Lab/Homework #1 Answer Key

HUMBIO 154D/HRP 204

4/21/2020

1. Load packages

#if you are installing packages for the first time:  
#install.packages("deSolve")  
#install.packages("ggplot2")  
  
library(deSolve) #differential equation solver  
library(ggplot2) #plotting package

2. Start by defining a vector of model parameters (e.g. effective contact rate, recovery rate) and a vector of initial compartment sizes (often 1 infected person and the rest susceptible). We also define a vector of time steps corresponding to how long we want to run the model.

parameters <- c(beta = 0.5, #effective contact rate (aka transmission rate)  
 gamma = 0.3 #recovery rate (1/duration infection)  
)  
  
state <- c(S = 99999, #population of 100,000, 1 person starts of infected  
 I = 1,   
 R = 0  
)  
  
T\_end <- 500 #run model for 500 time steps  
times <- seq(0, T\_end, by = 1) #runs the model for 500 time steps, and computes output at each time step

Note: Your time steps could be days, months, years - but make sure time step size matches time scale of parameters. For example, if your time step is days, make sure you are using daily rates.    
Note: by naming the components of state (e.g. S=99999), and parameters (e.g. beta=0.5), R can match them with the names of parameters and states in your SIR model function (BasicSIR) below.  

3. Define SIR (and related) model functions. These will be used with the deSolve package to simulate how your population moves between compartments (e.g. Susceptible, Infected, Recovered) over time, given a set of parameters and the initial compartment sizes that we defined in step 2. We’ll start by writing a function for a basic SIR model without demography. For this and all subsequent models, we will assume frequency dependence (not density dependence).

BasicSIR<-function(t, state, parameters) {  
 with(as.list(c(state, parameters)),{ #this tells R that "S" refers to the "S" in the "state" vector, "beta" refers to the "beta" in "parameters", etc.  
   
 N = S + I + R #define N (total population size)  
   
 #SIR model equations from lecture - rates of change in and out of each compartment   
 dS <- -beta\*S\*I/N  
 dI <- beta\*S\*I/N - gamma\*I  
 dR <- gamma\*I  
   
 #return the rates of change as a list  
 list(c(dS, dI, dR))   
 })  
}

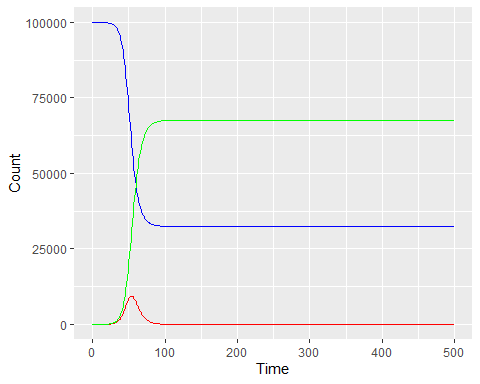
4. We’ve defined our SIR model function and its inputs. Now we can run it using the ode() function that is part of the deSolve package.

output <- ode(y = state, times = times, func = BasicSIR, parms = parameters)

Note: ode() is a function in the deSolve package. You must fill it in using the syntax: “ode(y= [vector of initial compartment sizes], times = [vector of time steps], func = [name of SIR model function], parms = [vector or list of parameter values])”

5. We can examine the model output by typing “View(output)” in the console, clicking on “output” in the Environment (top right of RStudio), or printing model output to the console. It is often easier to examine model output by plotting. We start by defining a function that will do basic cleaning and plot the epidemic curve of our model output from step #4.

show\_SIR\_model\_results<-function(output) {  
 df1 <- data.frame(output)  
 ggplot() +   
 geom\_line(data = df1, aes(x = time, y = S), color = "blue") +  
 geom\_line(data = df1, aes(x = time, y = I), color = "red") +  
 geom\_line(data = df1, aes(x = time, y = R), color = "green") +  
 xlab('Time') +  
 ylab('Count')   
}  
  
show\_SIR\_model\_results(output)



print(output[200:204,]) #prints rows 200 through 204, all columns

## time S I R  
## [1,] 199 32423.62 5.911470e-05 67576.38  
## [2,] 200 32423.62 5.150155e-05 67576.38  
## [3,] 201 32423.62 4.486888e-05 67576.38  
## [4,] 202 32423.62 3.909039e-05 67576.38  
## [5,] 203 32423.62 3.405610e-05 67576.38

print(head(output)) #head(object, x) refers to the first x rows of an object (default is 5 rows)

