Group1_Assignment6

Alyssa Obermayer, Vyoma Sheth, Shea Volkel, Grace Henderson, Taylor Kelsay, Sarita Emmanuel

11/7/2020

Question 1: Workplan (3 pts)

Describe your workplan for this assignment. How will you handle the larger group? How did you divide tasks? How well did the process work?

We decided to assign two people to working on each question, put it all together, and then meet to discuss how we figured out each question and sort out any issues. Sarita and Taylor worked on Question 2, Vyoma and Alyssa worked on Question 3, and Grace and Shea worked on question 4. For such a large group, it worked out fairly well because we divided the labor effectively. Each pair tackled their question, and we didn't have too much trouble explaining/understanding each other's answers to each question. This assignment probably would have been a lot of work for one person, so it was kind of nice.

Question 2: Threshold population size (10 pts)

In class we discussed the threshold population size and the reproductive number. Create an example that uses text and visuals to show how threshold population size and the reproductive number depend on beta and gamma. In your example imagine you are explaining the concepts to a non-biologist.

NOTES

beta=the transmission parameter (rate of infection for susceptible-infected contact). This transmission rate is the instantaneous rate for the number of new infections per unit time

gamma=recovery parameter (rate of infected transitioning to recovered)

The SIR model is governed by the differential equations in (1). Beta is the infection rate of the pathogen, and gamma is the recovery rate. Together, these two values give the basic reproduction number R0: the average number of secondary infections caused by an infected host. (https://towardsdatascience.com/infection-modeling-part-1-87e74645568a)

```
library(Ryacas)
```

```
## Warning: package 'Ryacas' was built under R version 4.0.3
```

```
##
## Attaching package: 'Ryacas'

## The following object is masked from 'package:stats':
##
##     integrate

## The following objects are masked from 'package:base':
##
##     %*%, diag, diag<-, lower.tri, upper.tri
```

```
library(deSolve)
```

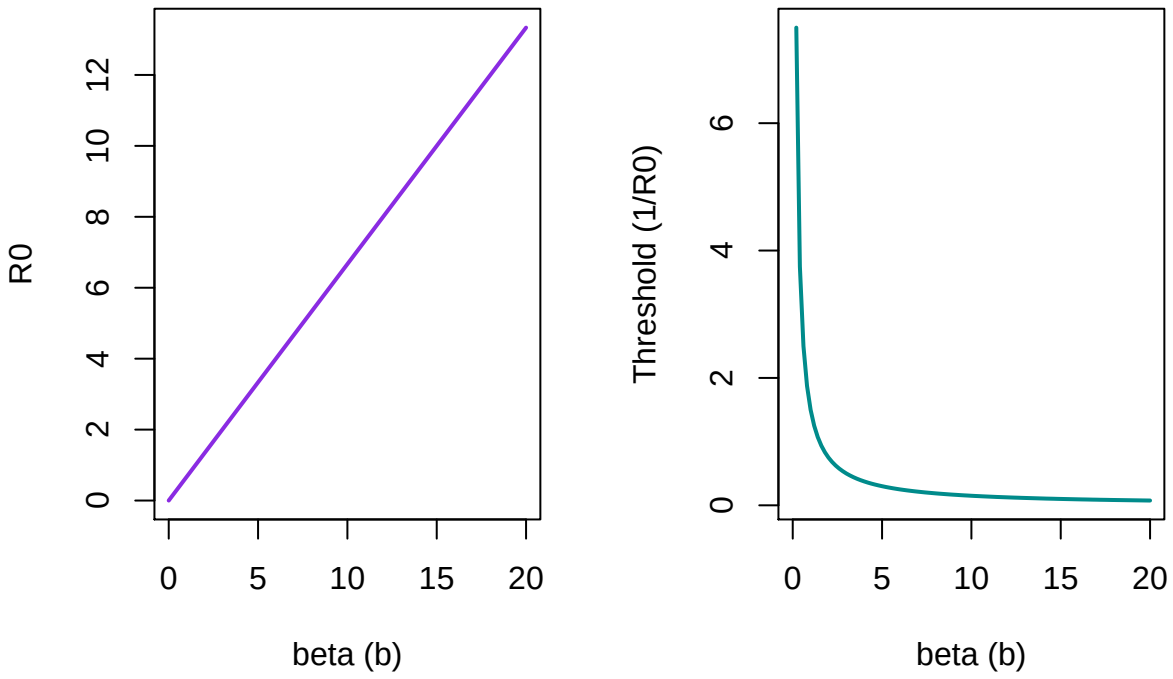
```
## Warning: package 'deSolve' was built under R version 4.0.3
```

To see how the reproductive rate (R0) and the threshold population size (1/R0) are affected by the beta and gamma parameters. We just varied each parameter one at a time while holding the other parameter values constant. Here we first vary the beta parameter, leaving gamma at a constant value of 1.5. We assume no net births/deaths, so d = 0.

```
R0 <- expression(b/(g + d))  #reproductive rate of the disease


# R0 and N dependent on beta
g <- 1.5  #gamma is set to a  constant
b <- seq(0, 20, by = 0.2)  #beta is given in a range of 0-20
d <- 0  #deaths is set to 0

R0 <- b/(g + d)
threshold_n <- 1/R0
threshold_n
```

```
##   [1]        Inf 7.50000000 3.75000000 2.50000000 1.87500000 1.50000000
##   [7] 1.25000000 1.07142857 0.93750000 0.83333333 0.75000000 0.68181818
##  [13] 0.62500000 0.57692308 0.53571429 0.50000000 0.46875000 0.44117647
##  [19] 0.41666667 0.39473684 0.37500000 0.35714286 0.34090909 0.32608696
##  [25] 0.31250000 0.30000000 0.28846154 0.27777778 0.26785714 0.25862069
##  [31] 0.25000000 0.24193548 0.23437500 0.22727273 0.22058824 0.21428571
##  [37] 0.20833333 0.20270270 0.19736842 0.19230769 0.18750000 0.18292683
##  [43] 0.17857143 0.17441860 0.17045455 0.16666667 0.16304348 0.15957447
##  [49] 0.15625000 0.15306122 0.15000000 0.14705882 0.14423077 0.14150943
##  [55] 0.13888889 0.13636364 0.13392857 0.13157895 0.12931034 0.12711864
##  [61] 0.12500000 0.12295082 0.12096774 0.11904762 0.11718750 0.11538462
##  [67] 0.11363636 0.11194030 0.11029412 0.10869565 0.10714286 0.10563380
##  [73] 0.10416667 0.10273973 0.10135135 0.10000000 0.09868421 0.09740260
##  [79] 0.09615385 0.09493671 0.09375000 0.09259259 0.09146341 0.09036145
##  [85] 0.08928571 0.08823529 0.08720930 0.08620690 0.08522727 0.08426966
##  [91] 0.08333333 0.08241758 0.08152174 0.08064516 0.07978723 0.07894737
##  [97] 0.07812500 0.07731959 0.07653061 0.07575758 0.07500000
```

```
par(mfrow = c(1, 2))
plot(R0 ~ b, type = "l", lwd = "2", xlab = "beta (b)", ylab = "R0",
    col = "blueviolet")
plot(threshold_n ~ b, type = "l", lwd = "2", xlab = "beta (b)",
    ylab = "Threshold (1/R0)", col = "darkcyan")
```



