ORIE 4580 Course Project

Executive Summary

As a result of the COVID-19 pandemic, discussions over how to effectively mitigate the impacts of contagious disease outbreaks have come to the center of global attention. The effects that vaccine rates, population size, and individual behaviors have on epidemic outcomes are of the utmost interest to disease experts. To better understand the impact of these factors, our group created a model to simulate an epidemic within a population. We aimed to accurately represent a population by modeling people with different types of occupations, behaviors, and other risk factors. We also allowed for different parameters such as vaccine rate, recovery rate, contact rates, and population sizes to be adjusted in order to study their impacts. Our model also captures the tendencies and actions different types of individuals may take and what the consequences of these actions are, i.e. how the virus spreads.

The initial analysis we performed looks into the efficacy of vaccinations and if there are certain groups of individuals that should be prioritized when distributing vaccinations. Our analysis in full can be found in Model Analysis, but the main conclusions drawn are that vaccinations significantly reduce the spread of the virus and prioritizing certain groups when distributing vaccinations appears to be more effective than randomly distributing vaccinations. In particular, prioritizing non-essential high risk agents is most effective at mitigating the average number of individuals infected per day. However, it should be kept in mind that prioritizing this particular group is only marginally, not significantly, better than prioritizing any other particular group.

We attempted to make our model as lean and efficient as possible by only simulating the actions where transmission of the virus takes place. In addition, we structured and formulated our model in a manner that makes it easy and straightforward for you to adjust the various parameters and test how different potential policies impact the spread of the virus. In particular, parameters such as initial population size, initial infected agents, vaccine rate, recovery rate, meeting rates, etc. can all be easily altered to better reflect whichever population you wish to model. In addition, you can use this model in the future to test how different policies regarding the number of vaccinations administered per day, prioritizing different types of individuals when vaccinating, etc.

impact the spread of the virus. The rest of this report outlines the high level modelling approach, the key assumptions we made, a more technical description of our model, our initial analysis, the conclusions we drew, and a technical appendix.

Modeling Approach, Assumptions, and Parameters

The following is a high level overview of our model's approach, parameters, and assumptions. When thinking about an individual during a pandemic, we are curious about two main things: their current state/traits and their behavior. At a high level, we first approached this model by ensuring that we captured all the different traits an individual may have that we care about (parameters) and all the different types of people an individual could be (classes). We represented an individual within this model with the following parameters:

- *class_n*: a type representing a group of people they belong to
- *meeting_rate*: a float representing a rate at which the agent has an interaction that results in transmission of the virus
- *infected*: a boolean representing whether or not they are infected
- *p*: a float representing a probability of being symptomatic if they are infected
- *name*: a string giving a unique identifier to that agent

Now these agents belong to different groups that will tell us information about them and how they behave during the pandemic. Our model contains five main groups of agents, i.e. classes: medical workers, essential non-medical workers, non-essential high risk, non-essential low risk, and infected-symptomatic. You can think of these groups as "buckets" that agents can be placed into where no individual can be in more than one bucket at a given time. We will refer to these groups as "Bucket-A" in the rest of this paper. These groups are all represented by separate arrays in code.

In addition, our model includes three additional groups (Bucket-B): all recovered or vaccinated agents, all susceptible agents, and all infected agents. Similarly to Bucket-A no individual can be in more than one group in Bucket-B at a given time, but an individual may be in a group in Bucket-A and a group in Bucket-B at the same time.

One essential parameter in our model is the recovery rate is the proportion of the infected population that recovers each day and becomes resistant. This parameter was

set to 0.0714 in our model because we assumed that recovery takes 14 days on average, meaning 1/14 = 0.0714 of the infected population recovers each day.

Another parameter in our model is the vaccination rate. The vaccination rate represents the number of agents that are vaccinated per day. This parameter is currently set to 200, because we assumed that the fastest our population of 4,000 agents could be completely vaccinated would be 20 days This can be easily altered by changing the value of vaccine_rate in the code.