## time S I R  
## [1,] 0 99999.00 1.000000 0.0000000  
## [2,] 1 99998.45 1.221395 0.3321034  
## [3,] 2 99997.77 1.491801 0.7377318  
## [4,] 3 99996.94 1.822067 1.2331627  
## [5,] 4 99995.94 2.225439 1.8382746  
## [6,] 5 99994.70 2.718093 2.5773427

print(tail(output)) #tail(object, x) refers to the last x rows of an object (default is 5 rows)

## time S I R  
## [496,] 495 32423.62 9.152090e-22 67576.38  
## [497,] 496 32423.62 8.043093e-22 67576.38  
## [498,] 497 32423.62 7.068478e-22 67576.38  
## [499,] 498 32423.62 6.211961e-22 67576.38  
## [500,] 499 32423.62 5.459232e-22 67576.38  
## [501,] 500 32423.62 4.797714e-22 67576.38

6. What is the basic reproductive number, R0, for this model? What is the population prevalence at t=50? Is there an endemic equilibrium? Why/why not?

#calculate R0  
R0 <- parameters["beta"]/parameters["gamma"]  
print(R0)

## beta   
## 1.666667

#model-predicted prevalence at t=50  
output\_df <- data.frame(output)  
output\_df$N <- output\_df$S + output\_df$I + output\_df$R  
print(output\_df$I[50]/output\_df$N[50])

## [1] 0.0783328

S\_star <- output\_df$S[T\_end+1]/output\_df$N[T\_end+1] #S prevalence at equilibrium  
Rt\_star <- R0\*S\_star #effective reproductive number  
I\_star <- output\_df$I[T\_end+1]/output\_df$N[T\_end+1] #I prevalence at equilibrium  
  
print(Rt\_star)

## beta   
## 0.5403937

print(I\_star)

## [1] 4.797714e-27

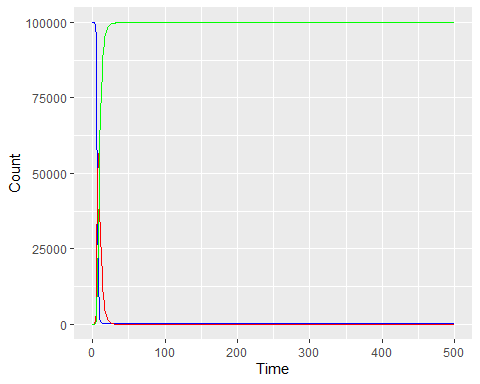
*Answer: The basic reproductive ratio is 1.67. The population prevalence at t=50 is 7.8% (meaning 7.8% of the population is infected during that time). The effective reproductive ratio at equilibrium is 0.54, which is less than 1, meaning that there will be no endemic equilibrium. This can observed from calculating the prevalence at the end of the model or observing the graph in part 5. In this case, there is no endemic equilibrium because there is no demography, so the population of susceptibles declines to a level at which the effective reproductive number is less than 1 (even though the basic reproductive number is greater than 1).*

7. How does the basic reproductive number change if we change beta to parameterize a contact rate (k) of 10 contacts per unit time and a 20% probability of infection per contact (p)? How do you expect this to change the initial epidemic dynamics and proportion of the population that gets infected? Plot the epidemic curve graph with this new beta and compare against your prediction.

