NABUKEERA LYDIA, 2022/HD07/1969U

Solving the SIR model

Adapted from; Infectious Disease modelling workshop Day 1, session3 computer lab - September 2021 (ILRI Nairobi & University of Warwick UK)

```r
# Check what packages are installed.
my_packages <- rownames(installed.packages())

# Install desolve if it is missing.
install.packages(setdiff("deSolve", my_packages))
```

Walkthrough: how to numerically solve an SIR model

A novel pathogen has been discovered in 1 individual in a population of 1000 individuals. It has been found to have an infectious period of 4 days and a daily transmission rate of 1.7. You are to analyse the spread of this pathogen during the first 30 days of the outbreak using an SIR model. We will find, plot and save the solution.

Step 1: write a function that computes the system of ODEs and returns a list containing the derivatives. The input parameters of this function are time `t`, the compartments as a vector `x`, and model parameters `parameters`. The outputs of the function are a list of derivatives. We name this function as `SIR.model`.

```r
SIR.model <- function(t, x, parameters){
  # Get the current number of individuals in each compartment.
  S <- x[1]
  I <- x[2]
  R <- x[3]

  # Get the model parameters.
  beta <- parameters$beta
  gamma <- parameters$gamma

  # Calculate the total number of individuals in the system.
  # This is just all the compartments added together.
  N <- S + I + R

  # Calculate the derivatives (the system of ODEs).
  dS <- -beta*S*I/N
  dI <- beta*S*I/N - gamma*I
  dR <- gamma*I

  # Return the derivatives inside a list.
  derivatives <- list(c(dS, dI, dR))
  return(derivatives)
}
```

Step 2: define the initial conditions, parameters and times at which you wish to compute the solution.

```
# The initial parameters: how many people are in each compartment at the start of
# the model?
init <- c(S = 999, I = 1, R = 0.0)

# The model parameters: what is the transmission rate (beta), and recovery rate (gamma).
pars <- list(beta = 1.7, gamma = 0.25)

# The solution times: the sequence of times for which to compute the solution.
# This will be numbers 0 to 30.
time <- seq(from = 0, to = 30, by = 1)
```

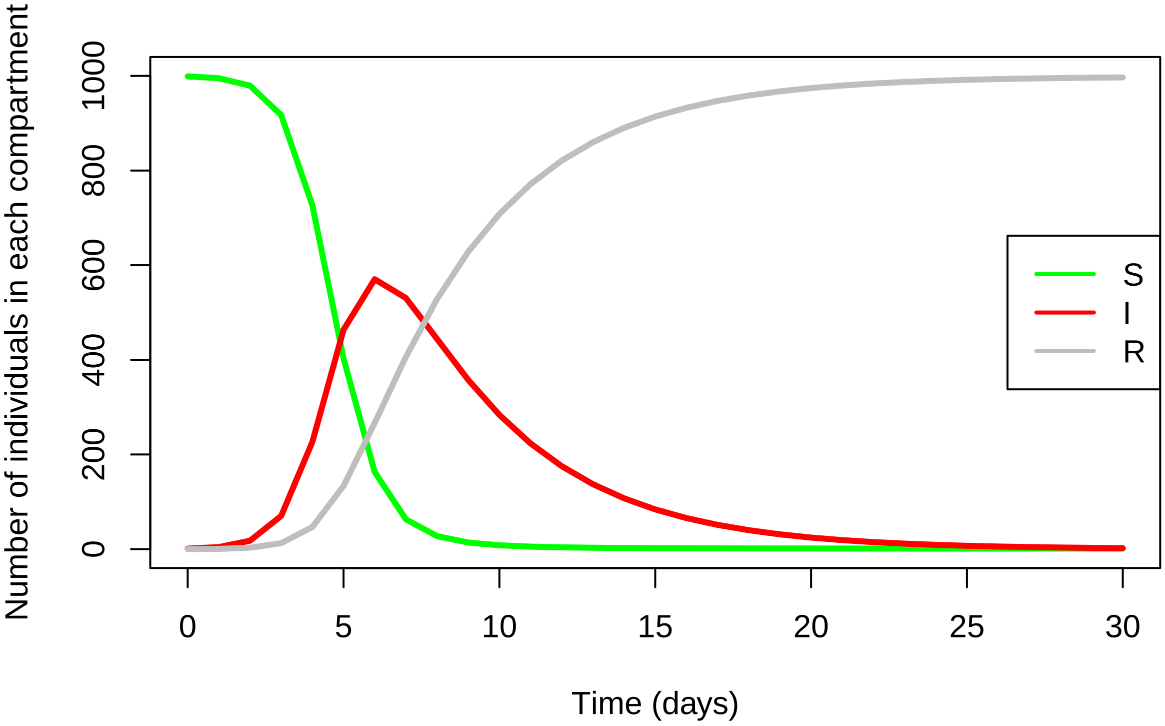
Step 3: solve the system using the 'lsoda' function from the 'deSolve' package (try help(lsoda) or ??Isoda for help). ## ode vs lsoda

```
# Load in the deSolve package.
library(deSolve)

# lsoda is a function with inputs: ODE system (as a function), initial conditions,
# parameters and times to compute the solution. The solution for the SIR model is
# stored in 'out'.
out <- lsoda(func = SIR.model, y = init, parms = pars, times = time)

# Convert the solution into a data.frame. This makes plotting easier.
out <- as.data.frame(out)
head(out)
```

```
##   time        S         I           R
## 1    0 999.0000   1.00000   0.0000000
## 2    1 995.1932   4.24533   0.5614516
## 3    2 979.3072  17.76493   2.9278517
## 4    3 917.3157  70.13967  12.5445858
## 5    4 726.7215 226.48275  46.7957389
## 6    5 403.1016 463.43270 133.4656895
```