The last two parameters used in our model are initial population size and initial number of agents infected. Initial population size refers to the initial number of agents in each of the groups in Bucket-A; we assumed 1000 agents each initially in medical workers, essential non-medical workers, non-essential high risk, and non-essential low risk, and 0 agents initially in infected-symptomatic. In terms of initial amount infected, we assumed 5 agents initially infected in medical workers, essential non-medical workers, non-essential high risk, and non-essential low risk each, and 0 agents initially infected in the infected-symptomatic group.

Our model also captures the different things that could happen to an individual during a pandemic. An individual may get infected, vaccinated, or recover after being infected. These are represented as functions in our model that can be called on any individual. These functions move agents around between groups; this is explained in more detail in “Model Details” below.

Our model also needed to take into account the rates that agents in the different groups meet each other. The final rate agents in one class meet agents in another class is based one two things: a meeting rate and how the agents they meet are distributed between different classes (a proportion). The meeting rate is specific to each class and represents the daily rate at which agents in one class have a meeting with other agents where an agent is infected. Medical workers, essential non-medical workers, non-essential high risk, non-essential low risk, and infected symptomatic agents were assumed to have the following respective meeting rates: 0.2, 0.3, 0.05, 0.1, 0.0714, which we define as the number of meetings per day where infection is transmitted. It was assumed that essential non-medical workers have the highest rate of these meetings (0.3 people per day) because they see the largest number of people per day

due to the nature of their occupation. Medical workers were estimated to have a slightly lower, but still high rate (0.2 people per day). Most of the people medical workers interact with are infected patients, but they still have the potential to transmit to their coworkers and other essential and non-essential workers in their day to day lives, but they still interact with less people per day than essential non-medical workers. It was assumed non-essential high risk workers would have a lower rate (0.05 people per day) than non-essential low risk workers (0.1 people per day) as the high risk workers would be more cautious and would likely take extra measures to avoid interactions with others. We assumed high risk agents had some underlying condition that would make contracting covid more serious/deadly and low risk agents as having no underlying conditions. Lastly, we assumed the meeting rate of infected individuals would be the same as their recovery rate (0.0714 people per day). We assumed that a symptomatic patient would infect one medical worker on average during their infection, which has an average duration of 14 days. Since symptomatic patients only meet medical workers, this gives a daily (important) meeting rate of 1/14 = 0.0714.

The other component of the final rate classes meet each other is the population proportion matrix. The population proportion matrix represents how the total agents an individual in a class meets are distributed between different classes. Rows 1, 2, 3, 4, and 5 respectively represent medical workers, essential non medical workers, non essential high risk, non essential low risk, and infected symptomatic. For example the first row can be interpreted as a medical worker's interactions are 40% infected symptomatic, 15% essential non medical, 15% non-essential high risk, 15% non-essential low risk, and 15% medical workers. The other rows representing the four other classes can be interpreted in a similar fashion. Our model assumes medical workers interact with mostly infected symptomatic agents (estimated to be 40%), as that is what they spend the majority of their day doing and are equally likely to interact with any other class of individual. We assume essential non-medical workers interact with primarily high and low non-essential workers (estimated to be 35% each), as well as some essential workers and medical workers. We assume both high and low risk non-essential workers have similar distributions and interact with primarily other high and low risk non-essential workers, as these are their colleagues and often their

customers. Lastly, it is assumed that infected symptomatic agents only interact with medical workers.

Every meeting between two agents of specified classes in our model includes an infected agent and a healthy one, and we assume that the healthy agent becomes infected in every one of these meetings. Therefore, the rate of infections is determined by the meeting rates between agents of specified classes. The calculations we performed to find our meeting rates are described in greater detail in “Model Details”.

