Worksheet: SIR and Gillespie algorithm

Nathan Green, Imperial College London

6/02/2020

1 Introduction

This practical will expore the use of the Gillespie algorithm for stochastic simulation of a Susceptible-Infective-Removed model.

Follow along the R code below and then answer the questions.

2 Basic model

We start by defining the input parameters. Set a single index case in a population of size 1000. We will run the model for at most 100 time steps (but if there are no infective present the simulation will end too).

```
# set rate parameters
params <- c(
  v = 0.001, # infection
  r = 0.05)  # removal

initial <- c(S = 999,
             I = 1,
             R = 0)
time.window <- c(0, 100)

# initialize state and time variables and write them into output
state <- initial
time <- time.window[1]
time_end <- time.window[2]

# define output dataframe
output <- data.frame(t = time,
                     S = initial["S"],
                     I = initial["I"],
                     R = initial["R"],
                     row.names = 1)
```

The main aglorithm as outlined in the lecture is below. We will use a `while()` loop which is similar to a `for()` loop. The difference is that it continues to loop around until the condition is met. Be careful though because this has the potential to get stuck in an infinite loop if that never happens.

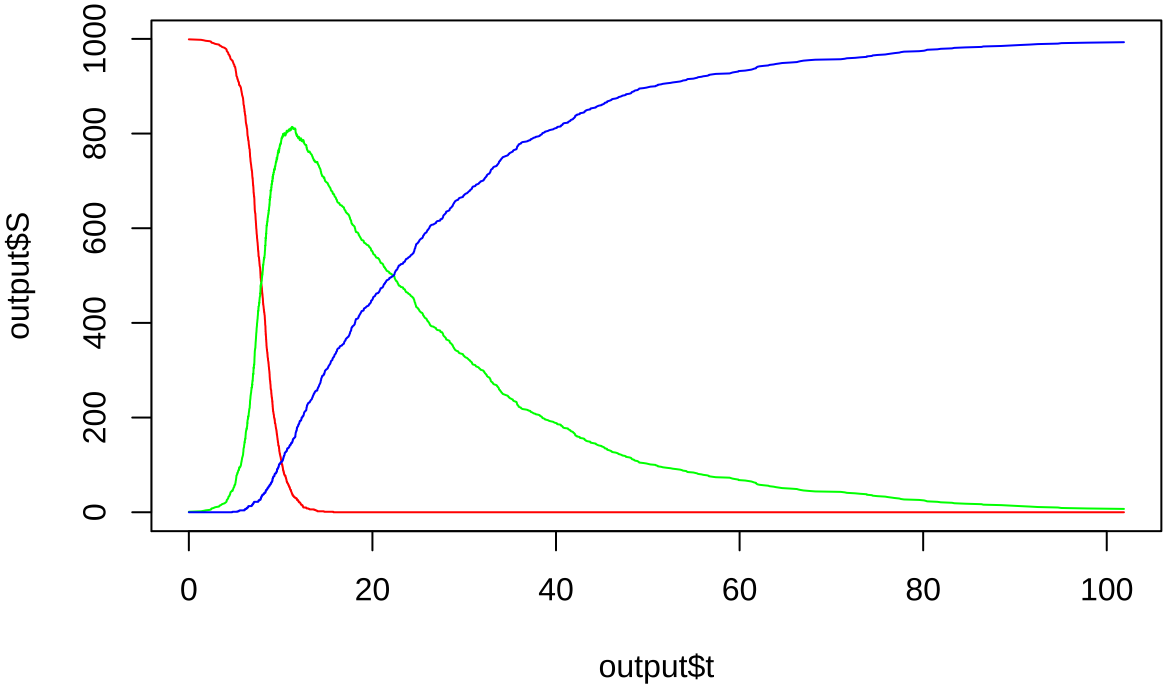
Notice that the code is composed of 2 main section the "when" and the "what".

```
while(time < time_end & state["I"] > 0){

  # generate event time
  w1 <- params["v"] * state["S"] * state["I"]
  w2 <- params["r"] * state["I"]
  W  <- w1 + w2
  dt <- -log(runif(1))/W
  time <- time + dt

  # select event
  if (runif(1) < w1/W) {
    state["S"] <- state["S"] - 1
    state["I"] <- state["I"] + 1
  } else {
    state["I"] <- state["I"] - 1
    state["R"] <- state["R"] + 1
  }

  output <- rbind(output,
                  c(time, state))
}

head(output)
```

```
##          t   S I R
## 1 0.000000 999 1 0
## 2 1.325990 998 2 0
## 3 1.619847 997 3 0
## 4 1.830674 996 4 0
## 5 2.286947 995 5 0
## 6 2.428728 994 6 0
```

We can plot the results of the simulation. The number of susceptibles over time is plots (red line). Then additional lines are added to this plot using the `lines()` command; the number of infectives and removed over time (green and blue, respectively).

```
plot(output$t, output$S, col = "red", type = "l")
lines(output$t, output$I, col = "green", type = "l")
lines(output$t, output$R, col = "blue", type = "l")
```



3 Your turn

Can you answer the following:

- What happend when you decrease the total population size? Rerun the same model and see how the output may change.
- Can you put the Gillespie code into a function which returns `output`?
- can you extend the model to inlude death? What would you need to change in the model set-up and main algorithm?