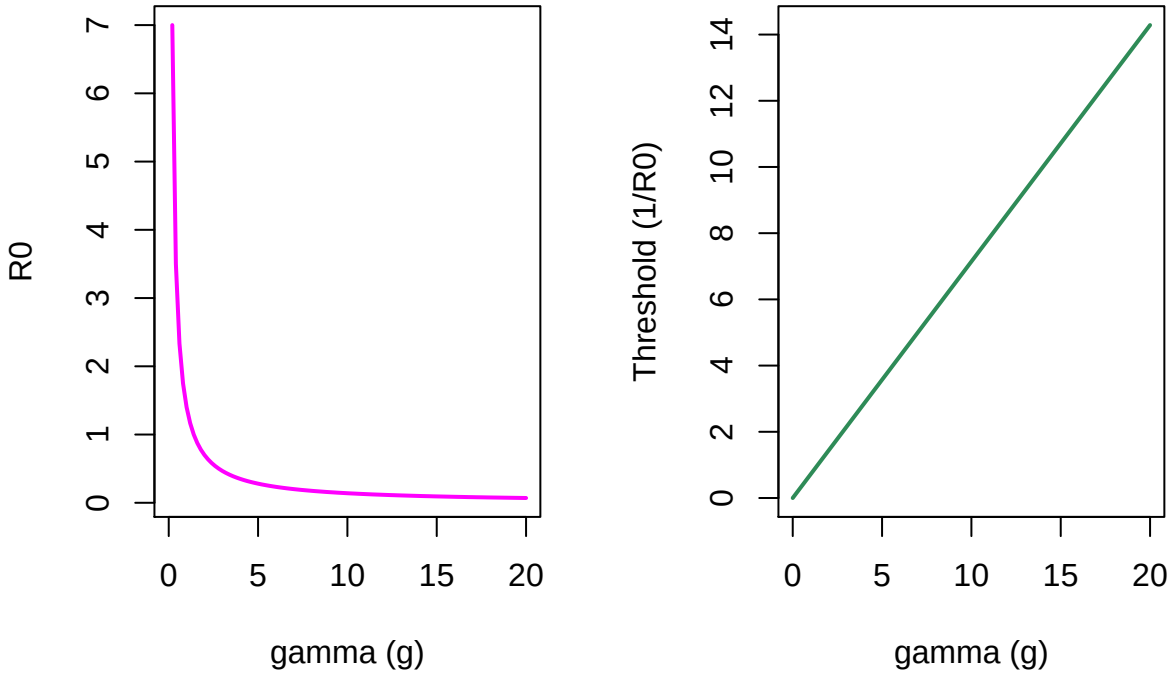
Now let's do the same thing, but instead keep beta at a constant value of 1.4 and vary gamma.

```
# R0 and N dependant on beta (keep beta and death constant)
g <- seq(0, 20, by = 0.2)  #range of values for gamma
b <- 1.4  # keep beta constant
d <- 0  # still no net deaths

R0 <- b/(g + d)
threshold_n <- 1/R0
threshold_n
```

```
##   [1]  0.0000000  0.1428571  0.2857143  0.4285714  0.5714286  0.7142857
##   [7]  0.8571429  1.0000000  1.1428571  1.2857143  1.4285714  1.5714286
##  [13]  1.7142857  1.8571429  2.0000000  2.1428571  2.2857143  2.4285714
##  [19]  2.5714286  2.7142857  2.8571429  3.0000000  3.1428571  3.2857143
##  [25]  3.4285714  3.5714286  3.7142857  3.8571429  4.0000000  4.1428571
##  [31]  4.2857143  4.4285714  4.5714286  4.7142857  4.8571429  5.0000000
##  [37]  5.1428571  5.2857143  5.4285714  5.5714286  5.7142857  5.8571429
##  [43]  6.0000000  6.1428571  6.2857143  6.4285714  6.5714286  6.7142857
```

```
##  [49]  6.8571429  7.0000000  7.1428571  7.2857143  7.4285714  7.5714286
##  [55]  7.7142857  7.8571429  8.0000000  8.1428571  8.2857143  8.4285714
##  [61]  8.5714286  8.7142857  8.8571429  9.0000000  9.1428571  9.2857143
##  [67]  9.4285714  9.5714286  9.7142857  9.8571429 10.0000000 10.1428571
##  [73] 10.2857143 10.4285714 10.5714286 10.7142857 10.8571429 11.0000000
##  [79] 11.1428571 11.2857143 11.4285714 11.5714286 11.7142857 11.8571429
##  [85] 12.0000000 12.1428571 12.2857143 12.4285714 12.5714286 12.7142857
##  [91] 12.8571429 13.0000000 13.1428571 13.2857143 13.4285714 13.5714286
##  [97] 13.7142857 13.8571429 14.0000000 14.1428571 14.2857143
```

```
par(mfrow = c(1, 2))
plot(R0 ~ g, type = "l", lwd = "2", xlab = "gamma (g)", ylab = "R0",
    col = "magenta")
plot(threshold_n ~ g, type = "l", lwd = "2", xlab = "gamma (g)",
    ylab = "Threshold (1/R0)", col = "seagreen")
```



DISCUSSION:

Basically R0 represents the rate of reproduction in a particular disease. This number (R0) will give us an average of the secondary infections per infected case, within a population. Therefore, to find R0, we must consider the rate at which people are becoming infected, which essentially is the rate of transmission, beta (b). In addition we consider the individuals who have become infected but recovered, or those that are resistant to the disease which is captured by gamma (g), the recovery rate. We would also have to consider deaths (d), but in this case, we are assuming no net birth/deaths and setting d = 0. We also must consider the population size (N), and what the threshold population size is (1/R0), which is the inverse of R0. This

threshold population size tells us how high the population needs to be for an epidemic to occur given the the reproductive rate (R0) of the disease. The disease will become an epidemic if the net number infections from one individual another is greater than 1 (if on average, an infected person is able to infect more than one other person at a time):

$R0N > 1$

Basically, this means there either has to be a large enough amount of people or a fast enough reproductive rate for the disease to become an epidemic. If the pathogen reproductive rate is slower, then the population size has to be larger for the pathogen spread. If the reproductive rate is faster, then the population does not have to be as large for the disease to spread.

This means if $R0N > 1$, then an outbreak can occur as the number of infectious cases will increase in the population (epidemic), if $R0N = 1$ the disease does not grow, and remains restricted to a certain population (endemic), and if $R0N < 1$ infections will decline, since there are more recoveries than infection spread.

Therefore the increase in transmissions (beta) positively affects the R0, as the graph illustrates that when beta increases, R0 increases proportionally, resulting in a positive relationship. If the transmission rate is high, it means more people are being infected, so likewise the R0, is consequently larger. However, the beta shows an inversely proportional relationship to the threshold value (1/R0). At low transmission rates (beta), the 1/R0 is much higher as it takes more people with infections to reach that threshold, before it results in an outbreak. Likewise the higher the transmission of the infectious disease, fewer infected people will attain the threshold much faster, as the R0 is higher. Basically, higher beta means sick people are infecting more people quickly, allowing the disease to spread quickly, and the population size does not need to be as large for an epidemic to happen.

With the recovery rate (gamma) and the relationship with R0, we see a slightly opposite relationship. The recovery rate shows a negative or inversely proportional development with the infectious rate of the disease (R0). When the rate of recovery is low, it means that the infection may be spending a longer time within the population, and as such influencing the infection rate, consequently showing that increase in R0. Basically, sick people are infectious for longer and infecting more people quickly, so population size does not need to be as large for an epidemic to occur. Likewise, if the recovery rate is high, the duration of the infectious disease if less in the population, reducing the spread, and thus resulting in R0 being at a lower value. The recovery rate shares a positive proportional relationship with the threshold population size. The threshold depends on the recovery rate, because gamma includes those individuals that are resistant but also those individuals that have been infected and moving towards recovery.Therefore as more people are recovering, there are less people to spread the disease, and therefore your threshold will be higher. Basically, higher gamma means sick people can recover faster and spreading the disease to less people more slowly, so you would need a larger population for the disease to become an epidemic.

