SIR Model Using Vaccinations

Yasin Akgun, Eli Sanderson, Nathaniel Schmuch

***Abstract*—In this paper we take the standard SIR model one step further and introduce a vaccination rate to measure the effects it has on the model.**

I. Introduction

Throughout class we have looked at examples of SIR models which all address the spread of a disease within a contained population given some disease length. Our work aimed to introduce a vaccination rate and study it's affect on the intensity and length of the epidemic. We introduced a daily vaccination rate of 15% which would be applied to susceptible individuals and cause the patient to become immune after three days.

II. SIR Model & Assumptions

As stated before our work revolved around the use of the SIR model. This model makes use of a series of difference equations in order to calculate the population of susceptible, infected, and recovered at various time steps during the epidemic. Below you will find the difference equations used in order to track these populations given the vaccination rate.

$$S_{n+1} = S_n - aS_nI_n \quad (1)$$
$$I_{n+1} = I_n - aS_nI_n - rI_n \quad (2)$$
$$R_{n+1} = R_n + rI_n \quad (3)$$

In the equations above, S_n is the number of susceptible people at time n, I_n is the number of infected people at time n, and R_n is the number of recovered people at time n. Furthermore r is the removal rate and a is the transmission rate. These constants r, a, and the initial conditions for S, I, and R can be calculated using the following relationships:

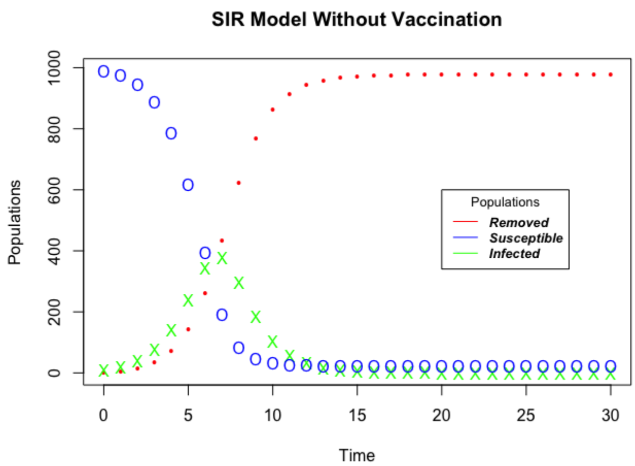
$$r = \frac{1}{\textit{length of disease}}$$
$$a = \frac{I_1 + I_0(r-1)}{I_0S_0}$$
$$S_0 = \textit{total population} - \textit{number of infected}$$
$$I_0 = \textit{number of infected}$$
$$R_0 = \textit{number of immune}$$

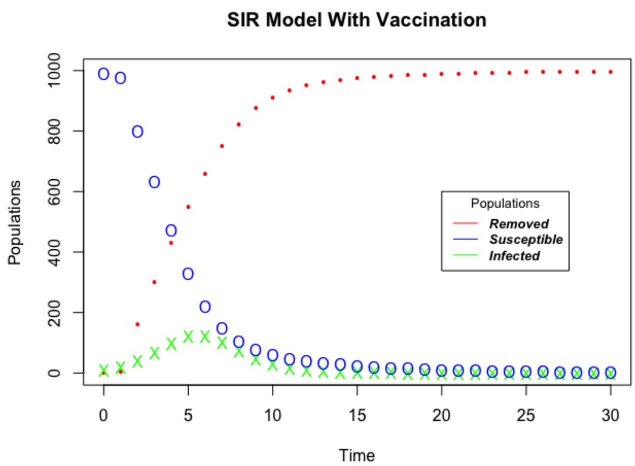
In order to adjust the model to accompany vaccinations, we introduced two more variables: v, the vaccination rate, and λ, a discriminatory term which depends on the first day of vaccinations. Moreover, we added a term to the equations for S_n and R_n which takes a fraction of susceptible people who have been vaccinated and adds them to the number of recovered people if the day n is a vaccination day. We used an if-else statement to determine the starting day of vaccinations, as shown in the appendix. The modifies equations for S_n, I_n, and R_n are shown below:

$$S_{n+1} = S_n - aS_nI_n - vS_n\lambda \quad (4)$$
$$I_{n+1} = I_n - aS_nI_n - rI_n \quad (5)$$
$$R_{n+1} = R_n + rI_n + vS_n\lambda \quad (6)$$

III. Results & Analysis

In order to measure the impact of a vaccination rate, we ran two versions of our SIR model, one with the vaccination rate and one without. Each version modeled a closed population of 1000 people 10 of which were initially infected. The length of our diseases was two days, and we ran the simulation for a 30 day period. For the vaccinated simulation we included a 15% vaccination rate, which would cause immunity after three days. The following figures show the relationship between the susceptible, infected and recovered population for the given simulations.