#p=20%  
parameters <- c(beta = 2, #2 is 10\*20%  
 gamma = 0.3  
)  
R0 <- parameters["beta"]/parameters["gamma"]  
print(R0)

## beta   
## 6.666667

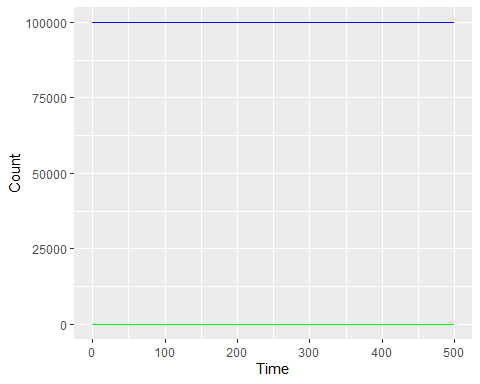
output <- ode(y = state, times = times, func = BasicSIR, parms = parameters)  
show\_SIR\_model\_results(output)



#what if p was 2% instead?  
parameters <- c(beta = 0.2, #0.2 is 10\*2%  
 gamma = 0.3  
)  
R0 <- parameters["beta"]/parameters["gamma"]  
print(R0)

## beta   
## 0.6666667

output <- ode(y = state, times = times, func = BasicSIR, parms = parameters)  
show\_SIR\_model\_results(output)

  
*Answer:The new beta equals 20% x 10, or 2. At beta=2 and gamma=0.3, the basic reproductive ratio is 6.7, which is even larger than 1. Thus we still expect an epidemic to occur. This is confirmed in the graph.*    
*Note: what if the probability of infection per contact was 2% instead of 20%? Then beta would be 0.2 and R0 would be 0.67. We would see that an epidemic does not occur when R0 < 1 (this code is also shown above).*

8. If we wanted to model a chronic infectious disease, what type of model would we use instead? Is there a way to model this using the BasicSIR function or do we need to write a new model function? Generate the epidemic curve plot for a chronic disease with beta=0.5. Does the population get infected faster or slower than in the previous version (part 5) and do more or fewer people get infected? Conceptually, why is this?

parameters <- c(beta = 0.5,  
 gamma = 0.0  
)  
output <- ode(y = state, times = times, func = BasicSIR, parms = parameters)  
show\_SIR\_model\_results(output)

  
*Answer: We would use an SI model to model a chronic disease, because infected individuals never recover, so there is no need for an R compartment. We could write a new model function using the SI model equations from class (effectively the same as BasicSIR, but removing the R line and removing recovery), or we could just set gamma, the recovery rate, to 0, which prevents individuals from leaving the I compartment. As observed in the graph, the epidemic dynamics move more quickly in this version of the model and everyone is eventually infected. This is because infected individuals are infected much longer (forever, essentially, so they can infect more people) rather than transitioning to an immune compartment, as in the SIR model.*

9. Now let’s add demography. To do this, we need to write a new model function (hint - use the OpenSIR function from Lab1\_functions.R) that includes births and deaths and we need to define additional parameters. You can set the birth rate and the death rate both to 0.03. Keep beta=0.5 and gamma=0.3 as before.

OpenSIR<-function(t, state, parameters) {  
 with(as.list(c(state, parameters)),{  
 N = S + I + R  
   
 #SIR w/ demography equations from lecture  
 dS <- -beta\*S\*I/N + birth\*N - death\*S + omega\*R  
 dI <- beta\*S\*I/N - death\*I - gamma\*I  
 dR <- gamma\*I - death\*R - omega\*R  
   
 # return the rates of change as a list  
 list(c(dS, dI, dR))  
 })  
}  
  
parameters <- c(beta = 0.5, #effective contact rate (aka transmission rate)  
 gamma = 0.3, #recovery rate (1/duration infection)  
 birth = 0.03, #birth rate (per capita)  
 death = 0.03, #all-cause mortality rate  
 omega = 0.0  
)

In addition to births and deaths, we’ve added another new parameter - omega. We’ll set omega to 0 for now, but what does omega represent?  
  
*Answer: Omega represents the rate of immunological waning. When omega is 0, there is no waning (immunity is lifelong).*

10. Calculate R0 and the compartment sizes at the endemic equilibrium. Do we expect the system to reach the endemic equilibrium or the disease-free equilibrium, and why? Did R0 increase or decrease compared to the BasicSIR model (without demography - part 6)? What is your intuition behind this?

