5-4-2022

SIR Model and COVID-19

Molly Johanson
College of Saint Benedict/Saint John's University, mjohanson001@csbsju.edu

Clara Noack
College of Saint Benedict/Saint John's University, cnoack001@csbsju.edu

Dacoda Speidel
College of Saint Benedict/Saint John's University, dspeidel001@csbsju.edu

Molly Johanson, Clara Noack, Dacoda Speidel

Math Capstone Project

Professor Sunil Chetty

5/4/2022

SIR Model and COVID-19

In March 2020, severe acute respiratory syndrome coronavirus, or COVID-19, emerged and drastically changed our world. This new disease called for a different outlook on how to best minimize the number of individuals infected and even worse, those that were dying. It was found that any contact over fifteen minutes with an infected individual made you susceptible to becoming infected yourself. Thus, it was becoming clear that the world needed some way to model the spread of COVID-19. This can be done using an area of mathematics called mathematical modeling.

Infectious diseases have been around for hundreds of years, and a way to track these diseases has been through mathematical modeling. Daniel Bernoulli was one of the first mathematicians to record a formula or method for modeling smallpox:

$$-\,dI_x^s = PI_x^s\,dx + \left[\left(-I_x^s\frac{dI_x}{Ix}\right) - P\pi I_x^s\left(\frac{I_x^{\bar{s}}}{Ix}\right)dx\right] \tag{1}$$

Where the left-hand side demonstrates a decrease in the group of the population that has never had smallpox, the first term on the right-hand side is the group who has contracted smallpox and the second term is the group that has died from both natural or other causes and smallpox[1].

Bernoulli's model was utilized and expanded on by Ronald Ross and William Hamer in the early 1900s[2]. Their model focused on a population that can easily move from one state to another in terms of being susceptible, infected, or recovered from a disease, or more easily seen in Figure 1. Ross and Hamer's findings gave rise to what is known as the Susceptible-Infected-Recovered or SIR model. This model has been used to model everything from malaria to influenza, and most recently COVID-19. The SIR model is a set of ordinary differential equations that can be used to show the changes seen in a population that encounters an infectious disease. The model's simplest version includes solely the differential equations for the susceptible, infected, and recovered populations, such that[3]

$$\frac{dS(t)}{dt} = \frac{-\beta S(t)I(t)}{N} \tag{2}$$

$$\frac{dI(t)}{dt} = \frac{\beta S(t)I(t) - \gamma I(t)}{N} \tag{3}$$

$$\frac{dR(t)}{dt} = \frac{\gamma I(t)}{N} \tag{4}$$

In equation two (2), the number of individuals that are susceptible decreases at a constant rate, given by β, proportional to the number of susceptible individuals times the number of infected individuals. From the interactions between the susceptible individuals and the infected individuals, all the susceptible individuals that became infected move into the infected category. In equation three (3), the first term shows that the number of individuals that are infected increases at the same rate as the number of individuals that are susceptible decreases. The second term in this equation shows a decrease in the number of infected individuals, proportional to the number of infected individuals, due to people recovering and moving into the recovered, or removed, category. Finally, in equation four (4), the number of individuals that have recovered is proportional to the number of infected individuals with a given constant rate of recovery, γ.[4]

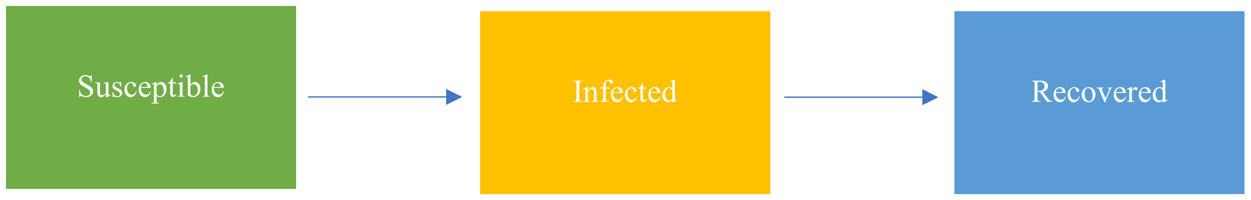


Figure 1. displays the pathway an individual can take from susceptible to infected to recovered in terms of contracting an illness. This pathway assumes once an individual has reached the Recovered state, they cannot go back into the Susceptible state.

