# C++ Model Disease Propagator

Kartik Nagpal

December 2018

## 1 Introduction

For this project, we were tasked with the creating a model for the propagation of a Disease through a population, with multiple potential states for a person. A person can either we "susceptible", "sick", "inoculated", or "recovered". The process of the model is that we must have a Person, of which we can create a group (in the form of an array or vector). Both of which must be able to take on states and propagate over time.

## 2 Implementation

The first part of the implementation is the creation of the Person class, and how we can model each state in relation to the specifications of the problem. Thus we must create a Person class, with variables storing state-values, counters(for general life, and for how long the individual has been sick), and most importantly must include methods which allow us to retrieve the encapsulated state values, infect the individual, and to update it's condition, so that we can then propagate over it's life. Once this was created, I went on to create a People class which held a vector of Person objects, and had it's own state variable and methods. Then we needed to establish the ways in which a population interacts, so in any People Object $Pop$, any

$$p, q \in Pop$$

if $p$ isInfected(), then $q$ has a chance of becoming infected. By default, the chance of being infected is **5%**. Lastly, I needed to construct the methods to generate Populations, with different characteristics, and then to store the states of each Person over the course of the propagation to a file.

At the end, I have it such that the input:

```
Person p1("Joe");
p1.status();
cout << p1.infect(5)<< endl;
cout << p1.update() << endl;
p1.propogate(-1);
```

We have the output

```
On day 1, Joe: Is susceptible but not infected.
On day 2, Joe: Infected for 5 more days.
On day 3, Joe: Infected for 4 more days.
On day 4, Joe: Infected for 3 more days.
On day 5, Joe: Infected for 2 more days.
On day 6, Joe: Infected for 1 more days.
On day 7, Joe: Is not susceptible!
```

And for the input:

```
People pop("Small Town", 10, 1, 0);
pop.status();
pop.propagate(6);
```

We have the output:

```
Day 1: 0   0   0   1   0   0   0   0   0   0   0
Day 2: 0   0   0   1   0   0   0   1   0   0   0
Day 3: 0   0   0   1   0   0   0   1   0   1   0
Day 4: 0   0   0   1   0   1   0   1   1   1   0
Day 5: 1   0   0   1   0   1   0   1   1   1   0
Day 6: 1   0   0   -1  0   1   0   -1  1   1   0
Day 7: 1   0   0   -1  0   1   0   -1  1   -1  0
```

And so, now that we have everything coded up, we can start running simulations and drawing conclusions from the data.

# 3 Results

Since there is a level of randomness and chance in the calculations, I averaged the scores over 40 propagations of identical populations for all data, and every statistic reported.

| Population Size | Average Days Till Healthy |
|---|---|
| 1 | 6.55 |
| 2 | 27.375 |
| 3 | 34.45 |
| 4 | 38.8 |
| 5 | 53.15 |
| 6 | 58.375 |
| 7 | 51.975 |
| 8 | 58.175 |
| 9 | 58 |
| 10 | 59 |
| 11 | 68.5 |
| 12 | 65.375 |
| 13 | 70.825 |
| 14 | 61.725 |
| 15 | 74.2 |

I continued gathering data up to a population size of 100 and graphed the result, as scatter plots. Additionally, I have put a restriction on the propagation, such that if it takes more than 700 days, it will be aborted. As such, this will skew the later data slightly, but due to the averaging over 40 runs, the discontinuity should be smoothed out.