R0 <- parameters["beta"]/(parameters["gamma"] + parameters["death"])  
print(R0)

## beta   
## 1.515152

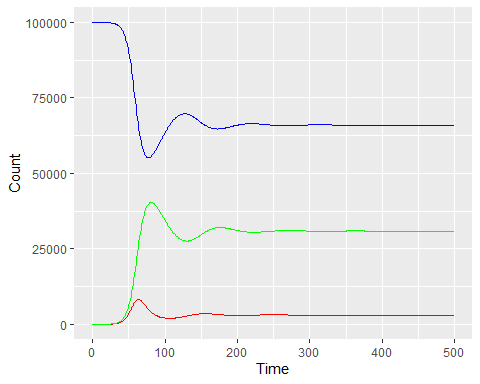
S\_star <- 1/R0  
I\_star <- (parameters["death"]/parameters["beta"])\*(R0-1)  
R\_star <- 1-(S\_star + I\_star)  
print(c(S\_star, I\_star, R\_star)) #prevalences are equilibrium

## beta death beta   
## 0.66000000 0.03090909 0.30909091

*Answer: R0 is 1.52, which is slightly lower than the R0 in question 6 (1.67). This is because some infected individuals now die before recovering, so the average duration of infectiousness is shorter, resulting in fewer susceptibles being infected per infected person. An endemic equilibrium is expected because R0 > 1 and the population of susceptibles is replenished via births (as opposed to the BasicSIR model without demography, where the epidemic eventually died out). At the endemic equilibrium, population prevalence will be 3.1% (and 66% of the population is susceptible, while 30.9% are immune).*

11. Plot the epidemic curve of this new model and examine the modeled equilibrium compartment sizes. Compare this against your prediction.

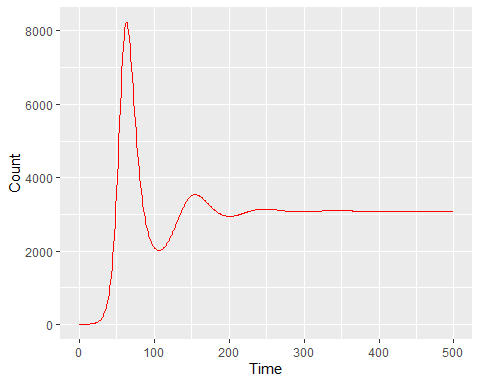
output <- ode(y = state, times = times, func = OpenSIR, parms = parameters)  
show\_SIR\_model\_results(output)



output\_df <- data.frame(output)  
output\_df$N <- output\_df$S + output\_df$I + output\_df$R  
print(output\_df$I[T\_end+1]/output\_df$N[T\_end+1])

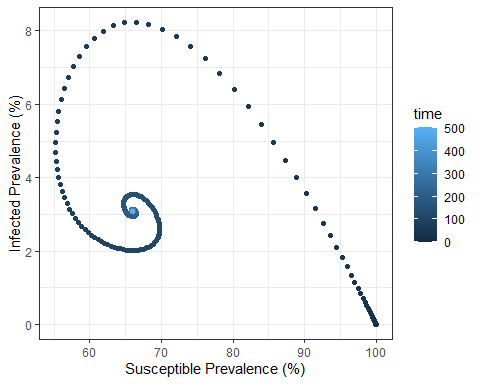
## [1] 0.03090826

#to "zoom in" on the I compartment (to observe oscillations)  
show\_SIR\_I\_output <- function(output) {  
 df1 <- data.frame(output)  
 ggplot() +   
 geom\_line(data = df1, aes(x = time, y = I), color="red") +  
 xlab('Time') +  
 ylab('Count')   
}  
  
show\_SIR\_I\_output (output)

  
*Answer: As predicted, an endemic equilibrium is reached, with prevalence = 3.1%, which matches almost exactly the calculated I\_star in question 10.*

12. We can also plot the phase diagram (prevalence of S against prevalence of I) to observe how the system reaches equilibrium (as we showed in lecture 3). We define an additional graphing function for this - phase\_diagram().