To use the simple SIR model, gamma and beta must be given, or the infection and recovery rate respectively. The number of individuals, or percentage of individuals of the total population, in each group at a given time, t, must also be given. Once all those values are known, one inputs each value in the correct place and solves the system. The outcome will indicate how the size of each group is changing either by the percentage of the total population at a given time t. More simply, if one were to look at a graph of the data and connect all points for each group together to form a line for each state, the calculated number would be the slope of each line at that specific point in time, t. For example, consider a population that encounters a disease that has an infection rate of .55 and a recovery rate of .15. The total population is 100, and 75 individuals are in the susceptible group, 20 are infected, and 5 are recovered. Then, plugging the information in its correct spot in equations 2 through 4 and simplifying:

$$\frac{dS(t)}{dt} = \frac{-\beta S(t)I(t)}{N} = \frac{-(.55)(75)(2)}{100} = \frac{-82.5}{100} = -.825$$

$$\frac{dI(t)}{dt} = \frac{\beta S(t)I(t) - \gamma I(t)}{N} = \frac{82.5 - (.15)(20)}{100} = \frac{79.5}{100} = .795$$

$$\frac{dR(t)}{dt} = \frac{\gamma I(t)}{N} = \frac{(.15)(20)}{100} = \frac{3}{100} = .03$$

Therefore, the susceptible group is decreasing in size by 82.5%, the infected group is increasing in size by 79.5%, and the recovered group is increasing in size by 3% at time t.

This model assumes that once an individual has moved into the recovered, they are unable to move back into the susceptible model. It also assumes there is no movement in and out of the population, the outbreak is short lived, and that the disease has no latent period[3]. Obviously, few of these assumptions are generalizable to the COVID-19 virus. These equations can be utilized to look at the spread of a disease through continuous time assumptions. A continuous dynamical system, such as the one described previously, assumes that the state evolves continuously throughout time according to a fixed set of rules. A dynamical system is a system or process in which motion is occurring and can be explained using one or more equations, whereas a static system is a system or process in which no motion or change is occurring.

However, COVID-19 is considered a discrete dynamical system because its set of rules changes from time t to time t+1. The discrete system calls for time to be measured in discrete units, such as hours, days, or weeks, as opposed to continuously. Therefore, it is important to use a discrete dynamical system to model a disease such as COVID-19. The equations from the continuous dynamical system can be modified to satisfy the discrete dynamical system requirements as seen in equations five through seven. The discrete system of differential equations which can be used to model COVID-19 are as follows[3]:

$$S(t+1) = S(t) - \frac{\beta S(t) I(t)}{N} \tag{5}$$

$$I(t+1) = I(t) + \frac{\beta S(t)I(t)}{N} - \gamma I(t) \quad (6)$$

$$R(t+1) = R(t) + \gamma I(t) \quad (7)$$

Where N is the total population, and S(t), I(t), and R(t) are the previous populations of the three respective groups and S(t+1), I(t+1), and R(t+1) are the new populations in each group. From here on, the discrete dynamical system will be used to model the spread of COVID-19.

Following the outbreak of COVID-19, studies revolving around how the SIR model could be utilized to model the disease became prevalent. These studies provide an expanded SIR model to include exposed (E), quarantined (Q), asymptomatic infected (I_A), and symptomatic infected individuals (I_S)[5]. A particular study created differential equations which included these new groups, as well as the original susceptible and recovered groups. The study pictured the spread of COVID-19 in a Markov chain way of thought, which brought about the introduction of new parameters for new movements of individuals between the various groups. The parameters were assigned as follows[5]:

Parameter	Definition	Value
τ	Transfer rate from susceptible to quarantine	0.002
β	Contact rate between susceptible and exposed	0.0805
δ	Mortality rate due to COVID-19 symptomatic individuals	1.6728×10^{-5}
γ	Transfer rate from exposed to quarantine	2.0138×10^{-4}

η	Transfer rate from exposed to symptomatic	0.4478
θ	Transfer rate from quarantine to asymptomatic	0.0101
μ	Natural death rate	0.0106
ν	Transfer rate from quarantine to symptomatic	3.2084×10^{-4}
σ	Transfer rate from exposed to asymptomatic	0.0668
A	Natural death rate	0.02537
R_1	Recovery rate of asymptomatic	5.7341×10^{-5}
R_2	Recovery rate of symptomatic	1.6728×10^{-5}

These parameters were then utilized in the following differential equations, where it is assumed that populations of each group add up to the total population[5]:

$$\frac{dS(t)}{dt} = A - (\tau + \mu)S(t) - \beta S(t)E(t) \tag{8}$$

$$\frac{dE(t)}{dt} = \beta S(t)E(t) - (\gamma + \mu + \eta + \sigma)E(t) \tag{9}$$

$$\frac{dQ(t)}{dt} = \tau S(t) + \gamma E(t) - (\mu + \nu + \theta)Q(t) \tag{10}$$

$$\frac{dI_A(t)}{dt} = \sigma E(t) + \theta Q(t) - (\mu + R_1)I_A(t) \tag{11}$$

$$\frac{dI_s(t)}{dt} = \eta E(t) + \nu Q(t) - (\delta + \mu + R_2)I_S(t) \tag{12}$$

$$\frac{dR(t)}{dt} = R_1 I_A(t) + R_2 I_S(t) - \mu R(t) \tag{13}$$

These equations can then be put into Microsoft Excel along with the parameter values to get a distinctive understanding of how the numbers of individuals in each group are changing from time t to time t+1. Because the system of equations eight through thirteen changes in increments of time, this would be considered a discrete dynamical system. The values of the equations can be found at time t and will be slightly different than those found at time t+1.