Aside from the meeting rates, our model also makes the key assumption that the probability that an individual is symptomatic if they have the virus is 0.8 for all types of agents (medical workers, essential non-medical workers, non-essential high risk, and non-essential low risk). This is the parameter p identified in the paragraphs above. The value of 0.8 was based on external scientific research (see “Appendix"); it is estimated that roughly 20% of agents with Covid-19 are asymptomatic, making the probability that an individual with covid is symptomatic.

Finally, our simulation only includes events that we defined as “important.” These events are someone becoming infected, someone getting vaccinated (becoming resistant), or someone recovering from an infection. Nothing else is modeled in the simulation, and we assume that any meeting results in an immediate infection, which is either symptomatic or asymptomatic and does not change after the initial infection. We then keep track of the number of susceptible, infected, infected symptomatic, and resistant agents in the total population at every event of the simulation based on all of our population arrays.

Model Details

As described in the previous section, the meeting rates we use determine the frequency of a person from a given class meeting with another person from a specified class. To calculate our final matrix of meeting rates, where each entry [i,j] represents the rate that a an agent from class i meets an agent from class j, we multiply:

- Meeting rate of [i,j], the maximum of the individual class meeting rates (described in previous section) of classes i and j

- Proportion matrix entry [i,j], the proportion of agents that class i meets in a day that are from class j
- The population of class i

The product of these 3 values gives us our final meeting rates (number of meetings per day) matrix. This matrix accounts for the asymmetry of meeting rates between two classes. For example, we know an essential worker such as a grocery store employee sees a large number of non-essential workers per day as part of their job. But these same non-essential workers that they are seeing likely do not see nearly the same number of essential workers per day. The matrix we have created accounts for this asymmetry by taking into account both the meeting rates of classes and *the proportion of agents they see from each class during a given day.*

After determining all meeting, vaccination, and recovery rates, we normalize them by dividing each rate by the sum of all rates of events that occur in the simulation. The simulation then runs by iterating over a while loop, where one event occurs during every iteration of the loop. To see an overview of the MC and state changes see the PNG file titled “4580 Markov Chain Diagram” in the github repo. At the beginning of each iteration, U, a uniform random variable on the range [0,1] is generated to determine which event occurs. U can be a value within different intervals within [0,1], where each interval corresponds to a specific event occurring, and the width of that interval is determined by the normalized meeting rate of that event. This method allows us to accurately model the frequency of different events such that their rates are directly related to their probability of occurring at each iteration of the simulation. At the start of every iteration, we update the meeting rate values since populations of different classes change as agents become infected and resistant.

The Agent class in our model has the functions infect(), vaccinate(), and recover(), which are called upon an Agent object when one of these events occurs in the simulation. When one of these functions is called on an Agent object, the object is moved to the appropriate arrays (both Bucket-A and Bucket-B), and its parameters are updated to reflect the new state of the agent being represented. This allows us to keep track of the number of susceptible, infected, infected symptomatic, and resistant agents in the total population at every event of the simulation.

If a recovery event occurs, we randomly choose an agent from the array of infected agents of the entire population and call the function recovery() on that Agent object. In a vaccination event, we choose an agent randomly from the set of susceptible agents in the entire population and call vaccinate() on that object, which moves it to the set of resistant agents where it will remain for the remainder of the simulation.

