Solving SIR Models

Our goal is to observe the epidemic trajectory in a population given its β and γ parameters.

Setup

First of all let’s load the necessary libraries:

```
library(dplyr)
library(ggplot2)
library(knitr)
library(deSolve)
```

Paramters and initial values

Than lets declare the epidemic parameters to get an R_0 = 2:

```
parameters_values <- c(
  beta  = 0.5, # infectious contact rate (/person/day)
  gamma = 0.5 / 2    # recovery rate (/day) to get R_0 = 2
)
```

We need now the solving function to use with the ODE solver:

```
sir_equations <- function(time, variables, parameters) {
  with(as.list(c(variables, parameters)), {
    dS <- -beta * I * S
    dI <-  beta * I * S - gamma * I
    dR <-  gamma * I
    return(list(c(dS, dI, dR)))
  })
}
```

and finally the starting point of the epidemic. For now let’s consider a totally susceptible population with a small introduction event:

```
initial_values <- c(
  S = .999,  # number of susceptibles at time = 0
  I =   1 - .999,  # number of infectious at time = 0
  R =   0   # number of recovered (and immune) at time = 0
)
N <- 1000
```

Finally, let’s define a time period to observe:

```
time_values <- seq(0, 100)
```

Solver results

Let’s run the ODE solver:

```
sir_values <- ode(
  y = initial_values,
  times = time_values,
  func = sir_equations,
  parms = parameters_values
) %>% as.data.frame()
```

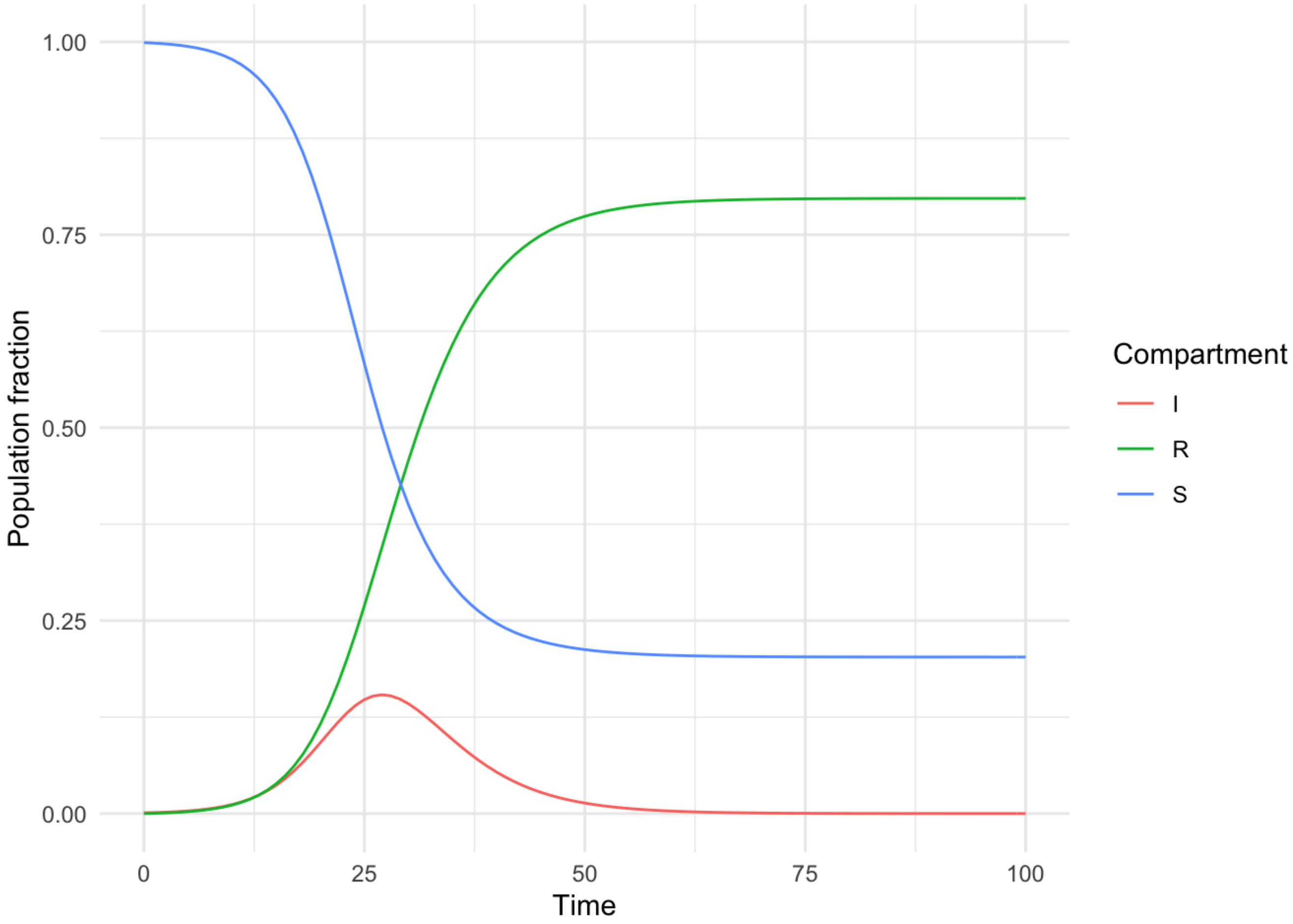
time	S	I	R
0	0.9990000	0.0010000	0.0000000
1	0.9984315	0.0012839	0.0002846
2	0.9977023	0.0016478	0.0006499
3	0.9967703	0.0021125	0.0011172
4	0.9955755	0.0027076	0.0017169
5	0.9940458	0.0034684	0.0024858

