Paper

Nathan Hemenway, Mark Hinds, Ian Hall

12/12/2021

Background: The paper "Novel approach for Monte Carlo simulation of the new COVID-19 spread dynamics" aims to investigate the transmission and incubation random processes of Covid-19. The goal of the study is to generate epidemiological data based on the natural mechanisms of transmission of the virus assuming random interactions of a large-finite number of subjects in very short distances. A few important assumptions were made in how the transmission of the virus occurs when the physical distance between subjects decreases, so that the probability of transmission converges to one as the distance between subjects goes to zero. Their model defined the steps of movement in a defined space as a random process. They approached this through the SIR model; A model defining Susceptible, Infected, and Recovered individuals. Any change in the SIR model is based on time, as the function iterates through a set number of days. They started with a 0.5% infection rate and simulated daily infection counts over a number of days. In order to capture the many factors in transmission, including the relationship between infection probability and physical distance, effective and ineffective mask wearing, and environmental conditions they distributed these factors Exponential(rate = lambda). Initial recovery and incubation times were distributed Gamma(α, β) [1].

Motivation: The motivation behind our Monte Carlo Covid-19 simulator is to be able to better understand the spread of Covid-19. Through Monte Carlo simulation, we can observe what affect variables such as starting infected population, total population, physical space, infection distance (affected by masks), level at which the population is vaccinated, as well as incubation and recovery times have on the spread of the disease. These variables can be easily adjusted in our code to give different scenarios. The way we measure the effect changes in the aforementioned variables is by looking at the difference between the daily number of cases. This can be well visualized in a plot giving number of infected people per day. At the beginning of the Covid-19 pandemic, there were public health measures put into place that aimed at 'flattening' the curve of daily case counts of Covid-19. We are interested in seeing what effect different policies and population characteristics have on the plot of daily Covid-19 infections. If we can understand what affect different input variables have on the outcome of the pandemic, we can possibly make better policy decisions regarding Covid-19.

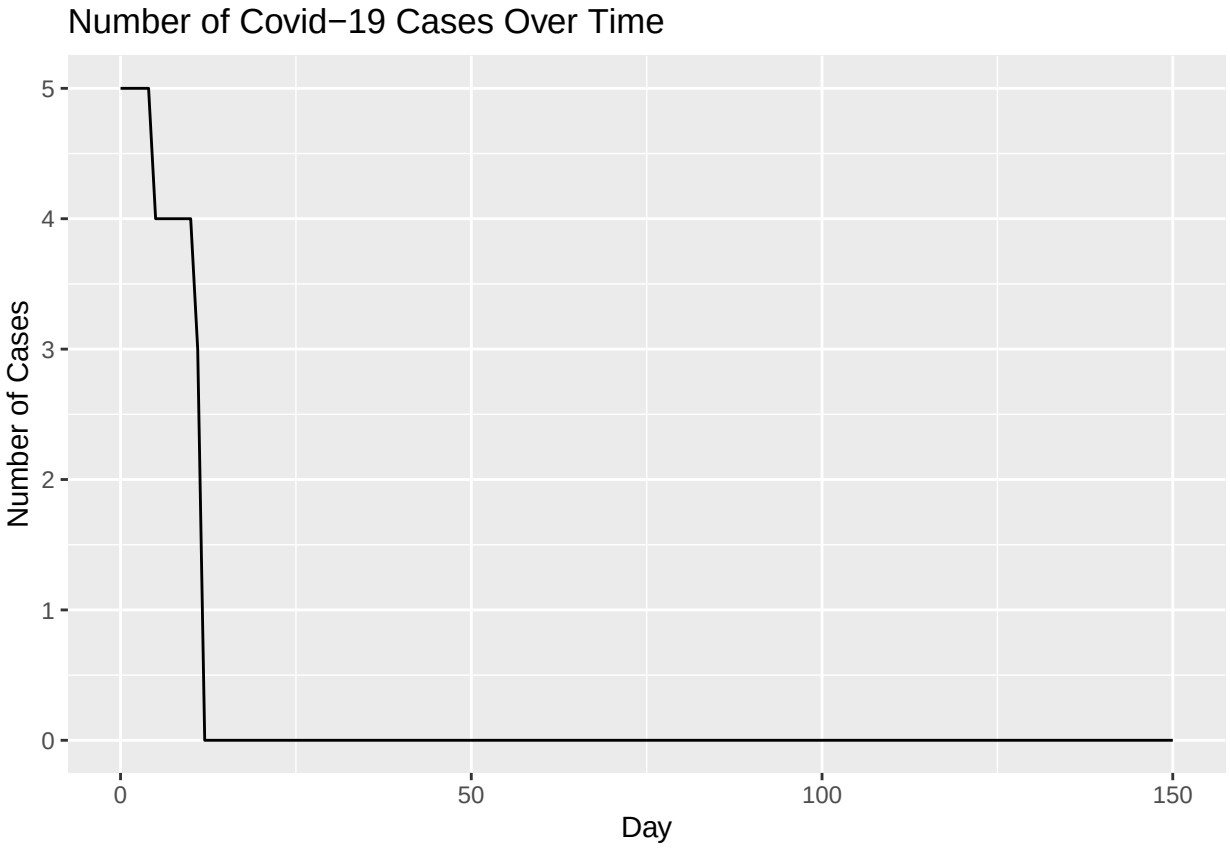
Methodology: To start, we created functions that generate a random step length. This was done using the rnorm function for the normal distribution with a mean and standard deviation of the minimum of the dimensions divided by four and twelve respectively. Once we have a step length we generate a step direction from a uniform(0, 2π) distribution using runif. This gives the movement direction angle in radians. From these two pieces of information, we can then update the coordinate values in the update_coords function by adding the change to the original x and y values. This is done using trigonometry equations for the legs of a triangle given the hypotenuse and angle.