Step 4: plot the solution for all compartments on a single plot.

```
# First plot time (x-axis) against the number of susceptible (y-axis).
plot(out$time, out$S,
     type = 'l', col = 'green', ylab = "Number of individuals in each compartment",
     xlab = "Time (days)", lwd = 3, ylim = c(0, 1000))

# Next, we can add lines for time against infecteds, and time against recovered/removed.
lines(out$time, out$I, col = 'red', lwd = 3)
lines(out$time, out$R, col = 'grey', lwd = 3)

# Finally, we can add a legend so we know what lines correspond to what compartment.
legend("right", legend = c("S","I","R"), col = c("green","red","grey"), lwd = 2)
```



Step 5: save the solution as a .csv file called "sir_model_demo.csv". This file will be saved in the same directory as the Rmarkdown file.

```
write.csv(out, "sir_model_demo.csv", row.names = FALSE)
```

Congratulations! You have now solved, plotted and saved the results of your first SIR model!

Exercises: exploring the SIR model

Now we can make some modifications to the model parameters, and observe how the model output changes.

Exercise 1

Find the number of individuals which will be infectious on days: 5, 10 and 25.4

Hint: combine the code from steps 2–3 in part A, and edit the times you want to compute the solution at in step 2…

```
TIME <- c(0, 5, 10, 25.4)
out <- lsoda(func = SIR.model, y = init, parms = pars, times = TIME)
out <- as.data.frame(out)
head(out)
```

```
##   time          S          I        R
## 1  0.0 999.000000   1.000000   0.0000
## 2  5.0 403.101095 463.433029 133.4659
## 3 10.0   8.089755 283.652787 708.2575
## 4 25.4   1.171354   6.389821 992.4388
```

```
# Extract the values of the "I" compartment for the specific time points.
infectious_day5 <- ceiling(out$I[2])
infectious_day10 <- ceiling(out$I[3])
infectious_day25_4 <- ceiling(out$I[4])
```

```
# Print the results
cat("Number of infectious individuals on day 5: ", round(infectious_day5), "\n")
```

```
## Number of infectious individuals on day 5:  464
```

```
cat("Number of infectious individuals on day 10: ", round(infectious_day10), "\n")
```

```
## Number of infectious individuals on day 10:  284
```

```
cat("Number of infectious individuals on day 25.4: ", round(infectious_day25_4), "\n")
```

```
## Number of infectious individuals on day 25.4:  7
```