phase\_diagram <-function(output) {  
 #convert model output to a data frame (type of object in R) and reshape it so that each row represents a time-compartment combination (e.g. how many ppl in S compartment at time 10) - the latter just makes it easier to plot your results  
 df1 <- data.frame(output)  
 df1$N <- df1$S + df1$I + df1$R  
 df1[,2:ncol(df1)] <- df1[,2:ncol(df1)]/df1$N #divide by pop size to calculate pop prevalence  
 ggplot() +   
 geom\_point(data = df1, aes(x = S\*100, y = I\*100, color=time)) + #plot S\_prev against I\_prev  
 xlab('Susceptible Prevalence (%)') + ylab('Infected Prevalence (%)') + #label x and y-axes  
 theme\_bw() #this isn't necessary but makes the graph look a bit nicer  
}   
phase\_diagram(output)



13. Calculate the approximate average age of infection for this model and the approximate period of oscillation. Does the estimated period of oscillation look similar to the oscillation in the graph from question 11? Assume that time steps (and rates) are in months.

A <- 1/(parameters["death"]\*(R0-1))  
print(A)

## death   
## 64.70588

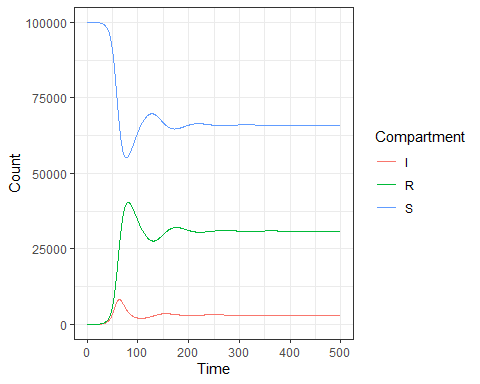
G <- 1/(parameters["death"]+parameters["gamma"]) #duration of infectiousness  
T <- 2\*pi\*sqrt(A\*G)  
print(T)

## death   
## 87.98219

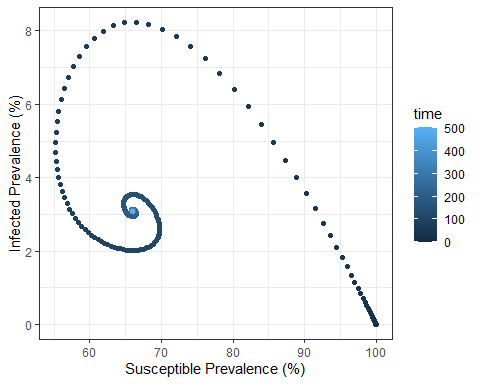
*Answer: The average age of infection is around 65 months, or a little under 5 and a half years. The period of oscillation is around 88 months (7.3 years). We can see that the distance between peaks in the graph (for part 11) is around 88 time steps (note that it may not be exact - the equation is an approximation).*

14. One at a time (holding all other parameters fixed), make the following changes:  
a.) change beta to 2.0 (keep gamma at 0.3, births and deaths at 0.03)  
b.) change gamma to 0.2 (keep beta at 0.5, births and deaths at 0.03)  
c.) change births and deaths (both) to 0.1 (keep beta at 0.5 and gamma at 0.3).  
Plot the resulting epidemic curves and phase diagrams. How do these changes affect the infection dynamics? Why?

