EID: RQL85

Infectious Disease Simulation

By: Robert Liao

Abstract

Being able to create a basic model to simulate infectious disease propagation and growth in a population is useful as a basis to develop more sophisticated models about specific viruses. For the purposes of this project, I will be incorporating some fundamental elements for disease modeling. These elements are: probability of transmission, time required to recover, patient status (sick, susceptible, recovered, or vaccinated) as according to the SIR (susceptible, infected, and recovered) model, two kinds of spread modeling (direct neighbor spread-only and the SIR epidemiology model), and a simple mutation simulation model (with the same inherent probability of transmission and required recover time as the parent disease). I will also explore the modeling for diseases without vaccines such as Ebola and COVID-19 (including functionality that will account for incubation periods).

Introduction

As mentioned above, I will be working with simple probability transmission information and a constant recovery time associated with my theoretical disease. For the sake of future and currently relevant analysis, I will assume a probability transmission of ~10% for COVID-19 (as of writing, Dec. 4th, 2023) [1] and a recovery period of 5 days (the minimum recommended recovery time) [2]. For later testing and pattern determination the probability of transmission and the percentage of the population that is vaccinated will be varied in an attempt to find a trend. Regarding the patient status, the four states a patient can be in are defined as follows:

- Sick - sick, and able to infect others
- Susceptible - healthy, but can be infected
- Recovered - was sick before, but can no longer carry disease/infect others or be infected again
    - Note that later when accounting for mutation, recovered individuals are only immune to the initial variant they got sick with, and do not develop immunity to other variants they have not been exposed to/infected with
- Vaccinated - healthy and can not carry the disease nor infect others

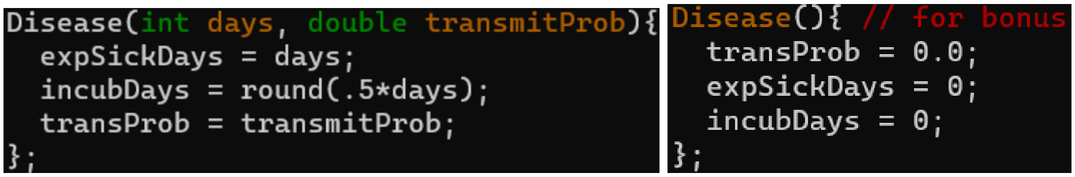
For the purposes of keeping the modeling simple, I made the assumption that anyone who is sick is infectious throughout their recovery period. Moving onto more detailed software implementation, the coding breakdown and walkthrough will be in the next section.

Implementation and Findings

Modeling Considerations

a. Person & Interaction Basics

To start off, it is essential to model an individual person. Key things to consider are the ability to get infected, get vaccinated, and if sick, how far along the recovery process they are. It is also important to model a simple disease functionality that will incorporate the chance of transmission and the number of days an individual will stay sick once infected. In practice, it is easiest to make account for a person with a Person class that also implants a Disease class as an internal variable (i.e. each person object will have an associated private class variable, disease for the associated disease object being modeled). Note that each person object will account for the current status of the person, an identifier (can be a name or patient number, in my case I chose a patient number to make large sample analysis and bookkeeping simpler), and if they are under a sick or recovered status, the number of days left till recovery and the current disease that they have been infected with. For the disease class implementation, there will be a touch method (which implements the interaction between two individuals) that will be associated with an infection method (which will model the infection process). For the infection method, I make the simple assumption that each person has a random genetically determined immunity towards the disease (modeled as a simple random number in decimal format generated between 0 and 1). To become infected, the person object of interest must have an immunity chance less than or equal to the probability of transmission of the disease (in this case 10%, meaning that we would expect only 10% of the individuals to be infected once touched by an infected person). As a side note, I found it beneficial to implement a default constructor and a specific constructor for my disease class. This is because a null default constructor could allow me to initialize a null disease object when initializing a person object, and later update that variable to the correct disease when that individual has interacted with one. This is useful for bookkeeping, and also leaves room for implementation of multiple diseases in a population to be modeled in one simulation if desired (though I will not be implementing this for this report). Below is a comparison of the implementation difference:



```
Disease(int days, double transmitProb){
  expSickDays = days;
  incubDays = round(.5*days);
  transProb = transmitProb;
};
```

```
Disease(){ // for bonus
  transProb = 0.0;
  expSickDays = 0;
  incubDays = 0;
};
```