Use `ggplot` to create a SIR visualization function

```
plot_sir <- function(SIR) {
    SIR %>%
    # transform data into "long format"
    tidyr::pivot_longer(-time, names_to = 'Comp', values_to = 'Value') %>%
    # plot three lines, with a differnt color for each comparment
    ggplot(aes(time, Value, color = Comp)) +
    geom_line() +
    # plot formatting
    theme_minimal() +
    labs(x = "Time", y = "Population fraction", color = "Compartment")
}
```

and let’s test it

```
plot_sir(sir_values)
```



Solver packaging

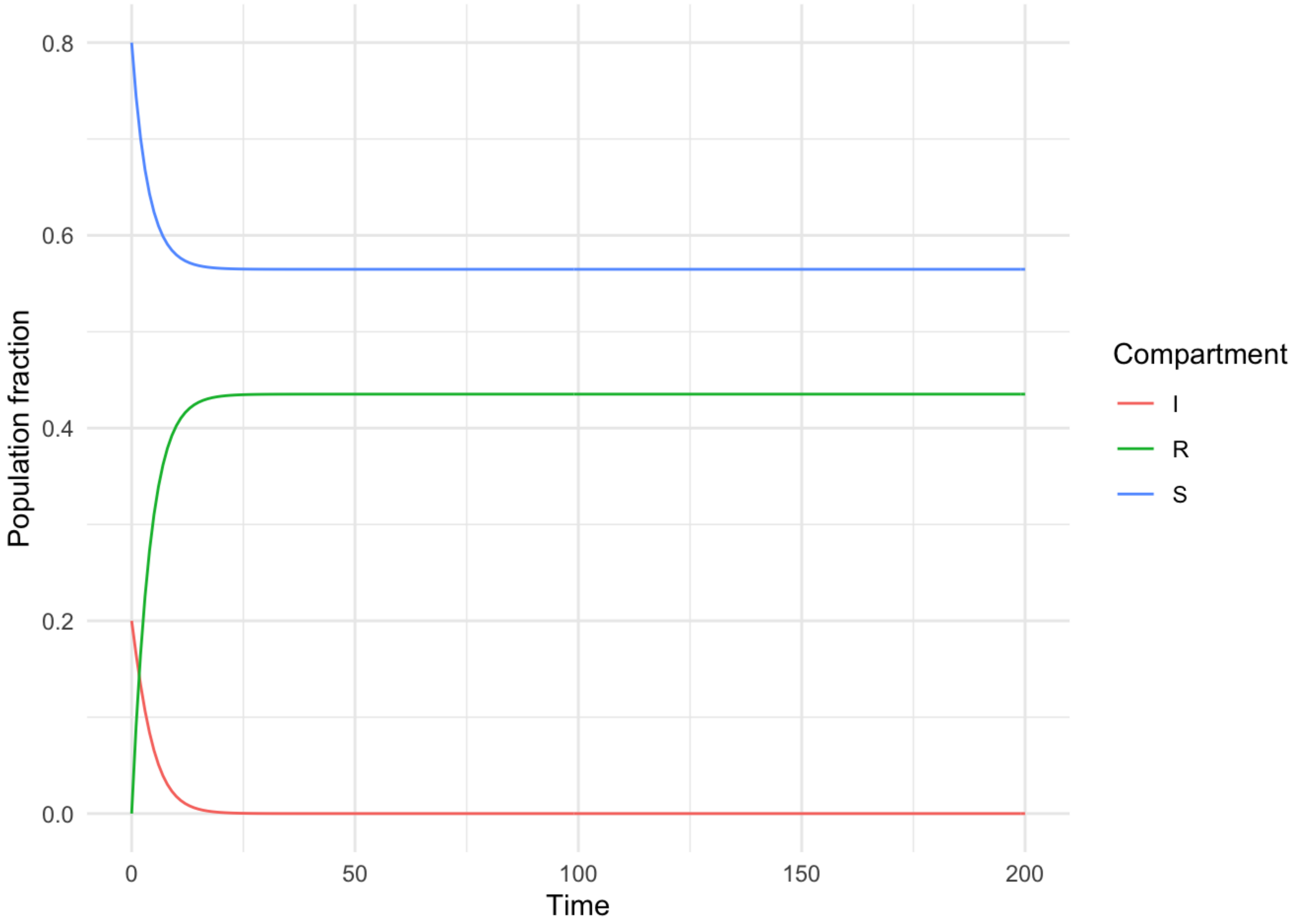
It may be helpful to package the SIR solver to make some experiments:

```
sir_fun <- function(beta, gamma, S0, I0 = 1 - S0, R0 = 0, times, N = 1) {

# the differential equations:
  sir_equations <- function(time, variables, parameters) {
  with(as.list(c(variables, parameters)), {
    dS <- -beta * I * S
    dI <-  beta * I * S - gamma * I
    dR <-  gamma * I
    return(list(c(dS, dI, dR)))
  })
  }

# the parameters values:
  parameters_values <- c(beta  = beta, gamma = gamma)

# the initial values of variables:
  initial_values <- c(S = S0, I = I0, R = R0)

# solving
  out <- ode(initial_values, times, sir_equations, parameters_values)

# returning the output:
  as.data.frame(out) %>%
    mutate(across(-time, ~ .x * N)) # Scale to population
}
```

R_0 < 1

What would happen with R_0 < 1 even with a large initial introduction?