Question 3

We chose to model the Ebola Virus because of its high mortality rate and high prevalence in West Africa, causing it to evolve into a major problem for public health. There is typically a rapid decline in health one week after the onset of symptoms which can lead to death after an average of 10 days in 50%-90% of infections.

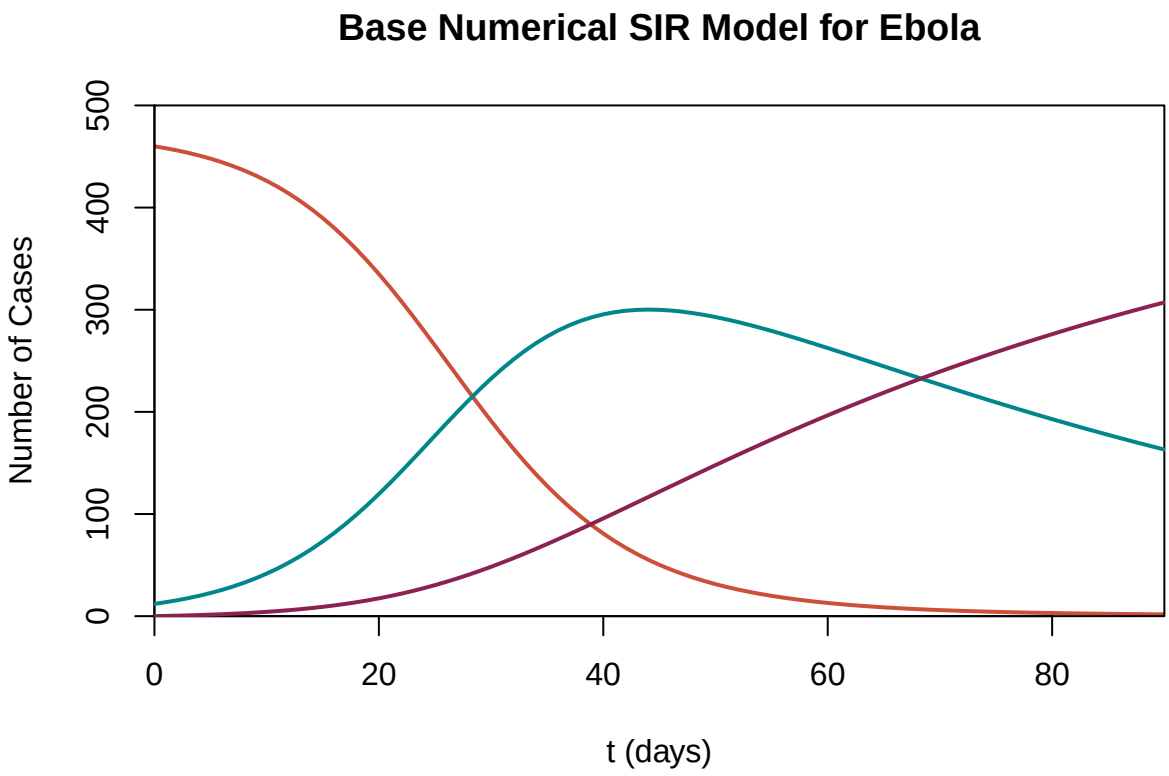
An important factor of why the Ebola Virus is important to model is the constant reemergence of the virus in various areas around West and Central Africa, large in part due to the animal and environmental reservoirs which the virus can reside in and be transmitted, through bodily fluids or nonhuman animals. Several bat species are possible reservoirs for the Ebola Virus with transmission to humans via feces, saliva, or food. This, in concurrence with the natural movement of bat colonies from Central to West Africa could account for multiple reemergence events of Ebola Virus in various populations over a relatively recent time-line (Alexander et al, 2015).

The use of compartmental host-pathogen modeling can play an important role in simulating these Ebola Virus reemergence events, as well as allow us to understand different aspects of the spread of the virus within

a population over time. Furthermore, the paper from Rachah and Torres (2015) presents an additional perspective in adding the effect of vaccination on the infected population and helps model the possible effect that may have on an outbreak over time. Overall, these models and the parameters used within them can help the researchers and public health officials determine if an infectious disease has the possibility of becoming an epidemic or spreading further and the possible actions that are necessary to take to contain the pathogen.

The following figure shows the SIR compartmental model based on numerical data obtained from the World Health Organization (WHO) of the 2014 Ebola Virus outbreak in Liberia. This model takes the starting data from the outbreak with a total population of 472 individuals, with 12 infected individuals, and 0 recovered individuals, along with a 0.000318 rate of infection (b) and a 0.0175 recovery rate (g). It then models the virus over a 90 day period showing that the number of infected individuals increases as the number of susceptible individuals decreases, likely because they are becoming infected. Additionally, as this occurs the number of recovered individuals increases at a slower rate but eventually surpasses the number of infected individuals, showing that the majority of the population has likely developed immunity to the virus. I will note that the research paper this figure was based on also plotted actual infection data obtained from the WHO but they did not put the exact numbers that were plotted in the material. I tried to obtain this data from the WHO pages that were referenced but was unable to find the concrete numbers I needed.

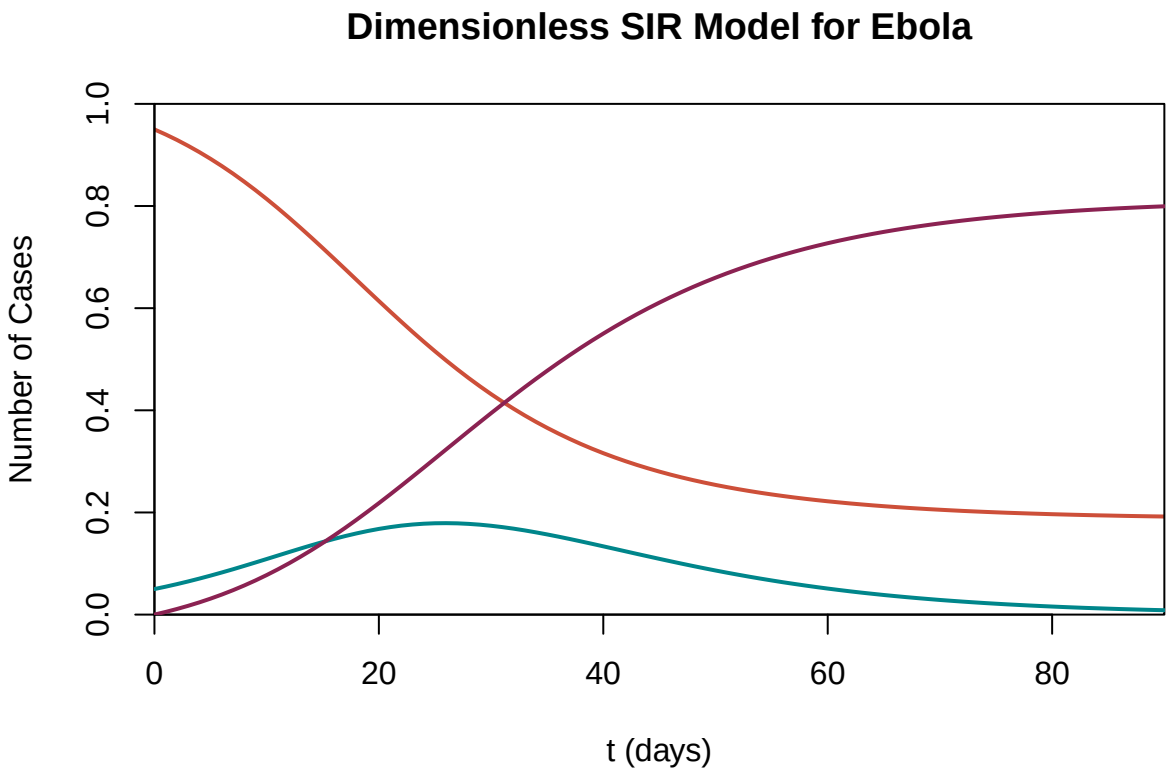
```
#SIR compartmental model function
SIR <- function (t, n, parms){
  with(as.list(parms), {
    S <- -b*n[1]*n[2]
    I <- b*n[1]*n[2] - g*n[2]
    R <- g*n[2]
    list(c(S, I, R))
  })
}

N <- 472      #Initial Population
b <- 0.000318 #Rate of infection
g <- 0.0175   #Recovery Rate
s0 <- 460     #Initial susceptible
i0 <- 12      #Initial Infected
r0 <- 0       #Initial recovered
initialN <- c(s0,i0,r0) #Initial population vector

#SIR ode function showing number of cases over 90 days
SIR.out <- ode(y = initialN, times = seq(0,90,by=1), func = SIR,
               parms=c(N=N,b=b,g=g),method="ode45")
#Plotting SIR Model
SIRplot <- plot(SIR.out[,2]~SIR.out[,1], #Plotting susceptible
                type="l", lwd=2, col="tomato3",
                xlab="t (days)",ylab="Number of Cases",
                xaxs="i", yaxs="i",
                xlim=c(0,90), ylim=c(0,500),
                main="Base Numerical SIR Model for Ebola")
lines(SIR.out[,3]~SIR.out[,1], lwd=2, col="turquoise4") #Plotting infected
lines(SIR.out[,4]~SIR.out[,1], lwd=2, col="violetred4") #Plotting recovered
```