Microsoft Excel was utilized to track the changes from time t to time t+1. It is important to note that the equations were written to measure what percentage of the population is susceptible, infected, quarantined, exposed, and recovered. Percentages were used because it remains the ambiguity of having a portion of an individual being in a group. For example, the Excel equations will not say that 2.4 individuals are susceptible, it will just say that .2% of the population is susceptible. Using the percentage of a population versus the actual population size allows for the elimination of having to divide each term by the total population, as is needed for the discrete dynamical system. For the Excel equations, Minnesota data was utilized, along with the parameter values obtained from previous research. The only Minnesota data used in the Excel file was the total population, which is 5.64 million as of 2019. It was assumed that at the beginning there was only one exposed individual, and all others were in the susceptible population. Utilizing that one assumption, the results from the Excel data are seen in Figure 2.

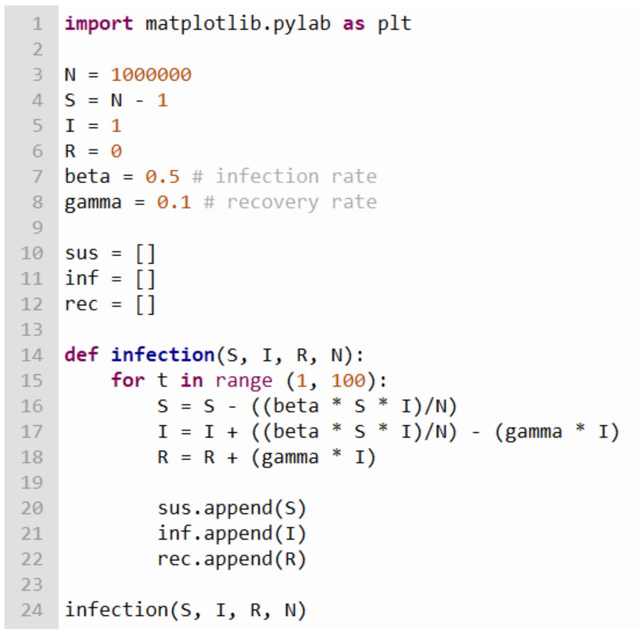
time (days)	Susceptible	Exposed	Quarantined	Infected (asymptomatic)	Infected (symptomatic)	Recovered
0	1	1.773E-07	0	0	0	0
1	0.974429986	9.842E-08	0.0002	1.1844E-08	7.93972E-08	0
2	0.948865092	5.443E-08	0.00039068	2.03829E-06	1.86794E-07	2.007E-12
3	0.923305315	2.999E-08	0.00057224	5.96609E-06	3.34529E-07	1.22E-10
4	0.897750652	1.646E-08	0.00074487	1.16842E-05	5.27999E-07	4.684E-10
5	0.8722011	9.003E-09	0.00090877	1.9084E-05	7.68742E-07	1.142E-09
6	0.846656659	4.905E-09	0.0010641	2.80597E-05	1.05617E-06	2.237E-09
7	0.821117328	2.662E-09	0.00121107	3.85085E-05	1.38854E-06	3.84E-09
8	0.795583104	1.439E-09	0.00134983	5.033E-05	1.76353E-06	6.031E-09

Figure 2. displays results from Excel with an assumed one exposed individual and utilizing the equations and parameters from previous research.

Compiling the data into an Excel spreadsheet provides the ability to look at how individuals are moving to the various groups from one time to another. In this case, how individuals are moving from one group to another from day to day. This is beneficial in seeing how COVID-19 can spread relatively quickly without the right precautions being put into place. However, this model does not allow an individual to move from the recovered population back into the susceptible population, which can occur after contracting COVID-19. This causes the population to start accumulating in the recovered state, eventually causing the disease to die off and lose its ability to spread. Again, this is not seen with COVID-19 because an individual can get reinfected. One downside with this study, however, was the lack of consideration regarding vaccination status. Further, one other downside was that the excel spreadsheet did not allow us to easily add in new conditions. Adding anything new would cause the spreadsheet to become increasingly messy and hard to follow. Despite Excel allowing for a day-to-day picture of what was occurring, there were many downsides which need to be accounted for. A new way of modeling the spread of COVID-19 had to be created to model the spread most effectively and cleanly.

