Rhea Samuel

Victor Eijkhout

COE 322

5 December 2023

Infectious Disease Simulation Writeup

In this project, I am using C++ functions to implement an infectious disease simulation. Within this project, I created a different function for each exercise, instead of just removing/adding to the current methods, to be able to display the answer to each exercise and showcase how I built upon them for future exercises. The model design is structured to accommodate four distinct statuses: susceptible, infected, recovered, and vaccinated.

Initial Thoughts

Before coding, I outlined the necessary functions for the Disease and Person classes. In the Person class, "Get infected" changes the status to infected, "Get vaccinated" changes it to vaccinated, "Progress by one day" determines the status after a day, and "Status" establishes the current day's status. In the Disease class, I only implemented the instance variable: Chance of transmission and number of days a person stays sick when infected along with their get and set methods.

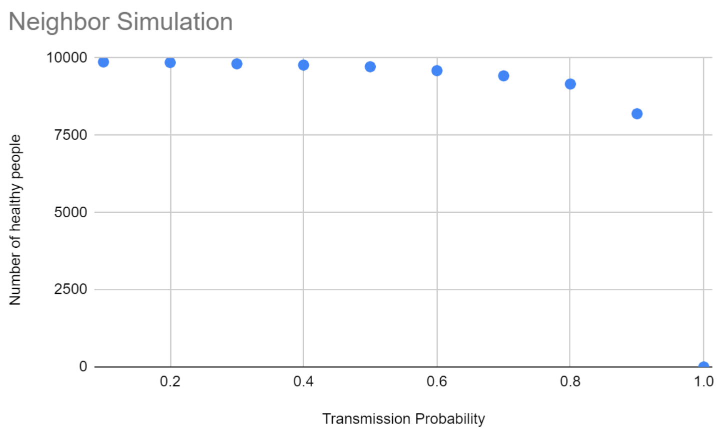
One of the tests I implemented were the Person tests using the Catch2 framework:

```
TEST_CASE("Infection with 100% transmittable disease", "[Person]") {
    Disease disease(5,1.0); // 100% transmission chance, 5 days
of infection
    Person person;
    // After being infected, the person should register as sick
    person.infect(disease);
    REQUIRE(person.get_status() == "Infected");
}
```

This is the person test written for the case: *After being infected with a 100% transmittable disease, they should register as sick.* I was able to implement different test cases for each question and received 100% passed test cases. The test cases will also be uploaded.

When looking at the bonus question, I understood the task as incorporating 'set' methods in the Disease class. I successfully added these set methods for both the chance of transmission and the number of days a person stays sick within the Disease class. I find this approach better than solely setting these values in the constructor because it provides flexibility to modify these variables at any point without the need to create a new Disease object.