Our simulations show that the inclusion of a vaccination rate can drastically reduce the severity of the epidemic. Our green line in both figures represents the infected population. As you can see in the SIR model without vaccination we see a large peak around day seven. This peak represents the maximum number of people who got infected which in this case is around 376. When comparing it to the vaccination model not only do we see a much smaller peak at 120 people, but this peak occurs at around day five. This implies that the length of the overall epidemic is shorter by approximately two days. In other words the number of infected people falls below one about two days earlier in the vaccination model than in the model without vaccinations.

IV. Conclusion & Future Work

The code created to model this problem is flexible and is able to show results for many similar systems due to it being a function with many different inputs. What can be found through changing the inputs is a give and take over the other results. This being if the Infected population starts higher as time progresses it will take more time for everyone to recover and be removed. One thing included in the model that changed things severely is the time between the start of the disease and the start of the Vaccinations. This changes the model so the traditional SIR model can't be used. Due to this before the vaccinations begin the number of infected people grow and therefore the susceptible decreases. This changes the initial conditions to start the traditional SIR model with based on the amount of time between the start of the disease and the start of the vaccinations.

What can be worked on in the future is reducing the number of assumptions made. This would include removing immigration and emigration as well as making the function able to include a changing vaccination rate and infection rate. This would change the model further to make it more accurate and therefore making giving the user a better understanding of what will happen in the future.

V. Appendix: R Code

The following is the R-code used to to create our vaccinated SIR model. We aimed to create an application which may used given multiple different initial conditions.

```r
# Model with vccination rate
SIR = function(disLength,pop,initInfect,aftInfect,vaccRate,vaccStartDay,simDays) {
  R = vector() #vector removed
  I = vector() #vector infected
  S = vector() #vector suseptible

  t = seq(0,simDays) #time seq
  R[1] = 0
  I[1] = initInfect
  S[1] = pop - initInfect
  r = 1/disLength #removal rate
  a = (aftInfect - initInfect + r*initInfect)/(initInfect*S[1]) #transmission rate

  cat("Model with vaccination", "\n")
  cat("Day","\t","R","\t","I","\t","S","\n")
  cat(1,"\t",R[1],"\t",I[1],"\t",S[1],"\n")
  for (i in 2:length(t)) {
    if (i - vaccStartDay < 0) {D = 0}
    else {D = 1}
    #determines when vaccinations occur
    R[i] = R[i-1] + r*I[i-1] + vaccRate*S[i-1]*D
    I[i] = I[i-1] + a*S[i-1]*I[i-1] - r*I[i-1]
    S[i] = S[i-1] - a*S[i-1]*I[i-1] - vaccRate*S[i-1]*D
    cat(t[i],"\t",R[i],"\t",I[i],"\t",S[i],"\n")}

  matplot(xlab="Time",ylab="Population(s)",cbind(R,I,S),col=c("red","green","blue"),
          pch=c("•","X","O"))
  legend(20, 600, legend=c("Removed", "Susceptible", "Infected"),
         col=c("red", "blue", "green"), lty=1, cex=0.8,
         title="Populations", text.font=4)}

SIR(2,1000,10,20,0.15,3,30)
```

Our SIR function takes in a number of constants from the user, and runs them through the difference equations described in section II. As the function runs through the loop it calculates values for each time step and stores them within vectors for easy manipulation afterwards. In order to properly monitor the vaccination rate we included the D variable which behaves as a check to see if the appropriate amount of time has passed for the vaccination to become effective in which case the vaccination rate term is included in the calculation.

References

[1] Shiflet, Angela B., and George W. Shiflet. Introduction to Computational Science: Modeling and Simulation for the Sciences. Princeton University Press, 2014.