Python allows for new conditions to be easily added in and does not generate a large list of numbers. It also gives an easy to read and clear graph of how COVID-19 is spreading. A simple version of the SIR model was coded first, which just included the Susceptible, Infected, and Recovered states. The beginning states as shown in Lines 3-8 include the total population (N), the initial number of susceptible individuals (S), the initial number of infected individuals (I), the initial number of recovered individuals (R), infection rate (β), and recovery rate (γ). These were all assumed values. The Python code runs through the equations from time 0 to time 100 days. To accomplish this, a "for" loop was incorporated into the code to repeat any set of coded instructions over and over, until a given condition is met. We wanted to allow our "for" loop to continue circulating within a time range, specifically from time 0 to 100 days, as shown in Line 15 in Figure 3. Within the "for" loop, the code includes the Susceptible, Infected, and Recovered equations from equations 5, 6, and 7, as seen in Figure 3. Within the "for" loop, the code appends the S, I, and R values into their respective lists in order to keep track of the total number of people in each state at a given time. The Python code generated a graph depicting how the population sizes changed from time 1 to time 100 as seen in Figure 4.

As before with our Excel model, this Python code does not factor in the ability for individuals to move back into the susceptible population after their immunity period is over. In this code, once an individual has reached the Recovered state, they do not move from that state. It also does not account for the possibility of an individual being less susceptible to COVID-19 due to their vaccination status.



```python
import matplotlib.pylab as plt

N = 1000000
S = N - 1
I = 1
R = 0
beta = 0.5 # infection rate
gamma = 0.1 # recovery rate

sus = []
inf = []
rec = []

def infection(S, I, R, N):
    for t in range (1, 100):
        S = S - ((beta * S * I)/N)
        I = I + ((beta * S * I)/N) - (gamma * I)
        R = R + (gamma * I)

        sus.append(S)
        inf.append(I)
        rec.append(R)

infection(S, I, R, N)
```

Figure 3. presents basic SIR Python code with an assumed population of one million and an infection rate of 0.5 and a recovery rate of 0.1.