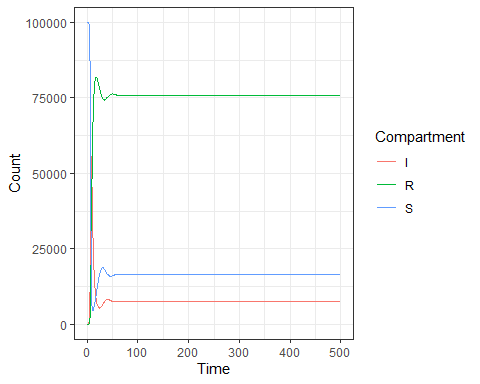
#plotting function that is technically more complex but can be applied to all model output (not just SIR) - you could also just use show\_SIR\_model\_results() for this as before.  
show\_model\_results<-function(output) {  
 #convert model output to a data frame (type of object in R) and reshape it so that each row represents a time-compartment combination (e.g. how many ppl in S compartment at time 10) - the latter just makes it easier to plot your results  
 df1 <- data.frame(output)  
 df2 <- reshape(df1, varying=colnames(df1)[2:ncol(df1)],   
 v.name="sizes", timevar="Compartment",  
 times=colnames(df1)[2:ncol(df1)], idvar="time", direction="long")  
 ggplot() +   
 geom\_line(data = df2, aes(x = time, y = sizes, color=Compartment)) + #plot the number of ppl in each compartment over time  
 xlab('Time') + ylab('Count') + #label x and y-axes  
 theme\_bw() #this isn't necessary but makes the graph look a bit nicer  
}   
  
#original (for comparison)  
parameters <- c(beta = 0.5, #effective contact rate (aka transmission rate)  
 gamma = 0.3, #recovery rate (1/duration infection)  
 birth = 0.03, #birth rate (per capita)  
 death = 0.03, #all-cause mortality rate  
 omega = 0.0  
)  
output <- ode(y = state, times = times, func = OpenSIR, parms = parameters)  
show\_model\_results(output)



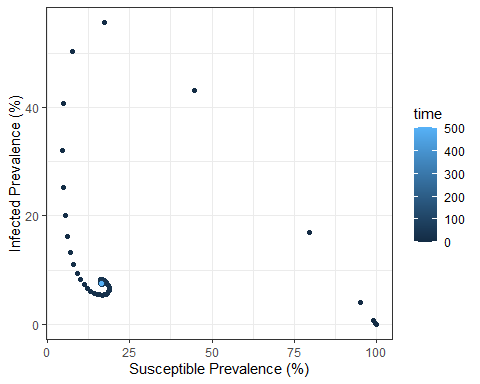
phase\_diagram(output)



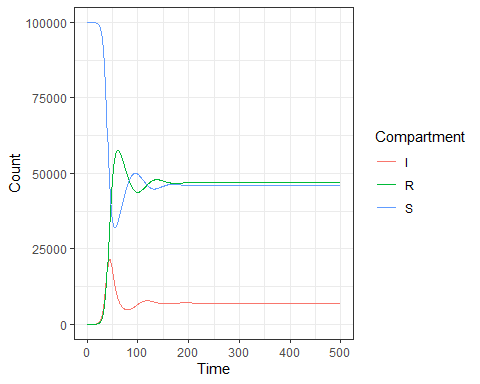
parameters["beta"] <- 2.0  
output <- ode(y = state, times = times, func = OpenSIR, parms = parameters)  
show\_model\_results(output)



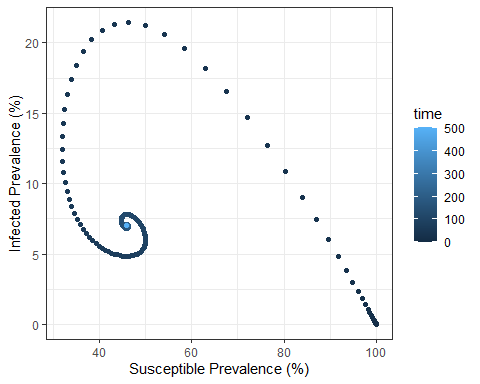
phase\_diagram(output)



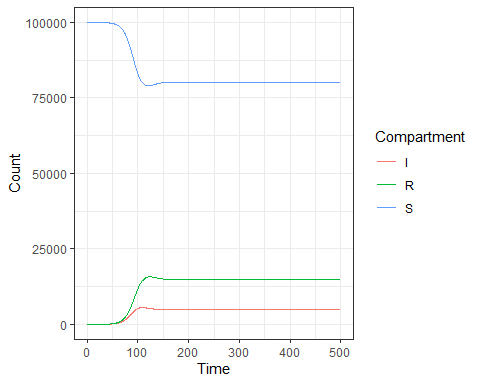
parameters["beta"] <- 0.5  
parameters["gamma"] <- 0.2  
output <- ode(y = state, times = times, func = OpenSIR, parms = parameters)  
show\_model\_results(output)