The figure below shows a dimensionless SIR compartmental model of a virus starting with 95% of the population being susceptible, 5% of the population being infected, and 0% being recovered, along with a 0.2 rate of infection (b) and 0.1 recovery rate (g). This shows a similar pattern to the numerical data model though the number of cases at a given time are given in percentages of a population.

```
# SIR dimensionless compartmental model function
SIRd <- function(t, n, parms) {
    with(as.list(parms), {
        S <- -b * n[1] * n[2]
        I <- b * n[1] * n[2] - g * n[2]
        R <- g * n[2]
        list(c(S, I, R))
    })
}

N <- 1
b <- 0.2
g <- 0.1
s0 <- 0.95
i0 <- 0.05
r0 <- 0
# Initial parameters changed because model is now based off
# of percentages of the population
initialN <- c(s0, i0, r0)

SIR.outd <- ode(y = initialN, times = seq(0, 90, 1), func = SIRd,
```

```
    parms = c(N = N, b = b, g = g), method = "ode45")
SIRdplot <- plot(SIR.outd[, 2] ~ SIR.outd[, 1], type = "l", lwd = 2,
    col = "tomato3", xlab = "t (days)", ylab = "Number of Cases",
    xaxs = "i", yaxs = "i", xlim = c(0, 90), ylim = c(0, 1),
    main = "Dimensionless SIR Model for Ebola")
lines(SIR.outd[, 3] ~ SIR.outd[, 1], lwd = 2, col = "turquoise4")
lines(SIR.outd[, 4] ~ SIR.outd[, 1], lwd = 2, col = "violetred4")
```



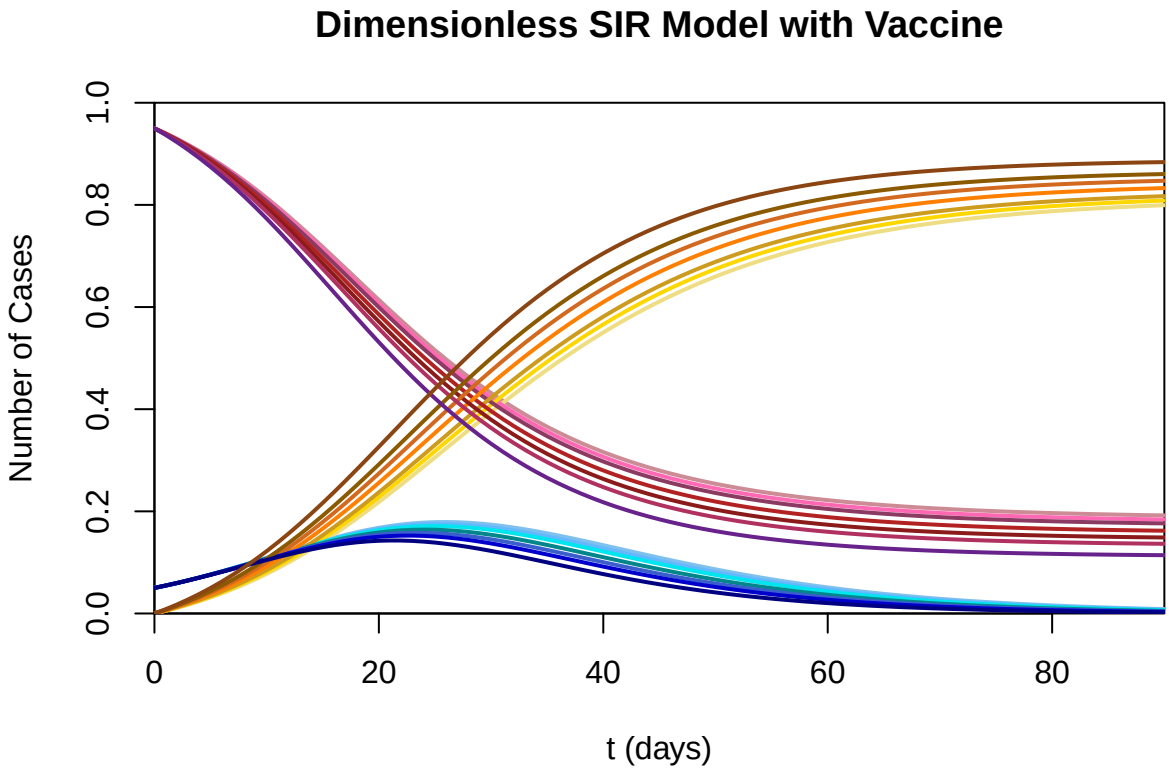
Lastly, the figure below represents a dimensionless SIR compartmental model of a virus with the introduction of varying vaccination rates ranging from 0 to 0.06. The addition of this parameter to the model decreases the number of days to peak infected individuals because as the rate of vaccination increases, those susceptible individuals that were vaccinated are able to transition straight to the recovered individuals group.