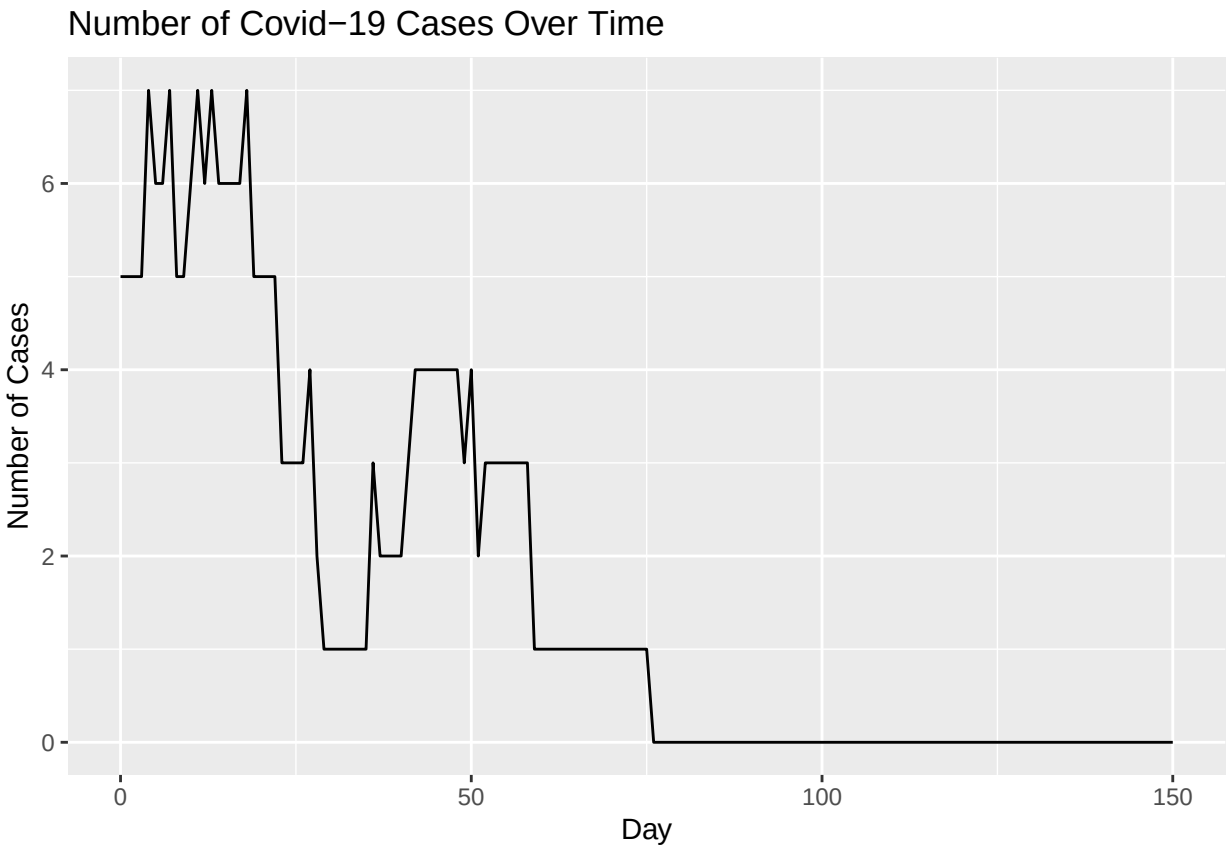
The problem that arises when using this approach is that subjects will inevitably leave the specified space they are supposed to remain in. The solution to this problem is given in the coord_checker function. For example, if we want the subjects to stay in a [0, 1] x [0, 1] square, and let them move in any direction, the mean step distance will be 0.25, so it would only take a centered subject a couple moves in the same direction before they leave the space. The solution we employed was to check and see whether a certain move would put a subject out of bounds, and if so, re-sample a different move. Then repeat this process for all subjects until they are all within bounds.