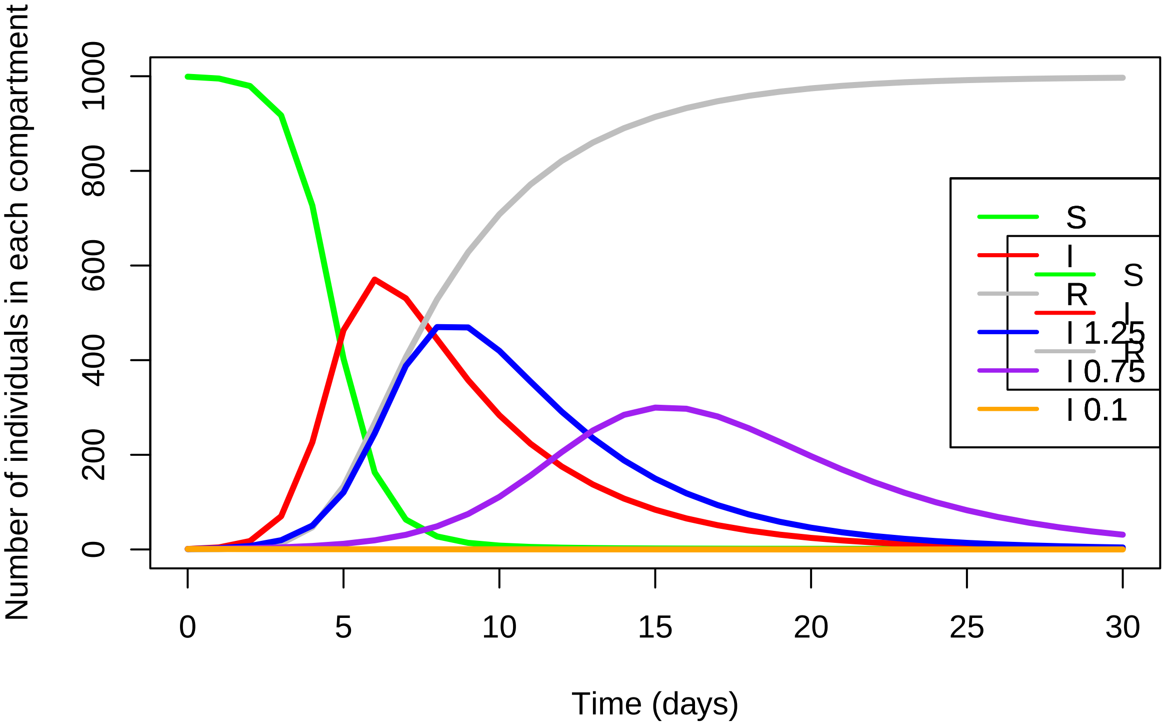
Exercise 2

Three interventions have been proposed which will lower the transmission rate. The expected transmission rates under each intervention are expected to be: 1.25, 0.75 and 0.1. For each of these interventions, find and plot the solution over 30 days.

Hint: for each new transmission rate, you can re-use your code in steps 2–4 of the walkthrough. You will need to change the model parameters you specify in step 2.

Advanced: if you are familiar with R programming, you are welcome to test yourself by completing this problem using using a for loop.

```
# Step 1: Define the SIR.model function
SIR.model <- function(t, x, parameters){
  S <- x[1]
  I <- x[2]
  R <- x[3]

  beta <- parameters$beta
  gamma <- parameters$gamma

  N <- S + I + R

  dS <- -beta*S*I/N
  dI <- beta*S*I/N - gamma*I
  dR <- gamma*I

  derivatives <- list(c(dS, dI, dR))
  return(derivatives)
}

# Step 2: Define initial conditions, parameters, and times
init <- c(S = 999, I = 1, R = 0.0)
pars <- list(beta = 1.7, gamma = 0.25)
time <- seq(from = 0, to = 30, by = 1)

# Step 3: Solve the system for the baseline scenario
library(deSolve)
out <- lsoda(func = SIR.model, y = init, parms = pars, times = time)
out <- as.data.frame(out)

# Step 4: Plot the solution for the baseline scenario
plot(out$time, out$S,
     type = 'l', col = 'green', ylab = "Number of individuals in each compartment",
     xlab = "Time (days)", lwd = 3, ylim = c(0, 1000))
lines(out$time, out$I, col = 'red', lwd = 3)
lines(out$time, out$R, col = 'grey', lwd = 3)
legend("right", legend = c("S","I","R"), col = c("green","red","grey"), lwd = 2)

# Step 5: Save the baseline solution as a .csv file
write.csv(out, "sir_model_demo.csv", row.names = FALSE)

# Step 6: Implement interventions using a for loop
beta_values <- c(1.25, 0.75, 0.1)
colors <- c("blue", "purple", "orange")

for (i in 1:length(beta_values)) {
  # Update beta value
  pars$beta <- beta_values[i]

  # Solve the system
  out <- lsoda(func = SIR.model, y = init, parms = pars, times = time)
  out <- as.data.frame(out)

  # Plot the solution
  lines(out$time, out$I, col = colors[i], lwd = 3)
  legend("right", legend = c("S","I","R", "I 1.25", "I 0.75", "I 0.1"),
         col = c("green","red","grey", colors[1], colors[2], colors[3]), lwd = 2)
}
```



```
# Step 7: Save the solution with interventions as a .csv file
write.csv(out, "sir_model_demo_interventions.csv", row.names = FALSE)
```

Exercise 3

Under each intervention, approximately how many people are infected in the first 30 days?