In the model shown, susceptible individuals are represented with reddish hues, infected individuals are represented with blue hues, and recovered individuals are represented with yellow/brown hues, additionally, the vaccination rate increases as these hues become darker in their respective categories. It can be seen that individuals who were susceptible to the Ebola Virus reduce gradually with increased vaccination and the beginning of the plateau of cases occurs in a shorter number of days. The cases for infected individuals is highest around 27 days but as vaccination rate increases the day to peak infections decrease. Moreover, the number of recovered individuals after 90 days increases from around 80% with no vaccine to 88% with a 6% vaccination rate of the population.

```
# SIR dimensionless compartmental model function with vaccine
# introduced
SIRv <- function(t, n, parms) {
    with(as.list(parms), {
```

```
        S <- -b * n[1] * n[2] - v * n[2]  #subtracting vaccinated individuals
        I <- b * n[1] * n[2] - g * n[2]
        R <- g * n[2] + v * n[2]  #adding vaccinated individuals
        list(c(S, I, R))
    })
}

N <- 1
b <- 0.2
g <- 0.1
v <- c(0, 0.005, 0.01, 0.02, 0.03, 0.04, 0.06)  #rate of vaccination, beginning at 0
s0 <- 0.95
i0 <- 0.05
r0 <- 0
initialN <- c(s0, i0, r0)

SIR.outv0 <- ode(y = initialN, times = seq(0, 90, 1), func = SIRv,
    parms = c(N = N, b = b, g = g, v = 0), method = "ode45")
SIR.outv0.005 <- ode(y = initialN, times = seq(0, 90, 1), func = SIRv,
    parms = c(N = N, b = b, g = g, v = 0.005), method = "ode45")
SIR.outv0.01 <- ode(y = initialN, times = seq(0, 90, 1), func = SIRv,
    parms = c(N = N, b = b, g = g, v = 0.01), method = "ode45")
SIR.outv0.02 <- ode(y = initialN, times = seq(0, 90, 1), func = SIRv,
    parms = c(N = N, b = b, g = g, v = 0.02), method = "ode45")
SIR.outv0.03 <- ode(y = initialN, times = seq(0, 90, 1), func = SIRv,
    parms = c(N = N, b = b, g = g, v = 0.03), method = "ode45")
SIR.outv0.04 <- ode(y = initialN, times = seq(0, 90, 1), func = SIRv,
    parms = c(N = N, b = b, g = g, v = 0.04), method = "ode45")
SIR.outv0.06 <- ode(y = initialN, times = seq(0, 90, 1), func = SIRv,
    parms = c(N = N, b = b, g = g, v = 0.06), method = "ode45")


plot(NA, type = "l", lwd = 2, xlab = "t (days)", ylab = "Number of Cases",
    xaxs = "i", yaxs = "i", xlim = c(0, 90), ylim = c(0, 1),
    main = "Dimensionless SIR Model with Vaccine")
lines(SIR.outv0[, 2] ~ SIR.outv0[, 1], lwd = 2, col = "lightpink3")
lines(SIR.outv0[, 3] ~ SIR.outv0[, 1], lwd = 2, col = "skyblue2")
lines(SIR.outv0[, 4] ~ SIR.outv0[, 1], lwd = 2, col = "lightgoldenrod")
lines(SIR.outv0.005[, 2] ~ SIR.outv0.005[, 1], lwd = 2, col = "hotpink")
lines(SIR.outv0.005[, 3] ~ SIR.outv0.005[, 1], lwd = 2, col = "steelblue1")
lines(SIR.outv0.005[, 4] ~ SIR.outv0.005[, 1], lwd = 2, col = "gold")
lines(SIR.outv0.01[, 2] ~ SIR.outv0.01[, 1], lwd = 2, col = "hotpink4")
lines(SIR.outv0.01[, 3] ~ SIR.outv0.01[, 1], lwd = 2, col = "turquoise2")
lines(SIR.outv0.01[, 4] ~ SIR.outv0.01[, 1], lwd = 2, col = "goldenrod3")
lines(SIR.outv0.02[, 2] ~ SIR.outv0.02[, 1], lwd = 2, col = "firebrick")
lines(SIR.outv0.02[, 3] ~ SIR.outv0.02[, 1], lwd = 2, col = "turquoise4")
lines(SIR.outv0.02[, 4] ~ SIR.outv0.02[, 1], lwd = 2, col = "darkorange1")
lines(SIR.outv0.03[, 2] ~ SIR.outv0.02[, 1], lwd = 2, col = "firebrick4")
lines(SIR.outv0.03[, 3] ~ SIR.outv0.03[, 1], lwd = 2, col = "royalblue3")
lines(SIR.outv0.03[, 4] ~ SIR.outv0.03[, 1], lwd = 2, col = "chocolate")
lines(SIR.outv0.04[, 2] ~ SIR.outv0.04[, 1], lwd = 2, col = "maroon")
lines(SIR.outv0.04[, 3] ~ SIR.outv0.04[, 1], lwd = 2, col = "mediumblue")
lines(SIR.outv0.04[, 4] ~ SIR.outv0.04[, 1], lwd = 2, col = "orange4")
```

```
lines(SIR.outv0.06[, 2] ~ SIR.outv0.02[, 1], lwd = 2, col = "darkorchid4")
lines(SIR.outv0.06[, 3] ~ SIR.outv0.06[, 1], lwd = 2, col = "navyblue")
lines(SIR.outv0.06[, 4] ~ SIR.outv0.06[, 1], lwd = 2, col = "saddlebrown")
```



References:

Alexander, Kathleen A et al. "What factors might have led to the emergence of Ebola in West Africa?." PLoS neglected tropical diseases vol. 9,6 e0003652. 4 Jun. 2015, doi:10.1371/journal.pntd.0003652

Amira Rachah, Delfim F. M. Torres, "Mathematical Modelling, Simulation, and Optimal Control of the 2014 Ebola Outbreak in West Africa", Discrete Dynamics in Nature and Society, vol. 2015, Article ID 842792, 9 pages, 2015. https://doi.org/10.1155/2015/84279

Question 4

Frequency-dependent transmission (10 pts). Implement an SIR model that you can switch between density and frequency dependent transmission. What are the most important differences in the dynamics? Illustrate these differences with at least two plots. Choose a parameter to vary between the models. How does this parameter interact with frequency dependent transmission?

Now, we will write out both the density dependent model and frequency dependent model. Both will have the same parameter values. For our basesline conditions in each model:

- Population size, N, is 1000. Only one individual will be infected and the rest will be susceptible initially.
- The transmission rate beta, b, is 0.01, which is relatively low. Density dependent and frequency dependent transmission rate are made of slightly different components, but we'll set both beta values to 0.01 here.
- There are no net births/deaths, d = 0, in the population. In the code, d denotes both the birth rate (typically shown with d misleadingly) death rate (typically denoted by mu). This is okay in this case, since they are equal to each other and we are going to keep it at zero. This assumption may not always be realistic.
- Gamma, g, is equal to 0. Infected individuals do not recover. (Infected for life.)
- Rho, p, is equal to 0. Recovered individuals (if there were any) cannot become susceptible again. (Lifelong immunity if recovered.)

First, write the function for the density dependent SIR model, and plot the results of infected and susceptible individuals over time. *We'll also show the equilibrium dynamics in another plot. Not sure about this*

```
# Density Dependent SIR Model
DD_SIR <- function(t, n, parms) {
    with(as.list(parms), {
        S <- -b * n[1] * n[2] + p * (N - n[1] - n[2]) + d * N -
            d * n[1]  # susceptible people
        I <- b * n[1] * n[2] - g * n[2] - d * n[2]  # infected people
        list(c(S, I))
    })
}

# Parameter values
N <- 1000  # population size of 1000
b <- 0.01  # transmission rate
p <- 0  # resistant return to susceptible (waning immunity), no loss of immunity
d <- 0  # birth/date rate (assume no net births/deaths)
g <- 0  # recovery rate of infected

initialN <- c(N - 1, 1)  # Only one individual infected initially

# Use ode function to calculate number of suspectible and
```