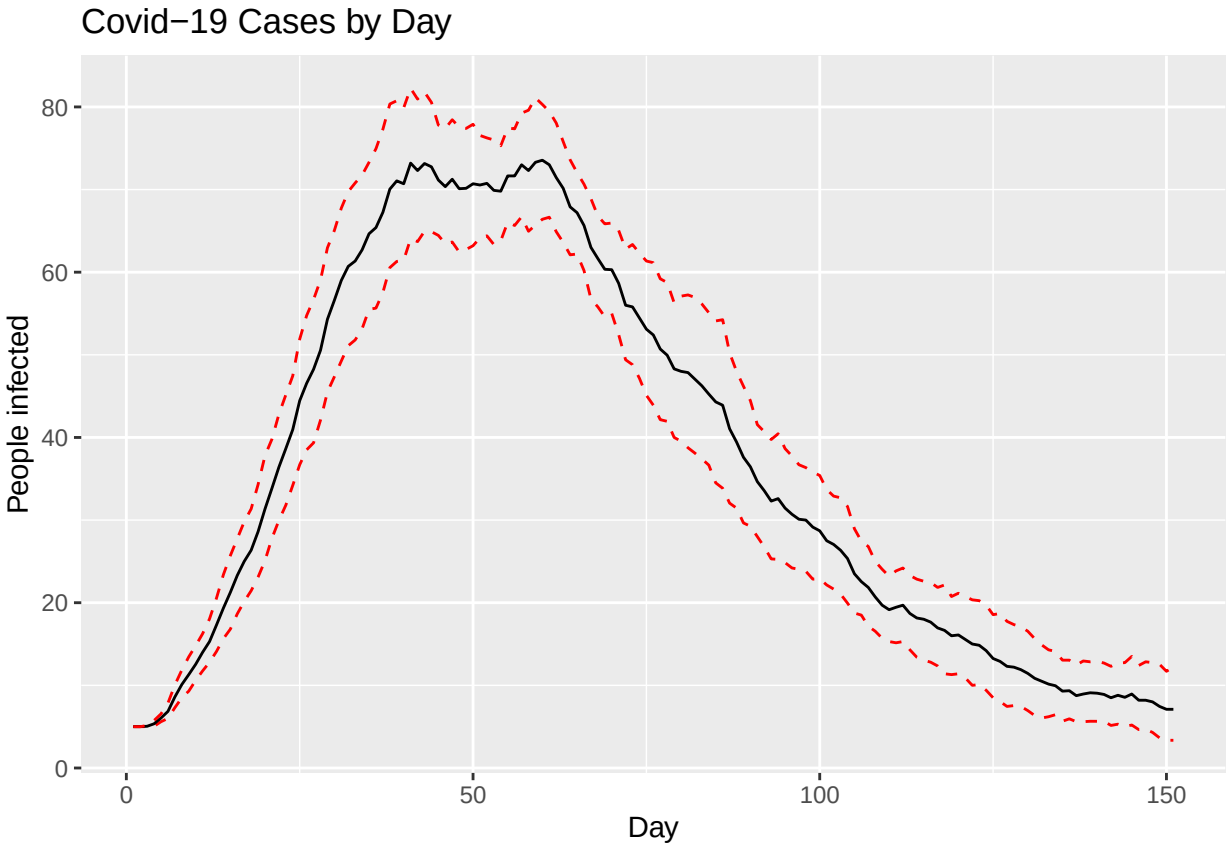
After moving the subjects randomly in the given space, the next step in the simulation is to check if any infected subjects have infected others. This is done via the infection_status function. The first thing we did was subset the subjects into different data frames by infection status. Then we can check how far the susceptible subjects are from the infected subjects. If they come within a crucial distance generated using an exponential distribution with the passed in mean crucial distance, then they are vulnerable to infection. If the susceptible person is vaccinated, we generate a binomial(1, 0.8) random variable to determine if the vaccine was effective in preventing infection. We chose 0.8 to be the likelihood the vaccine protects the susceptible subject against infection. The third condition necessary for infection to occur is that the incubation period of the virus has passed for the infected person. The incubation length is assigned upon infection using rpois(1, 5.15), and decreases by one each day until it reaches zero. If binomial vaccine effect returns false, and the susceptible person is within the generated crucial distance of the infected person, and the infected person is past the incubation period, then the susceptible person becomes infected. Once infected, a subject is assigned a recovery time using round(rgamma(1, 6, 2/3)). For each iteration in the simulator, one is subtracted from the recovery time. Once recovery time reaches zero, the infection status of the subject is changed from infected to recovered.

