Jason Kim
jk46965
COE 322

COE 322 Final Project: Infectious Disease Simulation

Abstract

The purpose of this project was to test how vaccinations and varying contagion rates affect the spread of a disease within a population. To do so, c++, a programming language, was used to create simulations of an infectious disease amongst people. Two classes were created, one of which focusing on individuals and the process of them being infected and thereby cured; and the other class focused on applying the first class to each individual within the population. Through running many simulations, findings showed that having more vaccinated people and a less contagious virus lessened the likelihood of the disease spreading, or rather slowed down the rate at which it spread - vice versa for the other case. However, a caveat was that if there were infections, the simulation would persist for longer, as compared to a population with less vaccines. Furthermore, the coded program proved that herd immunity in fact is real by testing if having 95% of the population vaccinated would significantly decrease the chances of the disease spreading.

Program

The infectious disease simulation model consists of two classes: ***Person*** and ***Population***. The *Person* class focuses on how the disease affects one, specific individual within a population. The *Population* class applies the *Person* class to each and every individual within a population. Lastly, the main function calls the classes and their specific member functions.

To begin with, the ***Person*** class consists of six member functions: *status_string(), update(), infect(n), vaccinate(), is_stable(), and check_state()*:

The *status_string()* method returns a description of the person's state as a string. If the person's status equals zero, the person is declared "susceptible"; if the person's state is greater than zero, the person is infected and declared "sick." if five days have passed, and the person has recovered from the infection, their status is equal to -1 and thus declared "recovered." Finally, if a person has been vaccinated, their status equals -2 and is declared "not susceptible."

The *update()* method updates the person's status to the next day. It updates the people who have had the infection for 5 days - used as a control and can be changed by adjusting argument parameters for *infect(n)* method - and recovered after the 5 days to a "recovered" *state*.

The *vaccinate()* method simply changes the person's status to -2 and thereby to vaccinate, regardless of the person's state. It is important to note that the *infect()* method must come before the *vaccinate()* method, as it must ensure that the person isn't vaccinated or recovered from the disease before infecting others.

The *is_stable()* method returns a *bool* indicating whether the person has been sick and is recovered. It utilizes an if statement that ensures that the loop continues as long as the person's state is less than zero. The *update()* method contributes to this method in that it is relied on to run the loop until a person is either vaccinated or sick, inhibiting self-infection or the contagion of others within the respective population.

Lastly, the *check_state()* method, utilized throughout the *Population* class, gets the state of an individual and directly applies it to the rest within the population

The ***Population*** consists of the following classes: *random_infection(), count_infected(), update(), display(), set_probability_of_transfer(probability), set_probability_of_vaccination(vaccprob),* and *contagion_of_random_people(probability)*.

The *random_infection()* method randomly chooses a person within the population to infect. It uses the *infect()* method from the ***Person.***

The *count_infected()* method finds the amount of people who are infected within a day. This method holds great weight, as it is responsible for stopping the simulation - breaks the loop once there are no infected people within the population.

The *update()* method simply calls the *update()* method from the ***Person*** class through a for loop so it can continue the simulation until the disease disappears from the population.

The *display()* method allows the user to visualize the state of each person within a population.

- "+" indicates a sick person
- "-" indicates a person who's recovered
- "?" indicates a person who's susceptible
- "x" indicates a vaccinated person

The next two methods, *set_probability_of_transfer(probability)* and *contagion_of_random_people(probability)*, are responsible for the disease spreading through contact:

The first method has it to where an infected person can infect the people next to him/her by generating a random number for the left and right person and thereafter looks for any sick people in the population vector - that is, if the random number is in the desired probability. After that, it checks if the right and left people are infected, and if not, they get infected.
The second method, *contagion_of_random_people(probability,* finds sick people, running through a for loop six times for each sick person, and picks a random person. If the randomly chosen person is susceptible, they are infected.

Finally, the *set_probability_of_vaccination(vaccprob)* method determines *x* amount of people in a population and runs through a loop *x* amount of times, randomly vaccinating people.