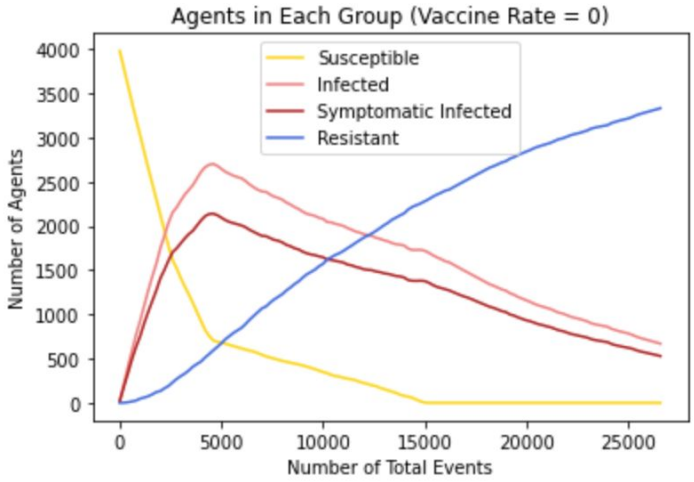
Model Analysis

This section details how one can use our model to compare different vaccination strategies. The two questions we wished to answer using our model was how much does vaccinating assist in reducing infections? and does vaccinating a particular group first result in a better outcome? The three main metrics we wished to compare were total agents vaccinated, average infected agents per day, and maximum infected agents per day. There were six vaccination strategies compared during this analysis: no vaccinations, randomly distributed vaccinations between the four groups, vaccination of medical workers first, vaccination of essential non-medical workers first, vaccination of non-essential high risk first, and vaccination of non-essential low risk first. The plots below show how the number of susceptible, infected, symptomatic infected, and resistant agents change as the simulation progresses. For each of the vaccination strategies we ran our simulation 20 times and included the mean and 95% confidence intervals for the three metrics we targeted in this analysis: total agents vaccinated, average infected agents per day, and maximum number of infected agents.

In the last four potential strategies, after the priority group was vaccinated the remaining vaccines of the initial 200 were randomly distributed between agents in the remaining three groups. We assumed a vaccination rate of 200 because we assumed that the fastest that our relatively small population of 4,000 agents in our simulation could reasonably be entirely vaccinated was 20 days, which would give us a rate of 200 per day. The most important thing to keep in mind here is that the vaccination rate was kept constant when comparing the potential strategies. Below are the mean and 95% confidence intervals for the three metrics outlined above for the six potential strategies.

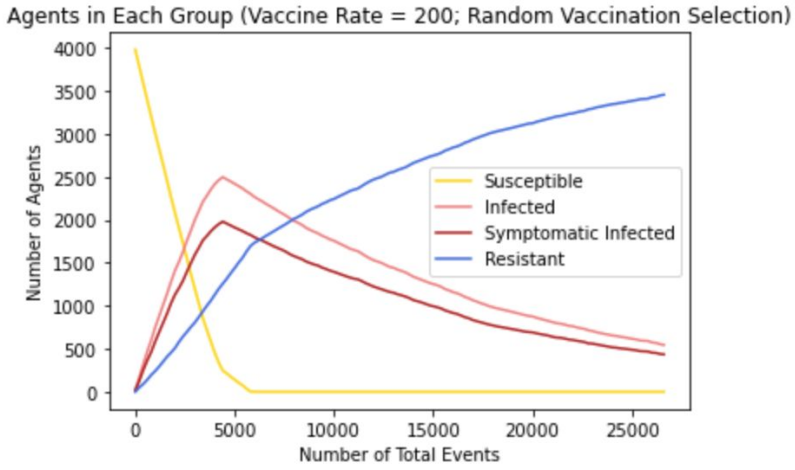
No Vaccine (Vaccine Rate = 0):

Total Agents Vaccinated: 0
Average Infected Agents: 1590.01 (95% CI: [1566.3, 1613.7])
Max Infected Agents: 2589.05 (95% CI: [2470.2, 2707.9])



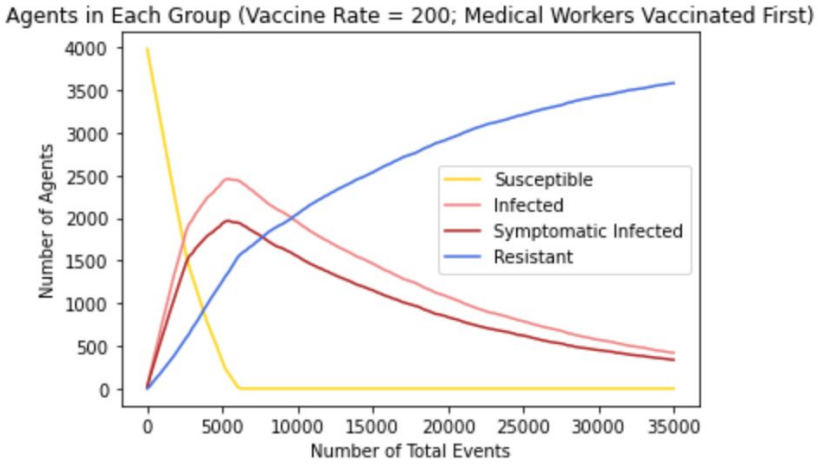
Vaccination Rate = 200; Random Vaccination Selection:

Total Agents Vaccinated: 1132.0 (95% CI: [1081.4, 1182.6])
Average Infected Agents: 1178.66 (95% CI: [1157.0, 1200.3])
Max Infected Agents: 2296.6 (95% CI: [2233.7, 2359.5])



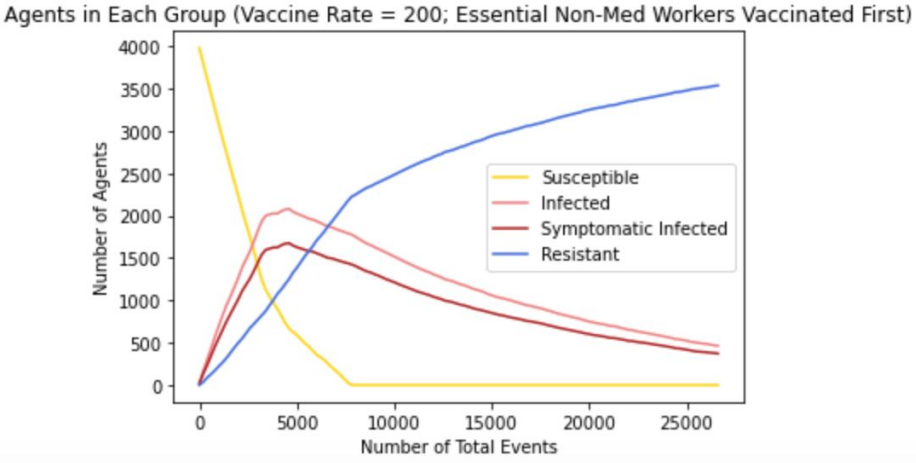
Vaccination Rate = 200; Medical Workers Vaccinated First:

Total Agents Vaccinated: 1183.55 (95% CI: [1117.1, 1250.0])
Average Infected Agents: 1160.28 (95% CI: [1132.8, 1187.7])
Max Infected Agents: 2266.55 (95% CI: [2182.4, 2350.7])



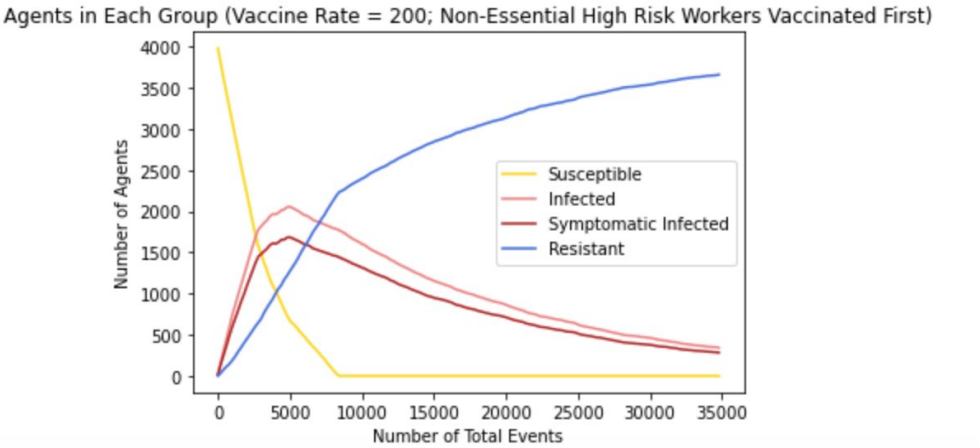
Vaccination Rate = 200; Essential Non-Med Workers Vaccinated First:

Total Agents Vaccinated: 1168.00 (95% CI: [1068.7, 1267.3])
Average Infected Agents: 1163.51 (95% CI: [1118.3, 1208.7])
Max Infected Agents: 2243.45 (95% CI: [2115.0, 2371.9])



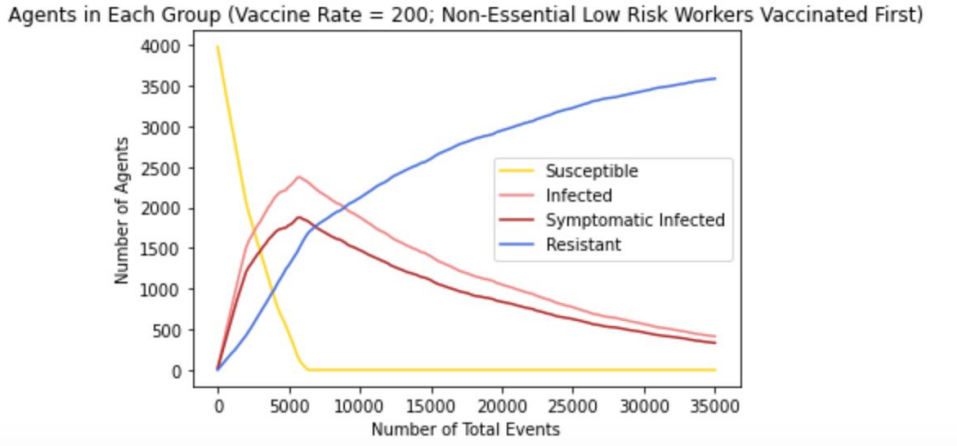
Vaccination Rate = 200; Non-Essential High Risk Workers Vaccinated First:

Total Agents Vaccinated: 1242.60 (95% CI: [1173.95, 1311.25])
Average Infected Agents: 1137.03 (95% CI: [1106.84, 1167.23])
Max Infected Agents: 2177.9 (95% CI: [2091.08, 2264.72])



Vaccination Rate = 200; Non-Essential Low Risk Workers Vaccinated First:

Total Agents Vaccinated: 1189.95 (95% CI: [1107.8, 1272.1])
Average Infected Agents: 1148.82 (95% CI: [1113.5, 1184.1])
Max Infected Agents: 2234.8 (95% CI: [2149.3, 2320.3])



Conclusions

The first conclusion that we can draw from the above analysis is that vaccines clearly are effective at mitigating the spread of the virus. The five last strategies administering 200 vaccinations per day result in between 412 to 453 less average infected agents per day as compared to the first strategy of administering no vaccinations, as well as between 293 to 411 less in the average maximum infected agents. This is a significant decrease especially when considering the fact that these

numbers are in the units of per day and not total. Therefore, we know that vaccinations are indeed effective at reducing the number of infected individuals and should be utilized in some capacity in your policy. The actual number of vaccinations you wish to administer will depend on your budget, but our model can be used to model any number of vaccinations per day.

Another conclusion that can be drawn from the above analysis is that prioritizing certain groups when distributing vaccines does mitigate the spread of the virus in some capacity. All the four strategies where some group is prioritized (vaccinating medical workers first, vaccinating essential non-medical workers first, vaccinating non-essential high risk first, and vaccinating non-essential low risk first), have a lower average infected agents per day than the strategy of randomly distributing vaccines across all four groups. The strategy exhibiting the lowest infected agents per day is the strategy of vaccinating non-essential high risk workers first with an average infected agents per day of 1137.03. One caveat to this strategy is that it requires the greatest amount of vaccines, about 1243. This means that although this strategy is the most effective at keeping the number of infected patients down, it will also likely be the most expensive and labor intensive.