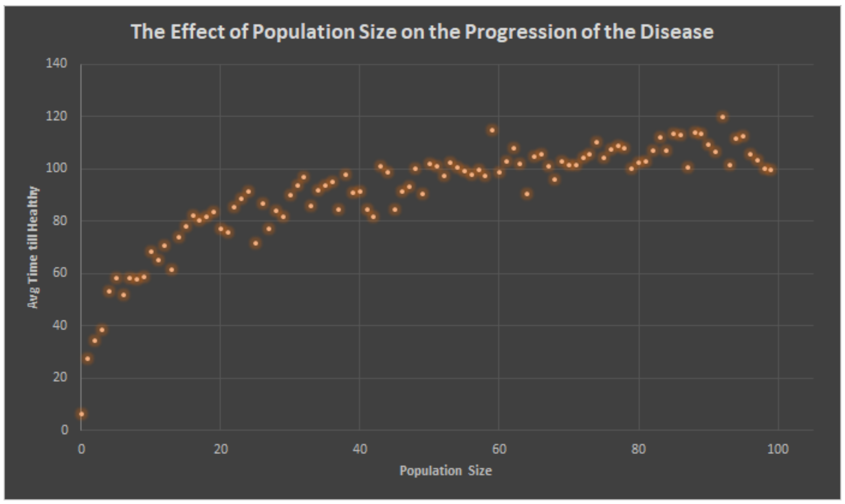


Figure 1: The Effect of Population Size on the Progression of the Disease

And thus we find that there is a positive correlation between the number of people in the population and how long it takes for the infection to get through the entire population. Next I did a general experiment on the effect that percent inoculated had on the progression of the disease, and again the 700 day cap has distorted the data, but once graphed, the correlation is clear despite the skewing.

| Percentage Inoculated | Average Days Till Healthy |
|---|---|
| 0% | 88.65 |
| 3.3% | 77.45 |
| 6.7% | 85.8 |
| 10% | 73.55 |
| 13.3% | 81.6 |
| 16.7% | 85.9 |
| 20% | 87.05 |
| 23.3% | 83.85 |
| 26.7% | 88.4 |
| 30% | 80.25 |
| 33.3% | 72.4 |
| 36.7% | 82.4 |
| 40% | 76.85 |

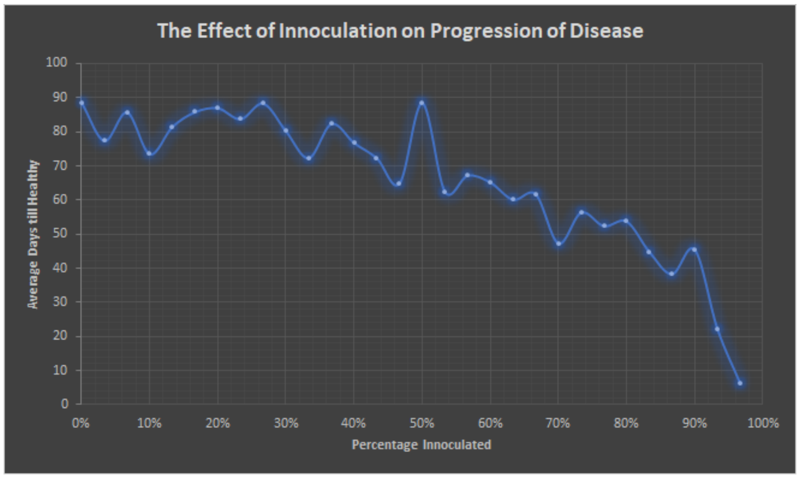


Figure 2: The Effect of Inoculation on the Progression of the Disease

We can clearly see a negative correlation between the percentage inoculated and the time it takes for the disease to propagate to a close. Next we can look at the effects of changing the chance of infection, when an infected person interacts with another susceptible person.