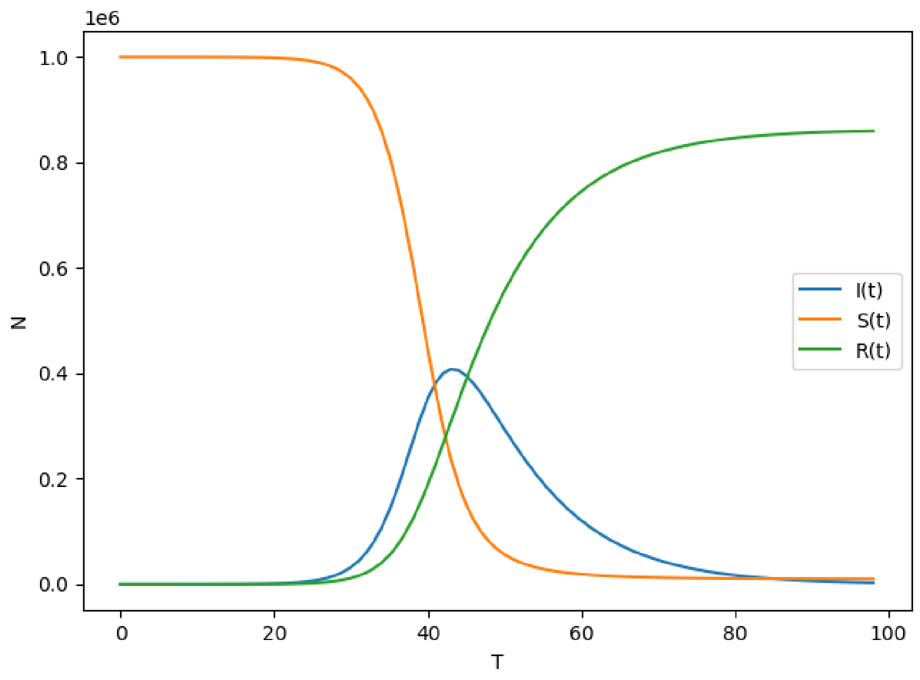
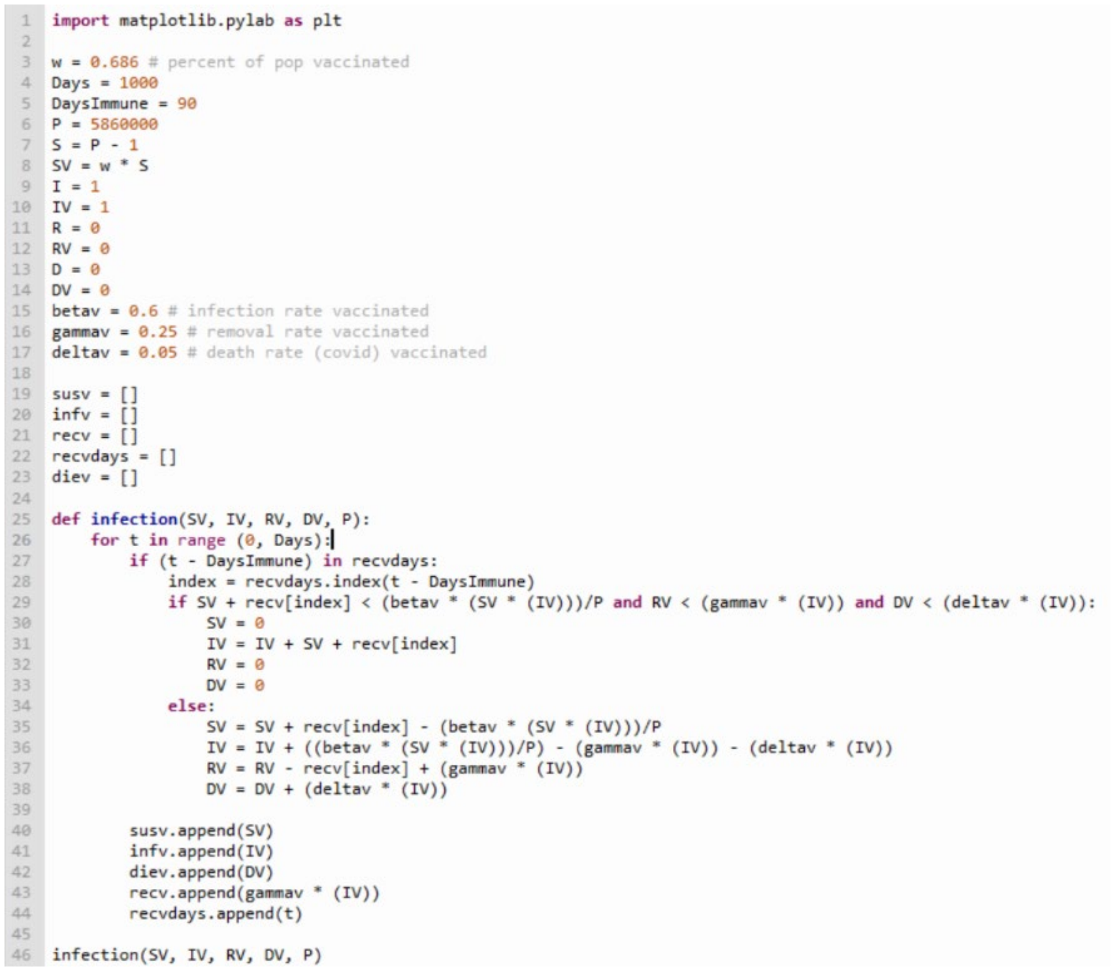


Figure 4. depicts the result from Python code given in Figure 3. Time is measured in days and the population is measured as the percentage of the total population of one million.

Following the creation of a simple SIR model using Python, a more complex version was created, and included vaccination status. In this revised version, each population from the original model was split up into a vaccinated and unvaccinated population. New parameters were also created to explain the spread from one group to another group, because there are new transfer rates between the groups following the addition of these new parameters. This new Python model also accounts for when an individual loses their immunity from COVID-19 following the end of their immunity period. It has been seen that about 90% of individuals can maintain immunity up to eight months following their infection with COVID-19[6]. However, this study decided to make an immunity of 90 days, because it is known that almost 100% of individuals will have immunity through that period. Following those 90 days of immunity, the code puts an individual who has reached the end of their immunity period back into the susceptible category. If the individual was originally unvaccinated, they will be placed into the susceptible unvaccinated group, and vaccinated individuals will go back into the susceptible vaccinated group. A core goal of this study was to find a way to model COVID-19 to the best extent and finding a way to bring individuals back into the susceptible groups was incredibly important. Python code for this new method of modeling COVID-19 is seen in Figure 5. The parameters for the transfer rates between groups are as follows, and the original transfer rates between groups were assumed:

Parameter	Definition	Value
βV	Transfer rate from susceptible vaccinated to infected vaccinated	0.6
βN	Transfer rate from susceptible unvaccinated to infected unvaccinated	0.9
γV	Transfer rate from infected vaccinated to recovered vaccinated	0.25
γN	Transfer rate from infected unvaccinated to recovered unvaccinated	0.15
δV	Death rate from COVID-19 of vaccinated individuals	0.05
δN	Death rate from COVID-19 of unvaccinated individuals	0.1

