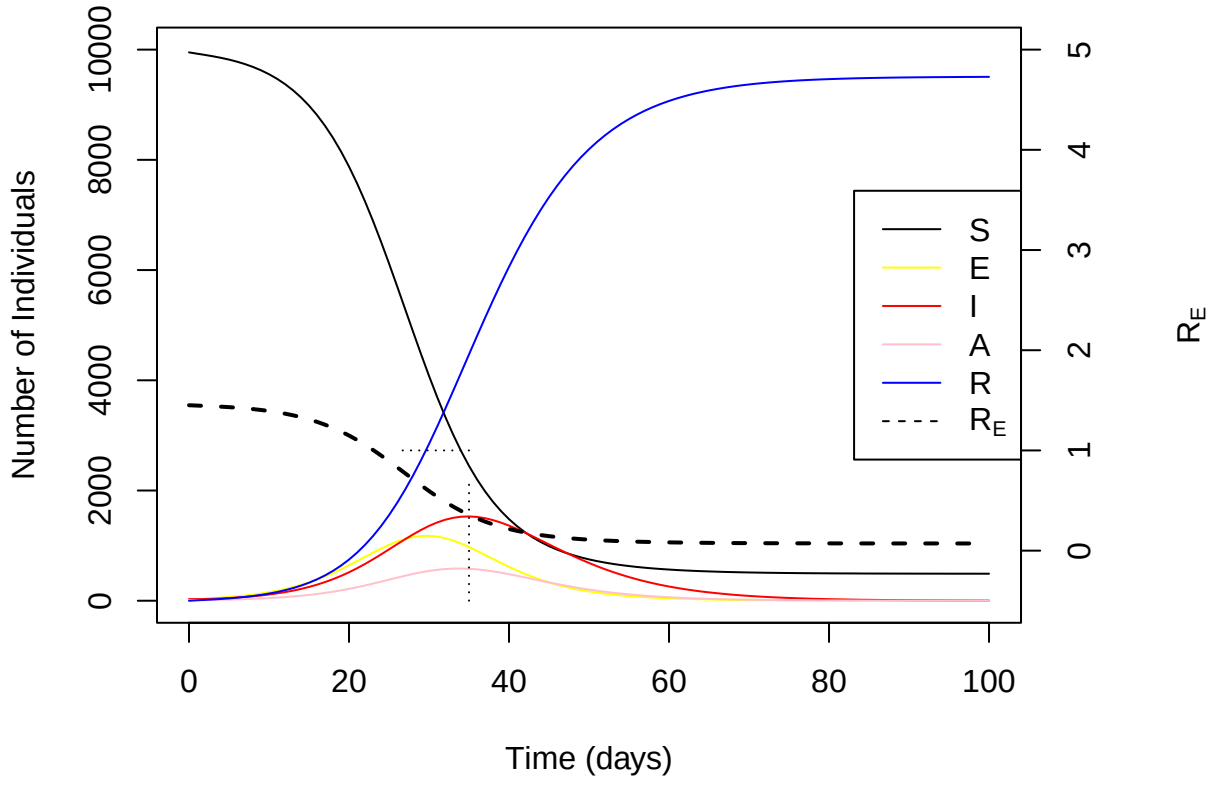



```r
#Calculate R0
R0 = parms["beta"]/(parms["alpha"]+parms["eta"])
R0
```

```
##     beta
## 1.458333
```

```r
#Adjust margins to accommodate a second right axis
par(mar = c(5,5,2,5))
```

```
#Plot state variables
plot(x = out$time, y = out$S, ylab = "Number of Individuals",
     xlab = "Time (days)",  type = "l", xlim=c(0,100), ylim=c(0,10000))
lines(x = out$time, y = out$E, col = "yellow")
lines(x = out$time, y = out$I, col = "red")
lines(x = out$time, y = out$A, col = "pink")
lines(x = out$time, y = out$R, col = "blue")

#Add vertical line at turnover point
xx = out$time[which.max(out$I+out$A)]
lines(c(xx,xx), c(1/R0,max(out$I+out$A)), lty = 3)

#prepare to superimpose 2nd plot
par(new = TRUE)
#plot effective reproductive ratio (w/o axes)
plot(x = out$time, y = R0*out$S/10000, type = "l", lty = 2,
     lwd = 2, col = "black", axes = FALSE, xlab = NA,
     ylab = NA, ylim = c(-.5, 5.0))
lines(c(xx, 26), c(1,1), lty = 3)
#Add right-hand axis for RE
axis(side = 4)
mtext(side = 4, line = 4, expression(R[E]))
#Add legend
legend("right", legend = c("S", "E", "I", "A", "R",expression(R[E])), lty = c(1,1,1,1,1,2),
     col = c("black", "yellow", "red", "pink", "blue", "black"))
```




```
require(rootSolve)
equil=runsteady(y=c(S=9950, E=0, I=33, A=17 , R = 0),
times=c(0,100), func=seiarmod, parms=parms)
```

```
round(equil$y, 3)
```

```
##        S        E        I        A        R
##  490.429    0.387    2.948    0.512 9505.724
```

# Question 4 (10 points)

**4a) How does this change to the structure of the model and assumptions change the outbreak that is observed compared to the SEIR model in Part 1? (6 points) (HINT: comment on the shape of the curve, outbreak timing, outbreak reproductive number, outbreak final size).**

**4b) Why does the outbreak behave differently? What is happening with the biology in this system compared to the SEIR system? (4 points)**