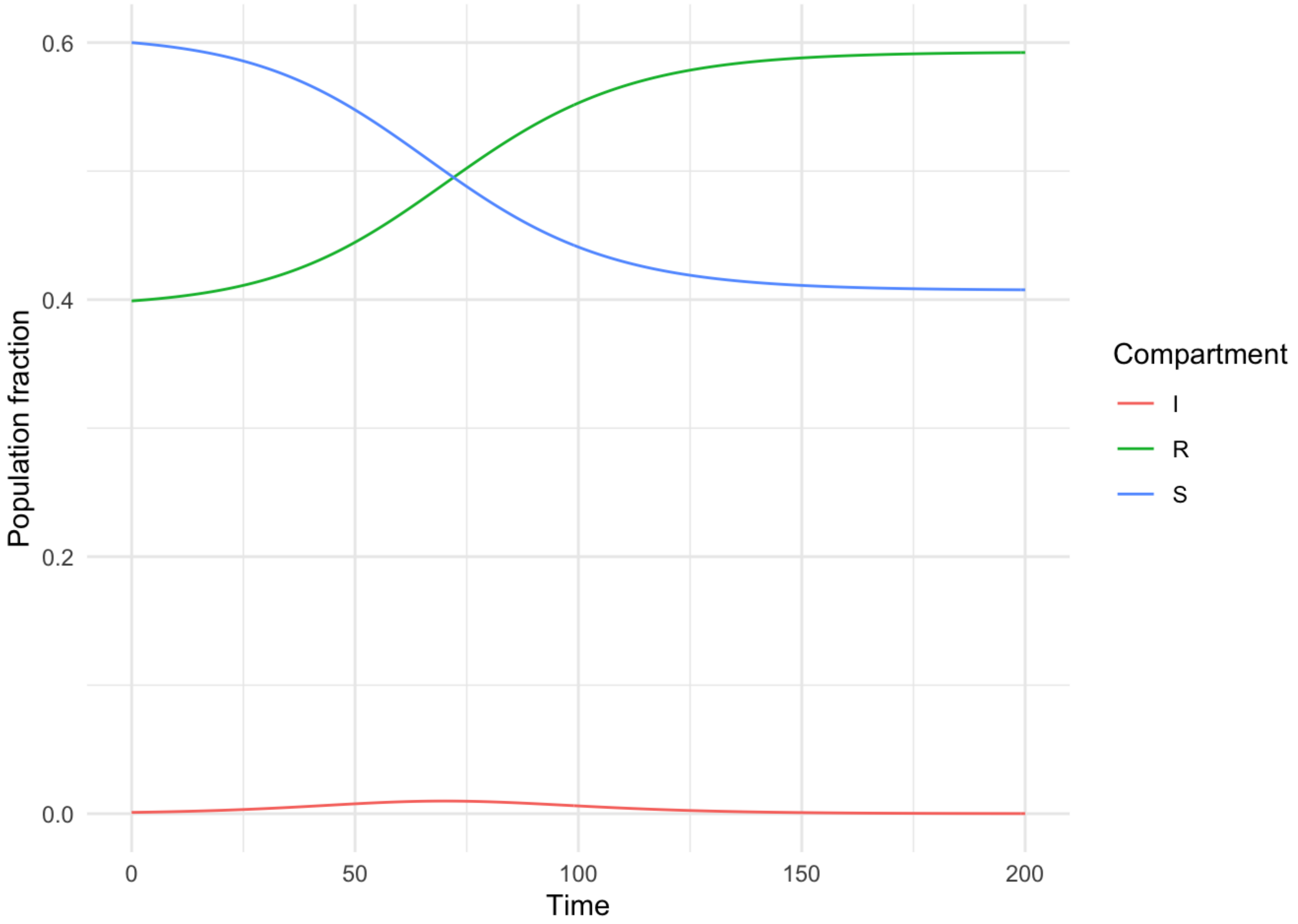
```
sir_fun(beta = 0.4, gamma = 0.5, .8, times = seq(0, 200)) %>%
    plot_sir()
```