Exercises and Results

The program was built step-by-step through several exercises.

Exercise one had us code for one person catching the disease and recovering from it.

Exercise two required to develop the ***Population*** class and to visually display it using symbols. In order to check my program behavior and make sure everything was correct, vaccinated people were added to the display with the symbol "x."

Exercise three had us incorporate contagion, and investigate the spread of the disease from a single infected person. After running a number of simulations, there were cases where people escaped getting sick.

Exercise four incorporated vaccinations for the population. Findings show that the vaccination helped in reducing the spread of disease, but with the caveat that vaccinated people could still carry the virus and potentially infect others even without showing symptoms. In the case of this program, vaccinated people almost acted as barriers, inhibiting adjacent people from getting sick. Such a case doesn't happen in the real world, which goes to show that this model is unrealistic.

Exercise five compensates for the model's problem by introducing random people who may have contagion - a more realistic approach compared to the two people next to an infected person.

Population = 50,000 people:

Conditions	Trials	Lowest Days	Highest Days	Avg Days
25% vaccinated 35% contagion	8	2	21	15.1
50% vaccinated 35% contagion	8	2	24	17.4
25% vaccinated 25% contagion	8	2	31	19.0
50% vaccinated 25% contagion	8	2	36	18.7

The condition with 25% vaccinated and 25% contagion has the highest average days. This makes sense because with fewer people vaccinated, the more susceptible people there are, and the fewer contagions, the more time it'll take to infect susceptible people.

The condition with 25% contagion and 50% vaccinated people had the second highest average number of days. These conditions show less days than the prior condition with 25% vaccinated because having more vaccinated people increases the probability of having no one get infected.

The condition with 25% vaccinated and 35% contagion was the lowest average number of days. This is so because there are more susceptible people, and having a higher chance of becoming sick increases the chances of higher infection rates.

In comparison to the prior case, the condition with 30% vaccinated and 35% contagion has a slightly higher average. This is so because there are less people to infect others, which increases the time longer to have everyone infected.

From these findings and observations, we can conclude that having more vaccinations decreases the length of the runs throughout each population, whereas lower vaccination rates causes a disease to reach the entire population more quickly. Thus, it is reasonable to say that having higher chances of getting sick through contact shortens the time of simulations, which thereby results in lower number of days.

Population size: 100

Conditions	Trials	% of infected	Highest # infected
30% contagion 95% vaccinated	10	23%	23
30% contagion 90% vaccinated	10	35%	35
30% contagion 85% vaccinated	10	47%	47

For exercise 6, we were asked to find the vaccination rate at which a population reaches herd immunity. To know whether a population reached herd immunity, I considered the amount of sick people each day: if less than 30% of the population is infected for the day with the highest rate of infection, then the population is considered having herd immunity with its respective vaccination rate for a population of 100. I used a very small number for time's sake. I tested the same contagion rate which was 30% and tested several different vaccination rates. Findings showed that 95% of vaccination was needed in order for a population to reach immunity. The results show that out of the 10 trials, the highest infection rate was 23%, which is less than 30%, hence herd immunity.

The topic of getting vaccinated is controversial and holds ethical questions behind it. Although a shorter simulation time may seem better, it actually means that there'll be much more people getting infected in a shorter amount of time. As a result, this may overwhelm limited resources. Take for instance, when COVID-19 pandemic began, there was no vaccination, and the virus was spreading rapidly throughout the world. This caused hospitals to fill up with hundreds to thousands of sick people, at a rate where people were left untreated and dying. However, with more vaccinations in a population, the simulation runs slower, which helps reduce the risk of infection. Thus, it is important to consider vaccinations when herd immunity comes into question. It would nearly be impossible for a population to reach herd immunity if there weren't enough people getting vaccinated. And if such is the case, the simulation would continually run - on and on, constantly infecting more people day by day, making the pandemic neverending. From an ethical viewpoint, it is crucial that people get vaccinated even if they don't believe it'll reap any benefits for them - because it is the thought and concern for others who are susceptible to viruses/diseases.