All the above culminates in the simulator function. This function takes in the given area, number of subjects, number of starting infected subjects, mean infection distance, number of days for the simulation, and proportion of subjects that are vaccinated as arguments. Using a for loop for each day, the function generates new positions for the subjects, and returns how many people have been infected per day. This can be used to generate a plot for the number of daily cases by plotting day on the x axis and number of cases on the y axis. If you run the simulation repeatedly, you can create a bootstrap confidence interval for the mean number of cases each day. This is done in the get_bootstrap_cis function. The function returns the bootstrap confidence intervals as well as the mean for each day, for a given sample size. For our purposes we used a sample size of 20 due to the computational burden of repeatedly running simulations. This data can be used to create a plot of the mean number of cases per day as well as 95% bootstrap confidence intervals for each day. We decided to use percentile bootstrap confidence intervals since they are the most interpretable.

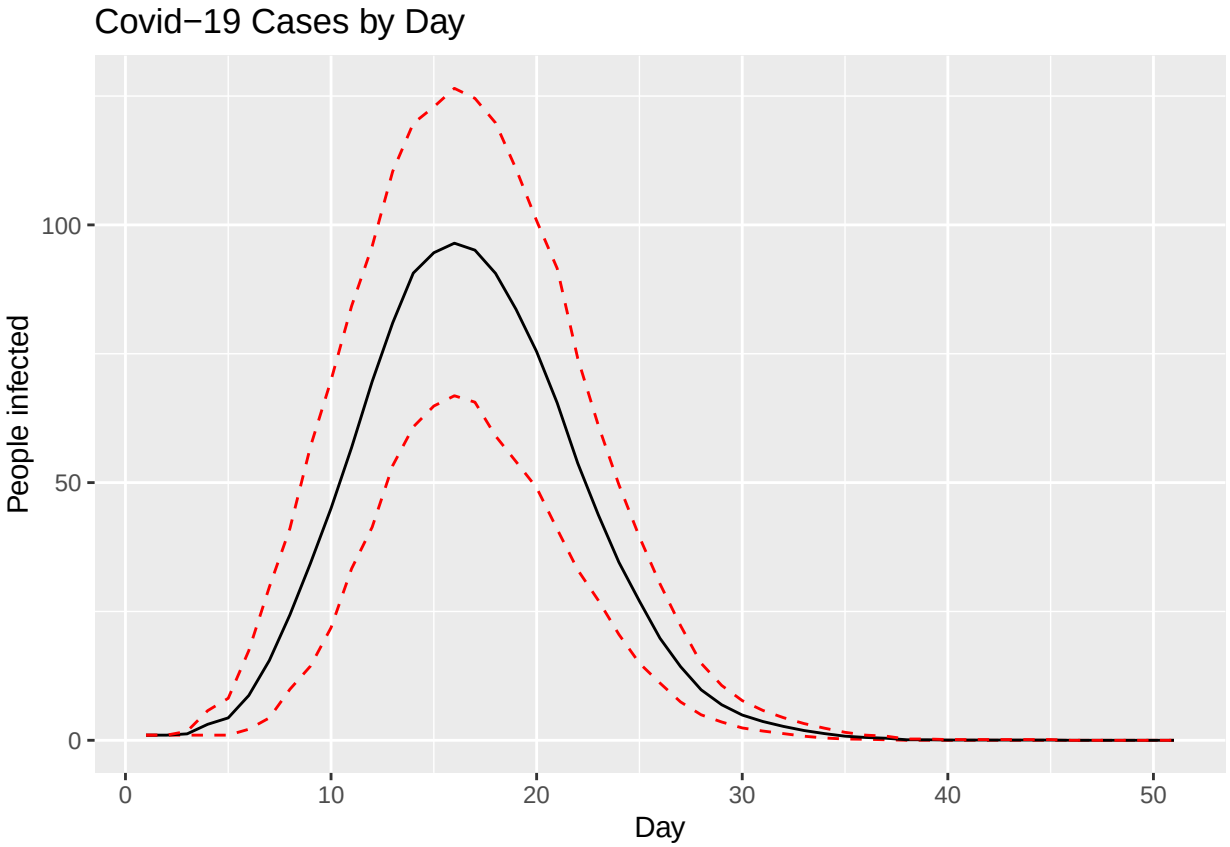
Results:

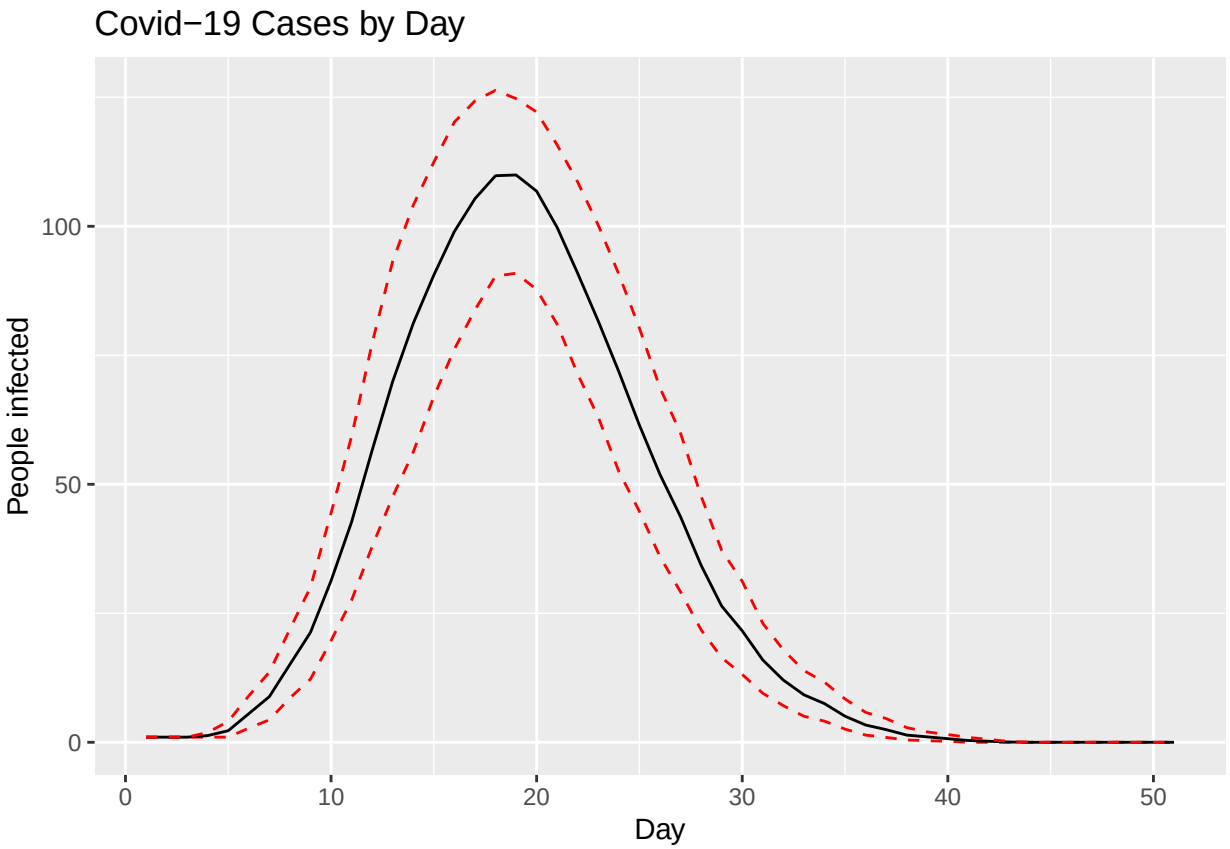

Number of Covid−19 Cases Over Time
Number of Cases
5
4
3
2
1
0
0
50
100
150
Day

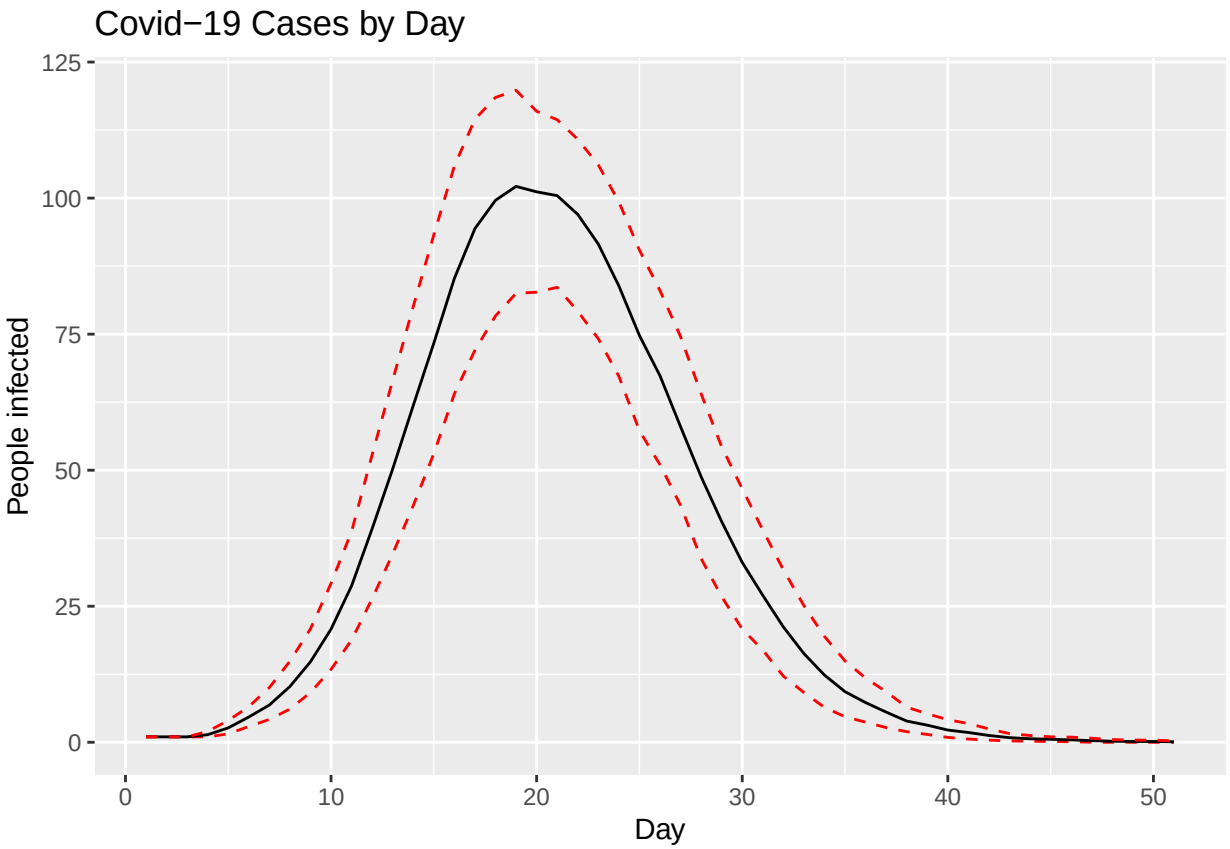

Number of Covid−19 Cases Over Time
Number of Cases
0
2
4
6
0
50
100
150
Day

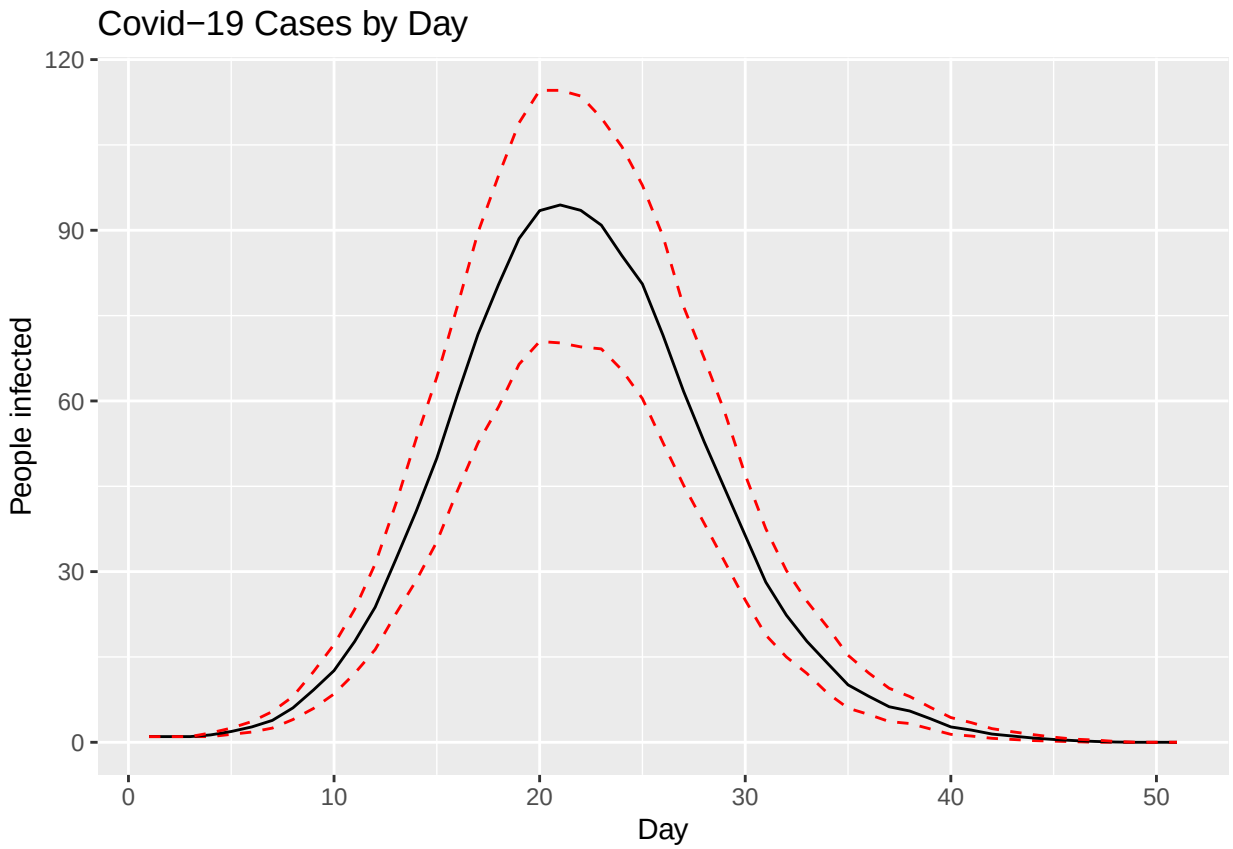


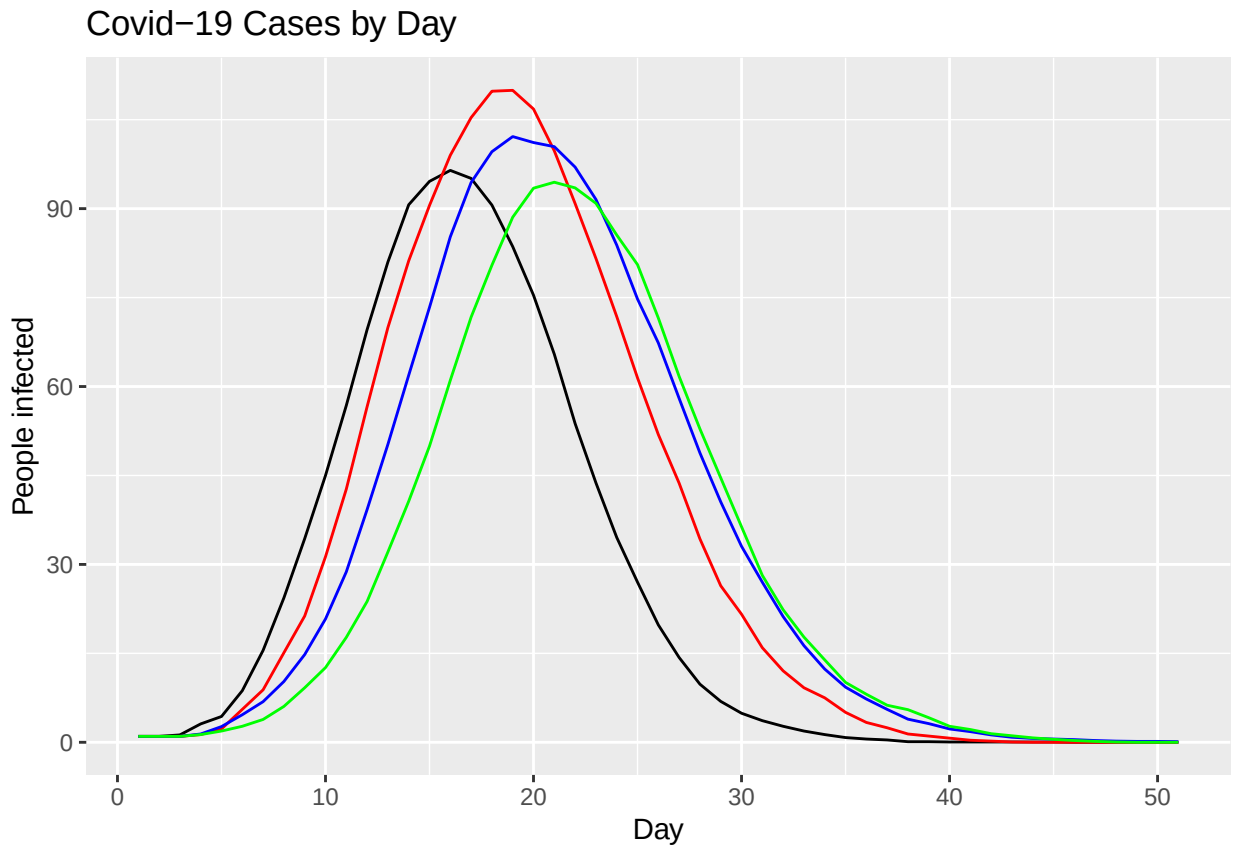
We started with 5 infected people in each simulation. In our first two simulations we were attempting to simulate a 6- and 26- foot infection distance. As you can see from the first two plots above, no one was infected from the 5 initial people who were infected for the mean crucial distance of 6 feet. For the crucial distance of 26 feet, some people got infected early on, but eventually everyone recovered and the number of infections went to zero. This is because in these examples we were attempting to simulate a small town with a low population density. This is supposed to represent 1000 people in a town that is 36 million square feet or about 1.3 square miles. Since the population density is low, it is unlikely for two people to come in close enough contact to contract Covid-19 in our simulation. This is a scenario in which our simulator might not be realistic. Our simulator is assuming every person is independent and randomly moves around from day to day. This doesn't account for seeing friends or going home to your family, which are common situations where contracting Covid-19 is more likely. Accounting for these factors might be possible, but it's beyond the scope of this project. In our third simulation, we used a bootstrap confidence interval for an infection distance of about 60 feet. This distance was finally large enough so that people still got infected despite the low population density. Based on the plot, the number of infected people will peak around 50 days and then slowly decrease as the majority of people start to recover and gain immunity. This simulation looks more realistic and follows the infection trend that viruses generally have.