This study began by creating two separate codes, one for vaccinated and one for unvaccinated. This allowed for the ability to check the codes success separately, before combining the vaccinated and unvaccinated groups to get the full picture of the spread of COVID-19 with the new parameters. The vaccinated code can be found in Figure 5. In this code, a few assumptions are made about the starting population, and from there, the code runs according to the equations provided. The Minnesota population[7] and the Minnesota vaccination rates at the beginning of April 2022[8] were used for starting assumptions. From there, it was assumed that the population started with one infected individual, and the immunity period was set at 90 days due to previously stated information. Finally, the βV, γV, and δV were set to 0.6, 0.25, and 0.05 respectively. These values were also assumed, but considerations about each value were taken into account. For the infection rate, in order for a disease to be infections, it must be able to spread relatively easily from individual to individual. Therefore, a value above 0.5 is normally used to describe a disease that is easily spreadable. The death rate from COVID-19 in vaccinated individuals is relatively small, simply because they have been vaccinated and are better protected against the disease. Even though a vaccinated individual can contract COVID-19, the individual is less likely to get as sick or for as long as their unvaccinated counterpart. Finally, the removal rate was generalized based on removal rates found in other studies, and a removal rate that seemed plausible based off current data.



```python
import matplotlib.pylab as plt

w = 0.686 # percent of pop vaccinated
Days = 1000
DaysImmune = 90
P = 5860000
S = P - 1
SV = w * S
I = 1
IV = 1
R = 0
RV = 0
D = 0
DV = 0
betav = 0.6 # infection rate vaccinated
gammav = 0.25 # removal rate vaccinated
deltav = 0.05 # death rate (covid) vaccinated

susv = []
infv = []
recv = []
recvdays = []
diev = []

def infection(SV, IV, RV, DV, P):
    for t in range (0, Days):
        if (t - DaysImmune) in recvdays:
            index = recvdays.index(t - DaysImmune)
            if SV + recv[index] < (betav * (SV * (IV)))/P and RV < (gammav * (IV)) and DV < (deltav * (IV)):
                SV = 0
                IV = IV + SV + recv[index]
                RV = 0
                DV = 0
            else:
                SV = SV + recv[index] - (betav * (SV * (IV)))/P
                IV = IV + ((betav * (SV * (IV)))/P) - (gammav * (IV)) - (deltav * (IV))
                RV = RV - recv[index] + (gammav * (IV))
                DV = DV + (deltav * (IV))

        susv.append(SV)
        infv.append(IV)
        diev.append(DV)
        recv.append(gammav * (IV))
        recvdays.append(t)

infection(SV, IV, RV, DV, P)
```

Figure 5. displays the expanded code to describe how the vaccinated population is affected by COVID-19.

The Python code runs through the equations from time 0 to the time given in the original parameters. To accomplish this, a "for" loop was again incorporated into the code, like the simple SIR code in Figure 3. We wanted to allow our "for" loop to continue circulating within a time range, specifically from time 0 to a predetermined time "Days" as shown in Line 26 in Figure 5. The code circulated through a "for" loop from time 0 to 1000 days, allowing us to visualize the projection of the virus based on the initial parameters. Within the "for" loop, the code includes the Susceptible, Infected, and Recovered equations expanded from the equations in Figure 3 to account for the vaccinated population. In addition to these three equations, an equation to account for the Death state was included. All four equations can be seen in Lines 35-38 in Figure 5. In contrast to the simple SIR code, this code includes "if-else" statements. In an "if-else" statement, if a given condition stated in the "if" statement if true, then the program will execute that condition. On the other hand, if the given condition in the "if" statement is not true, then the program will execute the condition in the "else" statement. In this Python code, "if-else" statements starting at Line 29 were included to ensure the values for SV, IV, RV, and DV were not negative values at the end of each "for" loop iteration. This was necessary to include because it is not realistic to have a negative number of people in a given state. If the people entering a given state were going to be a negative value, then we wanted the code to essentially say that zero people entered that given state at time t.