Neighbor Simulation Results

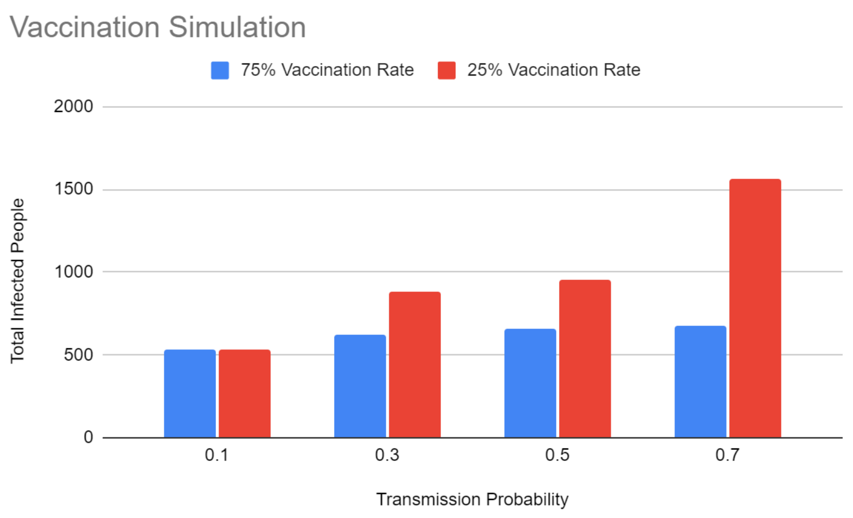


To enable direct neighbors to infect each other, I started by testing the function with smaller populations and a 100% transmission rate. Once it worked well, I scaled up to larger populations and tested with various transmission probabilities. The chart above displays the number of healthy people after running experiments with different transmission probabilities. Clearly, the results indicate that the concept of transmission probabilities plays a crucial role. A higher transmission probability leads to more infections, while a lower transmission probability results in fewer infections.

```
TEST_CASE("Test simulation with p = 0.5", "[simulation]") {
    Disease disease(5, 0.5);
    Population population(10000, disease);
    population.people[0].infect(disease);
    int days = 1;
    int x = 0;
    while(x < population.populationSize-1){
        population.one_more_day();
        population.neighbor(disease, 0.5, x);
        days++;
        x++;
    }
    REQUIRE_NOTHROW(days == population.populationSize); // populationSize
won't equal the days
}
```

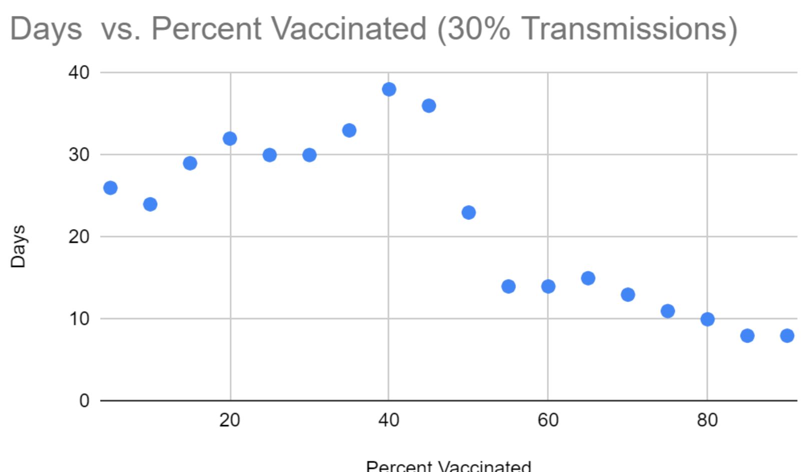
To ensure a specific number of people were sick per infected individual, I ran a few test cases for the neighbor function. For instance, if the transmission probability is 0.5 and only the first person is infected, there's a chance that not everyone will get infected. It relies on the transmission probability from each individual to the next. However, the process could also continue for some time. Consequently, the number of days the simulation runs may not be equal to the population size.

Vaccination Simulation Results



I now added the option for individuals in the population to get vaccinated. To assess the impact of vaccination on the infection rate, I conducted tests with various transmission probabilities and vaccination rates. The graph above illustrates the results, with the blue columns representing a 75% vaccination rate and the red columns representing a 25% vaccination rate. It is evident that the higher the vaccination rate in a population, the less likely the infection will spread. However, it's important to note that this model may not fully reflect reality due to factors such as the effectiveness and duration of immunity from vaccinations, as well as the presence of different disease variants.

Spreading Simulation Results



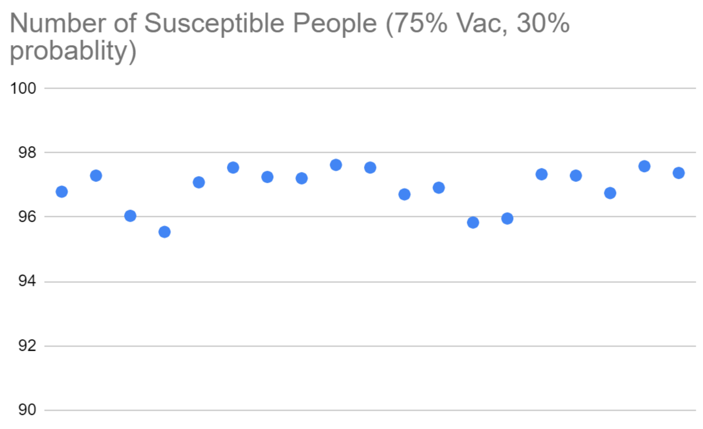
I added a function to establish random contacts instead of just neighbor contacts. The function was set for 6 contacts with a transmission rate of 30%. In the graph, I observed that the number of days is influenced by the vaccination percentage. Until the vaccination rate reaches 40%, the number of days remains relatively high, and it only starts decreasing significantly when the rate surpasses 45%. This observation suggests that fewer people can get infected when more individuals are vaccinated, resulting in a shorter duration for the simulation.