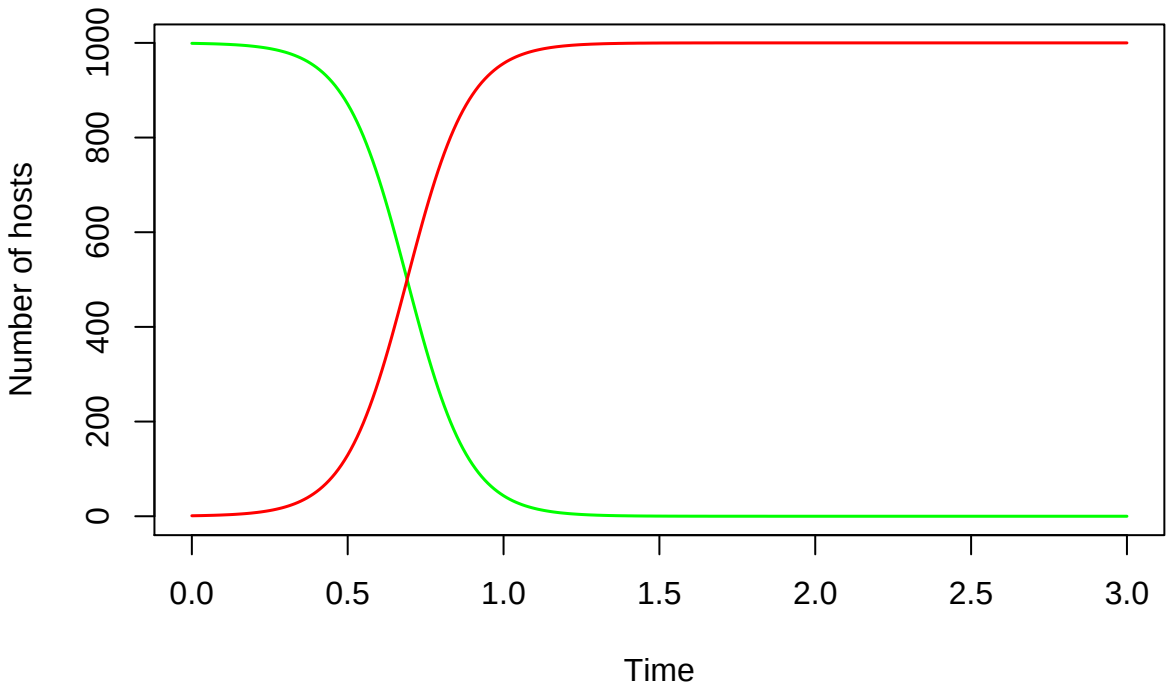
```r
# infected individuals over time based on parameter values
# and initial suspectible/infected conditions
DD_SIR.out <- ode(y = initialN, times = seq(0, 3, by = 0.01),
    func = DD_SIR, parms = c(N = N, b = b, p = p, d = d, g = g),
    method = "ode45")

# Show head and tail of results
head(DD_SIR.out)
```

```
##      time        1        2
## [1,] 0.00 999.0000 1.000000
## [2,] 0.01 998.8949 1.105055
## [3,] 0.02 998.7789 1.221132
## [4,] 0.03 998.6506 1.349387
## [5,] 0.04 998.5089 1.491091
## [6,] 0.05 998.3523 1.647652
```

```r
tail(DD_SIR.out)
```

```
##        time            1    2
## [296,] 2.95 1.541268e-07 1000
## [297,] 2.96 1.394597e-07 1000
## [298,] 2.97 1.261884e-07 1000
## [299,] 2.98 1.141800e-07 1000
## [300,] 2.99 1.033143e-07 1000
## [301,] 3.00 9.348266e-08 1000
```

```r
# Plot the results
plot(DD_SIR.out[, 2] ~ DD_SIR.out[, 1], type = "l", lwd = 1.5,
    col = "green", xlab = "Time", ylab = "Number of hosts")
lines(DD_SIR.out[, 3] ~ DD_SIR.out[, 1], lwd = 1.5, col = "red")
```




```
# Save results for later for comparison
DD_results_original_time <- DD_SIR.out[, 1]
DD_results_original_S <- DD_SIR.out[, 2]
DD_results_original_I <- DD_SIR.out[, 3]
```

In this case, as long as there is at least one infected individual to start out, the whole population will become infected (epidemic, or pandemic if population is global). No individuals are recovering and becoming susceptible again, and no net change in population means either new suspectibles are being added to the population and no one is dying. Or new susceptible individuals are becoming infected and replacing anyone who dies.

Now let's create the frequency dependent SIR model. For the susceptible equation, we instead use the more generalized value lambda, which is a combination of frequency dependent beta, b, and the number of infected individuals, I. In the case of frequency dependent transmission, lambda is based on a constant contact rate, c. Transmission, beta, for frequency dependent transmission is the product of the contact rate c, and the probability of transmission upon contact, v. For beta to equal 0.01 for this scenario, we picked values for the contact rate and probability of transmission where the product is equal to 0.01 (multiple combinations exist for this to happen).

```
# Frequency Dependent SIR Model
FD_SIR <- function(t, n, parms) {
    with(as.list(parms), {
        lambda <- c * v * n[2]/N  # frequency dependent lamba
        S <- -lambda * n[1] + p * (N - n[1] - n[2]) + d * N -
            d * n[1]  # susceptible people
        I <- (b * n[1] * n[2])/N - g * n[2] - d * n[2]  # infected people
```

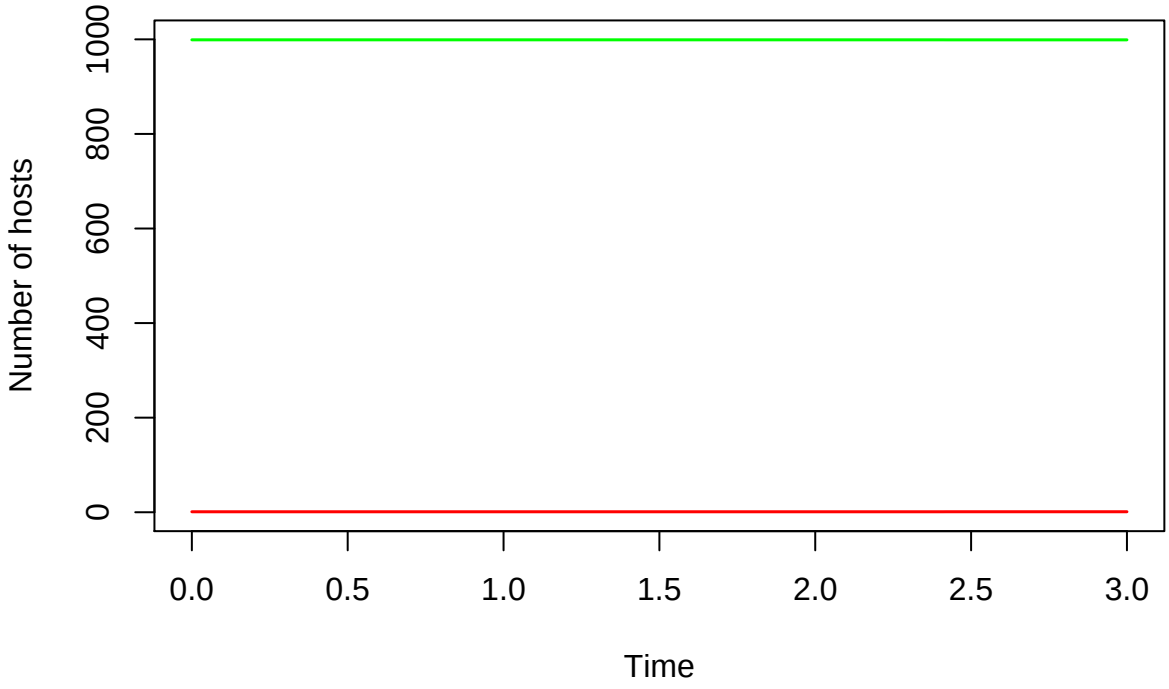
```
        list(c(S, I))
    })
}


c <- 1  # contact rate
v <- 0.01  # probability of transmission upon contact
N <- 1000  # population size of 1000
b <- c * v  # transmission rate of 0.01
p <- 0  # resistant return to susceptible (waning immunity), no loss of immunity
d <- 0  # birth/date rate (assume no net births/deaths)
g <- 0  # recovery rate of infected

initialN <- c(N - 1, 1)  # Only one individual infected initially

# Use ode function to calculate number of suspectible and
# infected individuals over time based on parameter values
# and initial suspectible/infected conditions
FD_SIR.out <- ode(y = initialN, times = seq(0, 3, by = 0.01),
    func = FD_SIR, parms = c(N = N, b = b, p = p, d = d, g = g),
    method = "ode45")

# Show head and tail of results
head(FD_SIR.out)
```

```
##      time        1      2
## [1,] 0.00 999.0000 1.0000
## [2,] 0.01 998.9999 1.0001
## [3,] 0.02 998.9998 1.0002
## [4,] 0.03 998.9997 1.0003
## [5,] 0.04 998.9996 1.0004
## [6,] 0.05 998.9995 1.0005
```

```
tail(FD_SIR.out)
```

```
##        time        1        2
## [296,] 2.95 998.9701 1.029909
## [297,] 2.96 998.9700 1.030011
## [298,] 2.97 998.9699 1.030114
## [299,] 2.98 998.9698 1.030217
## [300,] 2.99 998.9697 1.030320
## [301,] 3.00 998.9696 1.030423
```

```
# Plot the results
plot(FD_SIR.out[, 2] ~ FD_SIR.out[, 1], type = "l", lwd = 1.5,
    col = "green", xlab = "Time", ylab = "Number of hosts", ylim = c(0,
        1000))
lines(FD_SIR.out[, 3] ~ FD_SIR.out[, 1], lwd = 1.5, col = "red")
```




```r
# Save results for later for comparison
FD_results_original_time <- FD_SIR.out[, 1]
FD_results_original_S <- FD_SIR.out[, 2]
FD_results_original_I <- FD_SIR.out[, 3]
```