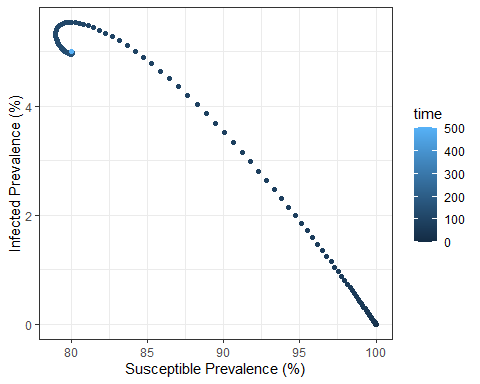
phase\_diagram(output)



parameters["gamma"] <- 0.3  
parameters["birth"] <- 0.1  
parameters["death"] <- 0.1  
output <- ode(y = state, times = times, func = OpenSIR, parms = parameters)  
show\_model\_results(output)



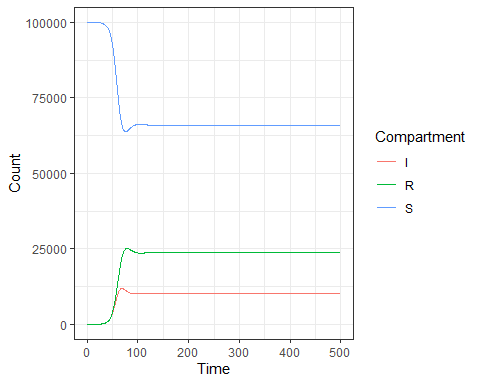
phase\_diagram(output)

 *Answers:*  
*a.) Increasing beta, the effective contact rate, increases the transmissibility, increasing the R0 and the endemic equilibrium prevalence, and allowing the system to reach equilibrium sooner (dynamics are faster)*  
*b.) Decreasing gamma increases the average duration of infection, allowing each infected person to infect more people. This increases the R0 and the endemic equilibrium prevalence.*  
*c.) Increasing the birth and death rate decreases the R0, because R0 = beta/(gamma + death\_rate) - average duration of infectiousness is shorter. However, this also increases the replenishing of susceptibles, which increases the effective reproductive number. In this case, the increased number of susceptibles outweighs the decreased duration of infectiousness, and the endemic equilibrium prevalence increases.*

15. Let’s revisit the omega parameter we added to OpenSIR. This parameter represents waning immunity (recall - when omega=0, immunity is lifelong). What type of model do we have as omega approaches infinity?    
  
*Answer: As omega approaches infinity, the duration of immunity is effectively 0, which corresponds to an SIS model.*

16. Set omega to equal 0.1. Run the model and plot the resulting epidemic curve. What do you notice? Has the R0 changed (from part 10)? If not, how do you explain the difference in output? (hint: think closely about the definition of R0)

parameters <- c(beta = 0.5, #effective contact rate (aka transmission rate)  
 gamma = 0.3, #recovery rate (1/duration infection)  
 birth = 0.03, #birth rate (per capita)  
 death = 0.03, #all-cause mortality rate  
 omega = 0.1  
)  
output <- ode(y = state, times = times, func = OpenSIR, parms = parameters)  
show\_model\_results(output)



output\_df <- data.frame(output)  
output\_df$N <- output\_df$S + output\_df$I + output\_df$R  
print(output\_df$I[T\_end+1]/output\_df$N[T\_end+1])

## [1] 0.1027907

*Answer: As expected, there are fewer recovered/immune individuals and more susceptibles, because there is now a flow from R back to S. The equilibrium prevalence is higher, despite the basic reproductive number staying the same (recall - the equation for the basic reproductive number is still beta/(gamma + death\_rate) - it is independent of omega). This is because the effective reproductive number has increased, due to the higher number of susceptibles. Remember that the basic reproductive number is defined as the average number of infectious individuals generated by a single infectious individual in a fully susceptible population.*