Herd immunity threshold

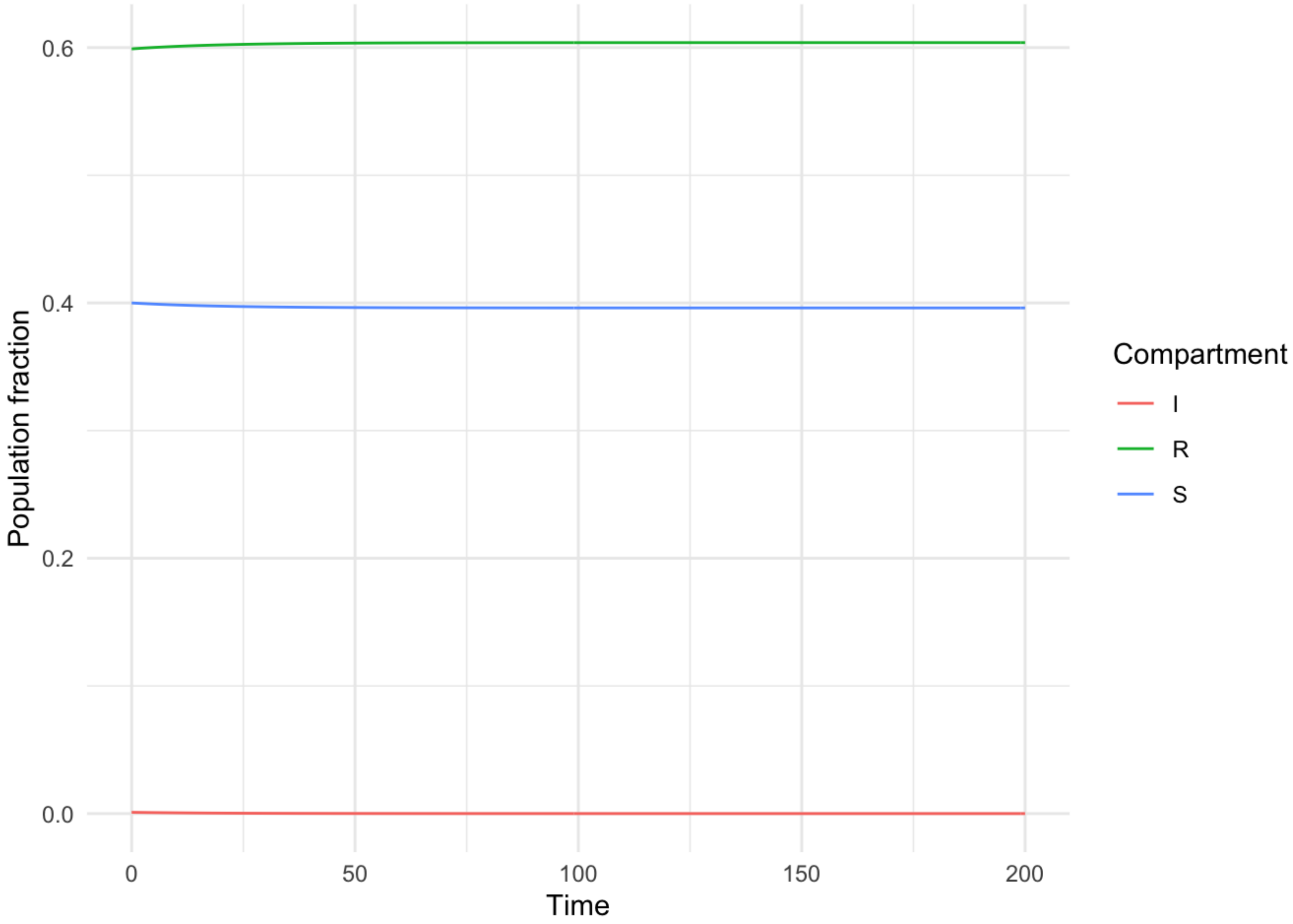
In this case we still use a large R_0 = 2, so the herd immunity threshold is 50%. What happens if the fraction of immunes is below the threshold?

```
sir_fun(0.5, .25, S0 = .6, I = .001, R = 0.399, times = seq(0, 200)) %>%
    plot_sir()
```



and how does it change if the vaccination rate overcome the threshold?

```
sir_fun(0.5, .25, S0 = .4, I = .001, R = 0.599, times = seq(0, 200)) %>%
    plot_sir()
```


0.6
0.4
0.2
0.0
Population fraction
0
50
100
150
200
Time
Compartment
I
R
S