Herd Immunity Simulation Results

(1):

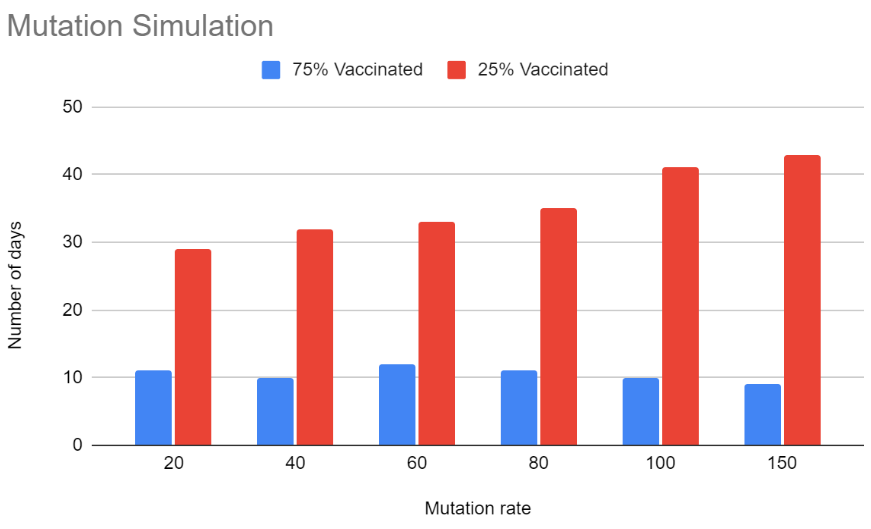
Transmissio n Probability	Vaccination Rate									
	0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	
0.1	98.797	99.0886	99.1739	99.1016	99.3061	99.2820	99.3103	99.4210	99	
0.3	35.4606	47.2151	61.2463	78.8135	91.0816	96.0769	95.8275	98.4210	98.4444	
0.5	8.9887	12.8101	18.6811	26.4067	39.4285	63.4871	86.1379	93.4210	96.55556	
0.7	3.1235	4.6962	7.33333	11.6610	19.775	30.6666	59.1724	85.3684	95.55556	

(2):



Herd immunity occurs when 95% of the original susceptible individuals remain susceptible by the end. In the initial graph (1), I systematically tested different vaccination rates from 0.1 to 0.9 and transmission probabilities from 0.1 to 0.75. After thorough testing, the most accurate values were determined to be a vaccination rate of 0.75 and a transmission probability of 0.3. While considering options between 0.7 and 0.8 for the vaccination rate, testing with 0.75 proved to be the most effective. Regarding transmission probability, 0.1 was considered unrealistic, so I settled on 0.3. In figure 2 (2), the results show that the percentage of susceptible individuals remains above 95% with a transmission probability of 0.30 and a vaccination rate of 0.75. Therefore, herd immunity is achieved at these specific values.

Mutation Simulation Results



I successfully integrated mutation with various variants into the functions. By experimenting with different mutation rates, I compared how these rates correlate with the number of days, considering two vaccination rates. In the case of 75% vaccination (blue column), not much change is observed, as a significant portion of the population is already vaccinated. However, with a 25% vaccination rate (red column), notable differences are seen, reaching higher values than the number of days found in the spreading simulation. It's important to note that this analysis was conducted with a transmission probability of 0.3.

Further Thoughts

After exploring the question, "The chances of something happening to me are very small, so why shouldn't I bend the rules a little?" I've come to the conclusion that bending the rules in all cases is not a wise approach. Many individuals share similar beliefs, and when people bend rules collectively, it can have a significant impact on the entire population. Increasing vaccination rates can contribute to an overall reduction in transmission and, consequently, aid in achieving herd immunity, as we examined earlier. Lower vaccination rates, on the other hand, can facilitate the spread of the disease. While the impact might seem minimal if only a few people bend the rules, I believe it could be less detrimental to society if herd immunity is established.