In this case, the disease is endemic. Overall, the number of infected individuals does not change noticeably over time for the given parameter values, though it does fractionally increase. Maybe this is because beta is too small.

For both the density dependent and frequency dependent model, let's see what change the transmission parameter, beta, does. Theoretically, we'd expect that people would get infected quicker in both cases, and more people in the freqency dependent case. Let's do it for the density dependent model first.

```r
# Density Dependent SIR Model
DD_SIR <- function(t, n, parms) {
    with(as.list(parms), {
        S <- -b * n[1] * n[2] + p * (N - n[1] - n[2]) + d * N -
            d * n[1]  # susceptible people
        I <- b * n[1] * n[2] - g * n[2] - d * n[2]  # infected people
        list(c(S, I))
    })
}

# Parameter values
N <- 1000  # population size of 1000
b <- 0.05  # transmission rate
```

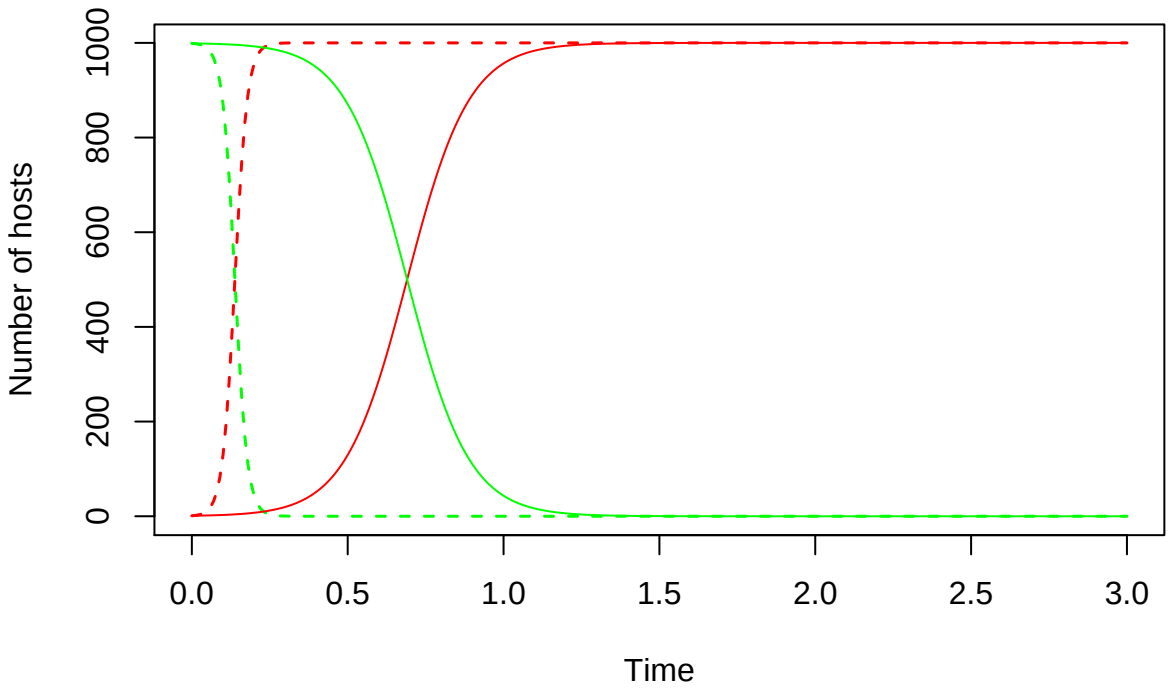
```r
p <- 0  # resistant return to susceptible (waning immunity), no loss of immunity
d <- 0  # birth/date rate (assume no net births/deaths)
g <- 0  # recovery rate of infected

initialN <- c(N - 1, 1)  # Only one individual infected initially

# Use ode function to calculate number of suspectible and
# infected individuals over time based on parameter values
# and initial suspectible/infected conditions
DD_SIR.out <- ode(y = initialN, times = seq(0, 3, by = 0.01),
    func = DD_SIR, parms = c(N = N, b = b, p = p, d = d, g = g),
    method = "ode45")

# Show head and tail of results
head(DD_SIR.out)
```

```
##      time        1         2
## [1,] 0.00 999.0000  1.000000
## [2,] 0.01 998.3523  1.647653
## [3,] 0.02 997.2864  2.713621
## [4,] 0.03 995.5339  4.466142
## [5,] 0.04 992.6578  7.342153
## [6,] 0.05 987.9522 12.047784
```

```r
tail(DD_SIR.out)
```

```
##        time            1    2
## [296,] 2.95 8.753980e-59 1000
## [297,] 2.96 5.309608e-59 1000
## [298,] 2.97 3.220471e-59 1000
## [299,] 2.98 1.953333e-59 1000
## [300,] 2.99 1.184768e-59 1000
## [301,] 3.00 7.186154e-60 1000
```

```r
# Plot the results compared with the orignal conditions



plot(DD_SIR.out[, 2] ~ DD_SIR.out[, 1], type = "l", lwd = 1.5,
    lty = 2, col = "green", xlab = "Time", ylab = "Number of hosts")
lines(DD_SIR.out[, 3] ~ DD_SIR.out[, 1], lwd = 1.5, col = "red",
    lty = 2)

lines(DD_results_original_I ~ DD_results_original_time, col = "red")
lines(DD_results_original_S ~ DD_results_original_time, col = "green")
```



As expected, increasing the transmission rate speeds up the epidemic. The coverse would also be true. Even relatively small changes in transmission rate speed things up a lot for the density dependent model. Therefore, the model is quite sensitive to this parameter and important to have a good estimate for so we may respond with countermeasures very quickly (which is unfortunately what we didn't do with COVID).