Covid−19 Cases by Day
People infected
100
50
0
0
10
20
30
40
50
Day


Covid−19 Cases by Day
People infected
100
50
0
0
10
20
30
40
50
Day


Covid−19 Cases by Day
People infected
125
100
75
50
25
0
0
10
20
30
40
50
Day


Covid−19 Cases by Day
People infected
120
90
60
30
0
0
10
20
30
40
50
Day



In the previous three simulations we assumed that no one was vaccinated, but we also wanted to look at how a vaccine would affect the spread of Covid-19. In these simulations we looked at 200 people with the same dimension parameters as before, but with only one person initially infected. We increased the infection distance since 200 people in a 1.3 square mile area is an even lower population density than before. All the parameters were held the same other than the vaccination rate for these simulations. The vaccine rates of the four examples were: 0, 0.5, 0.8, and 0.95. In the plot that shows the number of infected subjects by day for each vaccination rate, the proportion of vaccinated subjects for black is 0 vaccinated, red is 0.5, blue is 0.8, and green is 0.95. As we would expect, when no one is vaccinated, the number of infected people peaks and falls much faster than when some people are vaccinated. It only took around 15 days to peak and nearly everyone was recovered by day 30. When no one was vaccinated it took much less time to get sick, but people also recovered faster because everyone already had the virus. This is the idea of herd immunity that is often talked about, but it poses a problem for hospitals. If everyone gets sick in a short period of time, hospitals can become overwhelmed and the death tolls can skyrocket due to lack of equipment and staff. Vaccinations are not perfect, accounted for in our simulator by using a binomial random variable to see if effective, but they can help flatten the curve so the infection numbers don't peak as fast. It is still possible to contract Covid-19 while vaccinated, but it is less likely than otherwise. As you can see in the last plot, as the vaccination rate increased, the longer it took for the cases to peak. The peak was also lower than when no one was vaccinated, and it took longer for the cases to completely disappear since people were being infected slower. This shows that a higher vaccination rate helps slow down the spread of Covid-19 even if it cannot completely prevent it.

Bibliography:

[1] Maltezos, S., & Georgakopoulou, A. (2021). Novel approach for Monte Carlo simulation of the new COVID-19 spread dynamics. Infection, Genetics and Evolution, 92, 104896. https://doi.org/10.1016/j.meegid.2021.104896

[2] Lauer, S. A., Grantz, K. H., Bi, Q., Jones, F. K., Zheng, Q., Meredith, H. R., Azman, A. S., Reich, N.

G., & Lessler, J. (2020). The Incubation Period of Coronavirus Disease 2019 (COVID-19) From Publicly Reported Confirmed Cases: Estimation and Application. Annals of internal medicine, 172(9), 577–582. https://doi.org/10.7326/M20-0504

[3] H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

[4] H. Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

[5] Angelo Canty and Brian Ripley (2021). boot: Bootstrap R (S-Plus) Functions. R package version 1.3-28., Davison, A. C. & Hinkley, D. V. (1997) Bootstrap Methods and Their Applications. Cambridge University Press, Cambridge. ISBN 0-521-57391-2