Within the "for" loop, the code appends the SV, IV, RV, and DV values into their respective lists, similar to the simple SIR code in Figure 3. However, in this expanded model, we wanted to allow for those that have entered the Recovered state the ability to re-enter the Susceptible state after a given immunity period, and thus be able to become infected again. To accomplish this, we needed another list that kept track of the number of people that entered RV at each time t, represented by recvdays. A list was necessary because we needed to be able to know when a specific group of people entered the Recovered state at time t to ensure they stayed within that state for the entire duration of the immunity period. For this purpose, it would not have been appropriate to maintain a single sum of people within that state and have a set proportion of people leaving that state at any given time. In this case, there would have been no way to keep track of the time certain individuals stayed in the Recovered state, allowing for the possibility of some individuals to stay within this longer for a shorter or longer time than the given immunity period. With this, an “if” statement was included to essentially say, if a group of people that entered the entered the Recovered state at time t have finished their recovery period, then that group will re-enter the Susceptible state. One consequence of allowing people to become susceptible to a disease after recovering from it is the possibility of future outbreaks occurring, generating waves, as shown in Figure 6.

The figure produced from the expanded SIR code is only showing how the populations of each group are changing throughout time for vaccinated individuals. As seen in the graph, the susceptible individuals decrease greatly over a small amount of time, which is paired with an increase in the infected and recovered lines. Then, the trends for those three groups will switch. The susceptible individuals increase, while the populations in the infected and recovered states decrease. These swings demonstrate when the population is hit by a wave of COVID-19, which greatly impacts the populations in each group. Then, after a certain period a switch is seen and the trends reverse. This demonstrates how the infectiousness of that wave is decreasing because enough of the population has been infected. The increase in individuals in the susceptible population also indicates when individuals who have reached the end of their immunity period leave the recovered group and re-enter the Susceptible state. The higher the infection rate, the more spikes will be seen on the graph generated by the Python code. It is important to note that the death line does not undergo spikes like the other three lines. This occurs because once an individual enters the Death state, they cannot leave. Therefore, this line is additive over time.

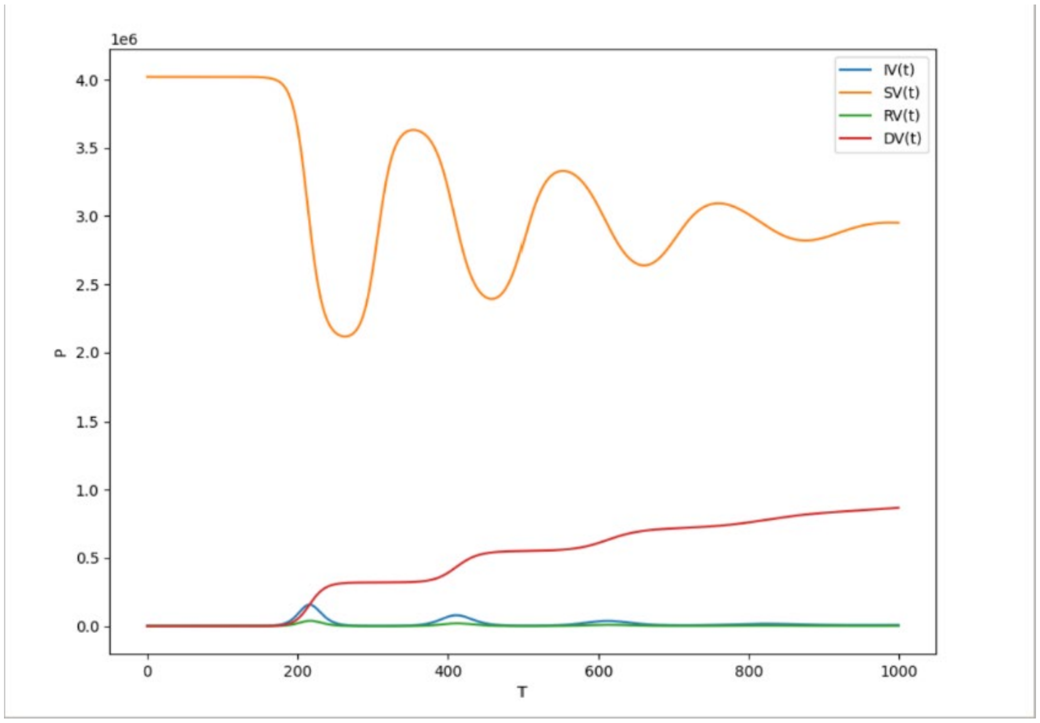


Figure 6. depicts the graph generated by the Python code displayed in Figure 5. Time is given in units of days, and the population is given in terms of number of individuals in each group.

A similar approach was taken with the unvaccinated individuals, however, issues with the code presented challenges. The same fluctuation should be seen with the unvaccinated individuals, just on a stronger scale, due to these individuals being more susceptible to COVID-19 and their likelihood to get sicker when they contract COVID-19. However, only one peak is seen, and after individuals re-enter the Susceptible state after their 90-day immunity period that population levels out. This means that those re-entered individuals are not getting sick or contracting COVID-19 again. However, this is simply not possible. Changes to the code must be made to correct this issue. Yet, due to time constraints, this study was not able to make those changes.