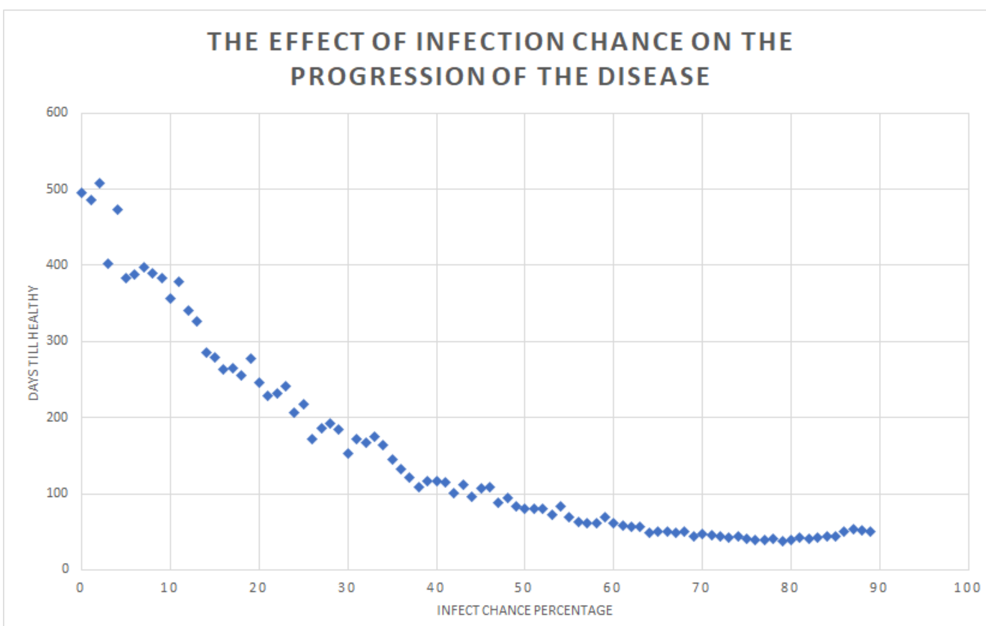


Figure 3: The Effect of Infection Chance on the Progression of the Disease

Here we can see the negative correlation, as the ”infection chance number” increases, the chance of generating a random number in the interval [0,.99], larger than it, grow slimmer, and so these results are as expected. Next, we are looking at Herd Immunity, and how at a particular inoculation point, there will be some portion of the population that never get sick. So I wanted to find, that in a series of 100 trials, what percentage of the trails ended with Herd Immunity. As under the value 60, there were no Herd Immunity Stops, I only plotted above those values.

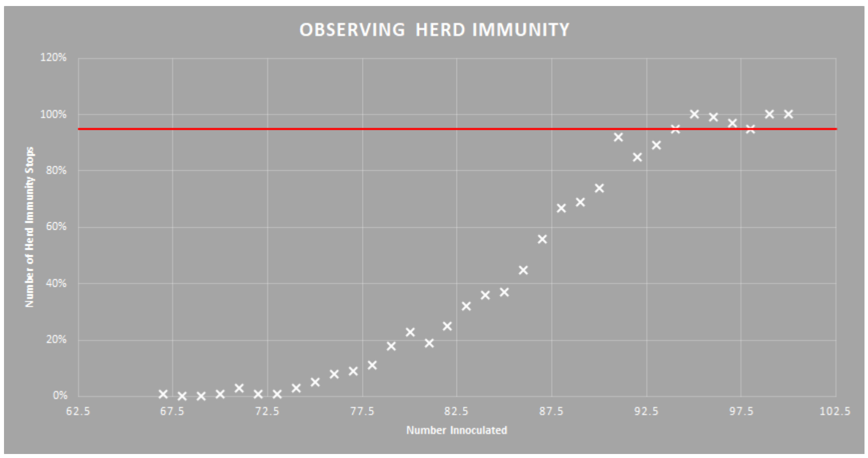


Figure 4: Observing Herd Immunity

Here we can see that a Herd Immunity stop can be achieved with $\geq 95\%$ certainty, which is illustrated by the red line threshold, is for cases where 94% or more of the population is inoculated.

# 4 Conclusion

Using an Object Orientated Programming approach, I wrote the code which can be found here. The results of our testing, tells us about the measurable effects of Herd Immunity, as well as the relationships of the output ”Days till Healthy”, in comparison to these input variables:

| Positively Correlated | Negatively Correlated |
|---|---|
| • Population Size | • Percentage of the Pop that’re Inoculated<br>• Infection Chance Number |