Now let's try for the frequency dependent case. Because we code beta to be made up of sub-components for this model in the susceptible equation, we'll just choose to increase one of the subcomponents. This suggests that increasing contact rate or increasing the probability of infection upon contact would ultimately have the same outcome if the product were the same either way (though probability can only reach a value of 1, so contact rate is likely the bigger player for a disease to become widespread). In real life, sometimes we cannot parse these out. Anyway, again we would expect people to become infected faster, but how much so in the case of frequency dependence?

```
# Frequency Dependent SIR Model
FD_SIR <- function(t, n, parms) {
    with(as.list(parms), {
        lambda <- c * v * n[2]/N  # frequency dependent lamba
        S <- -lambda * n[1] + p * (N - n[1] - n[2]) + d * N -
            d * n[1]  # susceptible people
        I <- (b * n[1] * n[2])/N - g * n[2] - d * n[2]  # infected people
        list(c(S, I))
    })
}


c <- 150  # contact rate
v <- 0.01  # probability of transmission upon contact
```

```r
N <- 1000  # population size of 1000
b <- c * v  # transmission rate
p <- 0  # resistant return to susceptible (waning immunity), no loss of immunity
d <- 0  # birth/date rate (assume no net births/deaths)
g <- 0  # recovery rate of infected

initialN <- c(N - 1, 1)  # Only one individual infected initially

# Use ode function to calculate number of suspectible and
# infected individuals over time based on parameter values
# and initial suspectible/infected conditions
FD_SIR.out <- ode(y = initialN, times = seq(0, 3, by = 0.01),
    func = FD_SIR, parms = c(N = N, b = b, p = p, d = d, g = g),
    method = "ode45")

# Show head and tail of results
head(FD_SIR.out)
```

```
##      time        1        2
## [1,] 0.00 999.0000 1.000000
## [2,] 0.01 998.9849 1.015098
## [3,] 0.02 998.9696 1.030423
## [4,] 0.03 998.9540 1.045980
## [5,] 0.04 998.9382 1.061771
## [6,] 0.05 998.9222 1.077800
```

```r
tail(FD_SIR.out)
```

```
##        time        1        2
## [296,] 2.95 922.8528 77.14718
## [297,] 2.96 921.7781 78.22191
## [298,] 2.97 920.6897 79.31032
## [299,] 2.98 919.5874 80.41256
## [300,] 2.99 918.4712 81.52876
## [301,] 3.00 917.3409 82.65906
```

```r
# Save these results
FD_results_time2 <- FD_SIR.out[, 1]
FD_results_S2 <- FD_SIR.out[, 2]
FD_results_I2 <- FD_SIR.out[, 3]


# We'll increase contact rate again to see how high contact
# rate needs to be to infect everyone (epidemic) Frequency
# Dependent SIR Model
FD_SIR <- function(t, n, parms) {
    with(as.list(parms), {
        lambda <- c * v * n[2]/N  # frequency dependent lamba
        S <- -lambda * n[1] + p * (N - n[1] - n[2]) + d * N -
            d * n[1]  # susceptible people
        I <- (b * n[1] * n[2])/N - g * n[2] - d * n[2]  # infected people
        list(c(S, I))
```

```
    })
}


c <- 500  # contact rate
v <- 0.01  # probability of transmission upon contact
N <- 1000  # population size of 1000
b <- c * v  # transmission rate of 0.01
p <- 0  # resistant return to susceptible (waning immunity), no loss of immunity
d <- 0  # birth/date rate (assume no net births/deaths)
g <- 0  # recovery rate of infected

initialN <- c(N - 1, 1)  # Only one individual infected initially

# Use ode function to calculate number of suspectible and
# infected individuals over time based on parameter values
# and initial suspectible/infected conditions
FD_SIR.out <- ode(y = initialN, times = seq(0, 3, by = 0.01),
    func = FD_SIR, parms = c(N = N, b = b, p = p, d = d, g = g),
    method = "ode45")

# Show head and tail of results
head(FD_SIR.out)
```

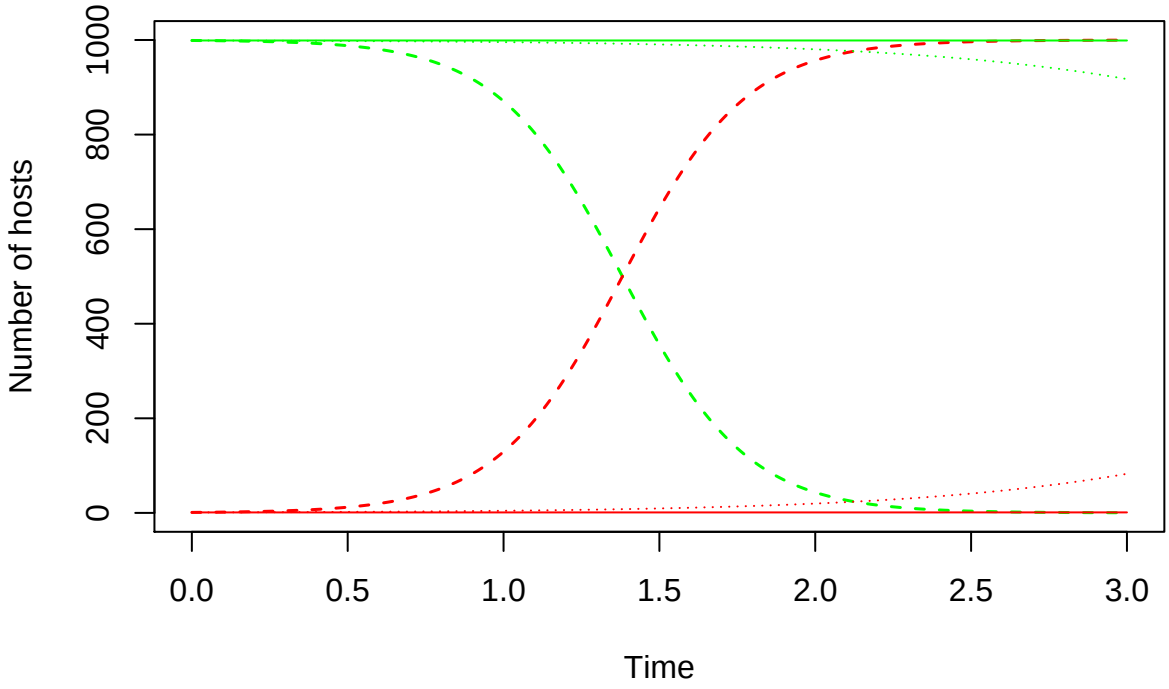
```
##      time        1        2
## [1,] 0.00 999.0000 1.000000
## [2,] 0.01 998.9488 1.051217
## [3,] 0.02 998.8949 1.105055
## [4,] 0.03 998.8384 1.161646
## [5,] 0.04 998.7789 1.221132
## [6,] 0.05 998.7163 1.283661
```

```
tail(FD_SIR.out)
```

```
##        time         1        2
## [296,] 2.95 0.3922397 999.6078
## [297,] 2.96 0.3731170 999.6269
## [298,] 2.97 0.3549264 999.6451
## [299,] 2.98 0.3376222 999.6624
## [300,] 2.99 0.3211615 999.6788
## [301,] 3.00 0.3055031 999.6945
```

```
# Plot the results
plot(FD_SIR.out[, 2] ~ FD_SIR.out[, 1], type = "l", lwd = 1.5,
    lty = 2, col = "green", xlab = "Time", ylab = "Number of hosts",
    ylim = c(0, 1000))
lines(FD_SIR.out[, 3] ~ FD_SIR.out[, 1], lwd = 1.5, col = "red",
    lty = 2)

lines(FD_results_original_I ~ FD_results_original_time, col = "red")
lines(FD_results_original_S ~ FD_results_original_time, col = "green")
```

```
lines(FD_results_I2 ~ FD_results_time2, col = "red", lty = 3)
lines(FD_results_S2 ~ FD_results_time2, col = "green", lty = 3)
```



Even with a large transmission rate (contact rate in this case), a frequency dependent pathogen does not spread very easily. If we think of STDs, people would have be be highly promiscous for the disease to spread broadly and rapidly. If contact rate is high enough though, a whole population can become infected, but it is much more difficult to acheive.

This means that density dependent pathogens are typically harder to prevent from spreading (probably COVID is an example seeing that extreme social distancing measures are only dampening the spread). Frequency dependent diseases can still be severe for those infected, and can still spread far for populations that are close-knit.