Python allows for the addition of variables and conditions to be much easier than it would be in Excel. Python provides the ability to tell the program when certain conditions will begin or end, such as with the immunity period of an individual ending 90 days after the individual contracted COVID-19. It is possible to generate a chart of the population sizes of each state for every time from time 0 to time 1000, but Python does not automatically make that for the user. This is beneficial, because it allows the user to overlook all the messy data and just look at the graph generated by the code. Despite Python and Excel doing similar things, Python does these calculations in a much cleaner way than Excel does. This is important, because the data sizes used when modeling COVID-19 get to be rather large, and the cleaner this can be done the better.

There are a few downsides to using Python. A primary issue is the limited ranges of β, γ, and δ values. If any respective value was too large or too small, then it resulted in skewed results. However, this would ultimately hold true in a real-world example, like COVID-19. For example, if an infection rate of a given strain of COVID-19 is very low, the virus is already limited in its capacity to rapidly spread, ultimately leading to minimal numbers of people becoming infected with that strain. That strain of the virus is not viable enough to produce the greatest impact on the human population and will in turn die off.

Another downside to using Python is that one needs to be well-versed in the language in order to add higher complexities to any given code. Even though this is more of an issue to the user, this does limit the programmer in various aspects, such as needing to generalize certain assumptions. For example, Minnesota’s state population is most likely changing every day, with people moving into or out of the state. However, we needed to assume our population stayed the same throughout the duration of our code, aside from the people that have died from the virus. Looking at COVID-19 in particular, researchers have learned that the virus is capable of changing in various ways, such as introducing new mutations or even the immunity period. Again, we had to keep our beginning assumptions the same throughout the duration of the code, mainly because we were unable to write these different factors into the code.

In addition, the lack of being well-versed in the code led to complications in writing out different aspects of the code. The part of the code that was the hardest for us to write was having people re-enter the Susceptible state after spending a certain amount of time in the Recovered state. We knew how we wanted this to happen, but it was difficult for us to write it out in the Python language. Being said, it is suggested that a user understands this language well if they were to take on this sort of project, or a project of similar complexity.

Areas of further research are vast, especially due to the fluidity of COVID-19, and its spread. One major area, if an individual were to pick up this research, would be to solve the issues with the unvaccinated code. It is likely that the issue(s) are relatively small, but for an individual not well versed in the Python language, the issue(s) would be hard to spot. Once the codes are fixed to be the most efficient they can be, it would be effective to combine the two codes. This allows for an understanding of how COVID-19 is spreading and how the conditions of the states are changing throughout time for vaccinated and unvaccinated individuals. By bringing light to these trends, it might bring about a new wave of individuals getting vaccinated, once they realize how much they can be benefitted by being vaccinated against COVID-19. Once the code has been fixed, it would also be possible to incorporate different mutations of COVID-19 into the code. All the mutations and strains of COVID-19 have slightly different infection, recovery, and death rates. These can be used in the code to show how those specific strains of COVID-19 can affect the vaccinated and unvaccinated individuals. Of course, it is important to start small with changes and new conditions, but once a simple code has been created it is easy to add onto that code. Then, one could look at mixing different mutations together and how that would affect the states of COVID-19 throughout time. A final area of further investigation would be to include actual data in the assumptions area of the Python code. If the code had been working, this study would have also incorporated the infection, recovery, and death rates of COVID-19 in Minnesota.

Works Cited:

1. https://www.sciencedirect.com/science/article/pii/S0025556402001220
2. https://mat.uab.cat/matmat_antiga/PDFv2013/v2013n03.pdf
3. https://www.lagrange.edu/academics/undergraduate/undergraduate-research/citations/19-Citations_2021_Aly_Lord.pdf
4. https://users.math.msu.edu/users/gnagy/teaching/mth235/Lab04-SS20-SIR-Models-Student.pdf
5. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7787076/
6. https://www.nih.gov/news-events/nih-research-matters/lasting-immunity-found-after-recovery-covid-19
7. https://www.google.com/search?q=minnesota+population&rlz=1C1RXQR_enUS968US968&oq=minnesot&aqs=chrome.1.69i57j69i59l2j35i39j46i433i512j46i131i433i512l2j69i61.3707j0j1&sourceid=chrome&ie=UTF-8

8. https://www.google.com/search?q=minnesota+vaccination+rate&rlz=1C1RXQR_enUS968US968&oq=minnesota+vaccination+rate&aqs=chrome..69i57j0i512j0i22i30j0i22i30i457j0i22i30l4j0i390.5531j0j1&sourceid=chrome&ie=UTF-8