However, it is important to note that the differences in average number of infected agents per day between the four strategies that prioritize a group do not vastly differ from each other; the 95% confidence intervals overlap for all of these strategies and the mean only differs between 10 to 30 agents. Therefore we cannot give a concrete recommendation of which group to prioritize, but still emphasize the importance of vaccinations.

Technical Appendices

In this section, the simulation code structure is discussed as well as the functions that are used during the simulation are discussed. As described in the model details section our code organizes the different classes into multiple arrays containing our agent objects. These arrays are initialized as global variables with the starting population for each class corresponding to each array. We then define our population matrix and our meeting rate matrix. In the actual simulation function we use 4 distinct

loops to ‘infect’ 5 of each class by running ‘.infect()’ (function discussed below) on any five agent objects in each class array. Our while loop is conditioned on time, ending only when our time variable, t, is greater than or equal to the input value determining the length of the simulation. Entering in the while loop, we update our time, calculate the new lambda matrix and total rate sum and generate the uniform random variable. Each event calls either *simOccSame* or *simOccDiff* which run the logic for different classes meeting each other (details discussed below in function descriptions). Our simulation function returns n_frames, a multi dimensional array containing arrays of the length of the susceptible array, the resistant array, the infected array, and infected symptomatic array at a given t. This variable, n_frames, is what’s used to plot and animate what has happened during the simulation. See below for a more complete description of individual function descriptions.

- *Infect:* This function is a method specifically for class Agents. It sets the infected attribute of the instance to True. The instance is then removed from the susceptible array and appended to the total infected array. Finally, we generate a uniform RV to simulate the probability that they are symptomatic or an asymptomatic case. If symptomatic they are removed from their class list, i.e. c1.
- *vaccinate:* This function is a method specifically for class Agents. It removes the instance from the global susceptible array and also removes them from their specific class array. Then appends them to the resistant and vaccinated arrays.
- *recover:* This function is a method specifically for class Agents. It removes the instance from the total infected array and removes them from their class array. Lastly, it adds them to the global resistant array.
- *disp:* This function is a method specifically for class Agents. Think of this as a basic repr function for the agents class, where it displays the “name” of the instance, i.e. “Medical Worker #5”.
- *simOccDiff:* This is a helper function that is called within the main simulation while loop, which models the interaction of two instances of different classes. The function takes two inputs, class type 1 and class type 2. It searches for an infected instance of either class and breaks if one is found. If an infected instance

of either type is found another loop is run on the other classes array searching for an uninfected (susceptible) instance. When a susceptible instance is found they are infected. If none are found, we continue with the main while loop.

- *simOccSame:* This function is very similar to *simOccDiff,* because it has the same purpose except for modeling the meeting between two people of the same class. This function searches for an infected instance of the input class and when one is found, it then searches for a suspsetible instance of the input class. When both of those events happen (an infected and susceptible both found) the susceptible instance is infected.
- *Bar_graph:* This function is for visualization purposes. This takes in our n_frame variable and then plots the bar graph animation showing the change in the length of the different global arrays (infected symptomatic, resistant, susceptible etc) over the number of total events.
- *Line_graph:* This function is for visualization purposes. This takes in our n_frame variable and then plots the line graph animation showing the change in the length of the different global arrays (infected symptomatic, resistant, susceptible etc) over the number of total events.
- *Average_inf:* This is a helper function for investigating and analyzing the results of our model. The input is our global variable n_frame (representing the epidemic at a certain point in time) and it calculates the average number of infected over the total epidemic. Used to generate the confidence interval of the average number of infected for a given set of parameters.
- *Max_inf_agents:* This is a helper function for investigating and analyzing the results of our model. The input is our global variable n_frame (representing the epidemic at a certain point in time) and this function finds the maximum number of infected individuals that occurred in the entire simulation.

Appendix

https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.1003346