Hint: look at the number of susceptibles on the final day (day 30). Everyone on the final day who is not susceptible must have been infected at some point during the course of the epidemic. Recall there were 1000 individuals in the population at the start of the epidemic.

Intervention with transmission rate of 1.25

```r
pars <- list(beta = 1.25, gamma = 0.25)

# lsoda is a function with inputs: ODE system (as a function), initial conditions,
# parameters and times to compute the solution. The solution for the SIR model is
# stored in 'out'.
out <- lsoda(func = SIR.model, y = init, parms = pars, times = time)

# Convert the solution into a data.frame. This makes plotting easier.
out <- as.data.frame(out)

#Number of people infected in the first 30 days.
cat("For intervention with transmission rate of 1.25, the number of people infected is " , ceiling(1000
```

```
## For intervention with transmission rate of 1.25, the number of people infected is 993.
```

Intervention with transmission rate of 0.75

```r
pars <- list(beta = 0.75, gamma = 0.25)

# lsoda is a function with inputs: ODE system (as a function), initial conditions,
# parameters and times to compute the solution. The solution for the SIR model is
# stored in 'out'.
out <- lsoda(func = SIR.model, y = init, parms = pars, times = time)

# Convert the solution into a data.frame. This makes plotting easier.
out <- as.data.frame(out)

#Number of people infected in the first 30 days.
cat("For intervention with transmission rate of 0.75, the number of people infected is " , ceiling(1000
```

```
## For intervention with transmission rate of 0.75, the number of people infected is 934.
```

Intervention with transmission rate of 0.1

```r
pars <- list(beta = 0.1, gamma = 0.25)

# lsoda is a function with inputs: ODE system (as a function), initial conditions,
# parameters and times to compute the solution. The solution for the SIR model is
# stored in 'out'.
out <- lsoda(func = SIR.model, y = init, parms = pars, times = time)

# Convert the solution into a data.frame. This makes plotting easier.
out <- as.data.frame(out)

#Number of people infected in the first 30 days.
cat("For intervention with transmission rate of 0.1, the number of people infected is " , ceiling(1000-
```

```
## For intervention with transmission rate of 0.1, the number of people infected is 2.
```

Exercise 4

a) Calculate the expected R0 value of the disease under each intervention.

R0 = formula?

R0 under each intervention?

```
# Define the recovery rate
gamma <- 0.25

# Define the list of expected transmission rates for each intervention
beta_list <- c(1.25, 0.75, 0.1)

# Calculate the R0 value for each intervention
R0_list <- beta_list/gamma

# Print the R0 values for each intervention
cat("Expected R0 values for each intervention:\n")
```

```
## Expected R0 values for each intervention:
```

```
for (i in 1:length(R0_list)) {
  cat(paste0("Intervention ", i, ": R0 =", ceiling(R0_list[i]), "\n"))
}
```

```
## Intervention 1: R0 =5
## Intervention 2: R0 =3
## Intervention 3: R0 =1
```

b) How can these R0 values explain the behaviour in your model output?

Intervention 1 has an R0 value of 5, which means that each infected person is expected to transmit the disease to 5 other people. This is a relatively high R0 value, indicating that the disease is expected to spread quickly in the population without any intervention. An intervention with an R0 value of 5 could involve measures such as social distancing, mask-wearing, and limiting large gatherings. Implementing these measures would help to reduce the transmission rate of the disease, and lower the R0 value.

Intervention 2 has a lower R0 value of 3 compared to Intervention 1. This means that each infected person is expected to transmit the disease to 3 other people. This could indicate that more stringent measures are being implemented, such as lockdowns, contact tracing, or widespread vaccination. These measures are expected to reduce the transmission of the disease further, leading to a lower R0 value and fewer new infections.

Intervention 3 has the lowest R0 value of 1, which means that each infected person is expected to transmit the disease to only one other person. An R0 of 1 indicates that the disease is no longer spreading through the population. This could mean that the disease has been effectively controlled through measures such as mass vaccination, herd immunity, or effective treatments.