Graphics 1: Disease Class Constructors

As we can see the default constructor on the left side, takes inputs and initializes the Disease class private variable properties accordingly. On the right side however, the null default constructor initializes all the same variables to null or zero values. This is convenient if we want to make a null disease option and later change is as needed. Note that to change the variables of the null Disease object, the implementation of a setter or getSetter method functionality will be required. Rather than making separate get and set methods for each variable, it would be much easier to implement a getSet method that allows you to get/return variables of interest but also edit them outside of the class using the same method. This can be achieved with a pointer notation for the method type declaration in the header (or wherever the method is constructed) file. The code associated with this report has further comments on this matter.

b. Population and Epidemics

Moving forwards, an overall population functionality is desirable for macro-management and oversight of the simulation to allow for ease of interpretation and generalization. To do so, it is easiest to implement a Population class to model the various person objects and the disease status associated with them. In the population class I found it useful to have a method to initialize a random starting segment of infected individuals in the population along with another method to initialize a random starting segment of vaccinated individuals in the population. It is also important to have counting methods to count the total number of infected and vaccinated individuals in the population. Furthermore, it is crucial to also incorporate a temporal advancement method that can update the conditions of the population. Since diseases usually run through a population on the timescale of months or years, for the smaller simulation size I will be using a time scale of days. Additionally, since the recovery time for a disease is typically measured in units of days, it makes sense to make my temporal advancement method a “one-more-day” method that will advance the population’s (and by association each individuals’) status by another day.

In general it can be said that with a simple no-contact simulation of a disease, the only infected individuals would be initial infectees (and by extrapolation, the only recovered individuals will be the same initial infectees after the total number of days has reached the required recovery time associated with the model disease). A more realistic model will be one that includes contagion principles so that the disease can spread to others.

c. Contagion and Spreading

As mentioned earlier, for the sake of this project, I will be testing two simple disease spreading models. The first model will be a direct neighbors model, where only the person before and after an infected person can be infected. In this model, there can be several cases where unvaccinated (and therefore susceptible) individuals can escape infection if they are not near any infected individuals and the probability of transmission is low enough. With these two models, taking into account vaccination and its role in preventing infection is also beneficial to have. For ease of implementation, I will make a simplifying assumption that the vaccine is 100% effective (meaning that any vaccinated individual will never be infected by the same disease as the vaccination). In reality, this assumption is unrealistic because all vaccines have limited effectiveness and are not guaranteed to prevent infection (not to mention that most will require subsequent booster doses to help maintain a high level of relevant antibodies in order to retain its efficacy against the target disease). Moving on, the second model that will be implemented will be the SIR model, which is a general compartmental model for modeling epidemiology. For this model, I will assume that every person comes into contact with about 6 people per day. The implementation for this model takes into account the current statuses of individuals that interact with one another. For example, an infected person will not be able to infect a vaccinated individual or someone who is already infected, or someone who has recovered from the same disease previously.

Simulation Findings

a. Spread Model Trends

When implementing the direct neighbors infection model, below is a table, Figure 1 that characterizes various population sizes and contagion probabilities. For these simulations, assume a constant population of 5000 people. The numbers reported are averaged across 5 simulations for each test parameter.

Transmission Probability (p)	Percent Vaccinated	Total infected (then recovered on average)	Total susceptible	Days Disease ran its course (measured in days)
0.2	0.0	18	4982	28.3
0.2	0.5	9	4991	12
0.4	0.0	132.1	4867.9	173.4
0.4	0.5	14.2	4886.8	13.78
0.8	0.0	910.6	4099.5	940
0.8	0.5	63	4937	9.4

Figure 1: Neighbor-Infection Model

As seen from the table, there are many cases where patients are able to escape infection in the neighbor model, especially when the vaccination rate goes up. We see that for the same transmission probability, as the vaccination rate goes from 0 to 50%, the total infected drops almost by almost 90% in most cases. The days required for the disease to run through also gets cut by roughly 90% in most cases.

For comparison, see below in Figure 2 that showcases a SIR model with a number of contacts of 6 people with a fixed probability of transmission of 10% (as we expect for COVID-19 same as mentioned in the introduction) that plots days disease runs through population versus the percentage of the population that is vaccinated.

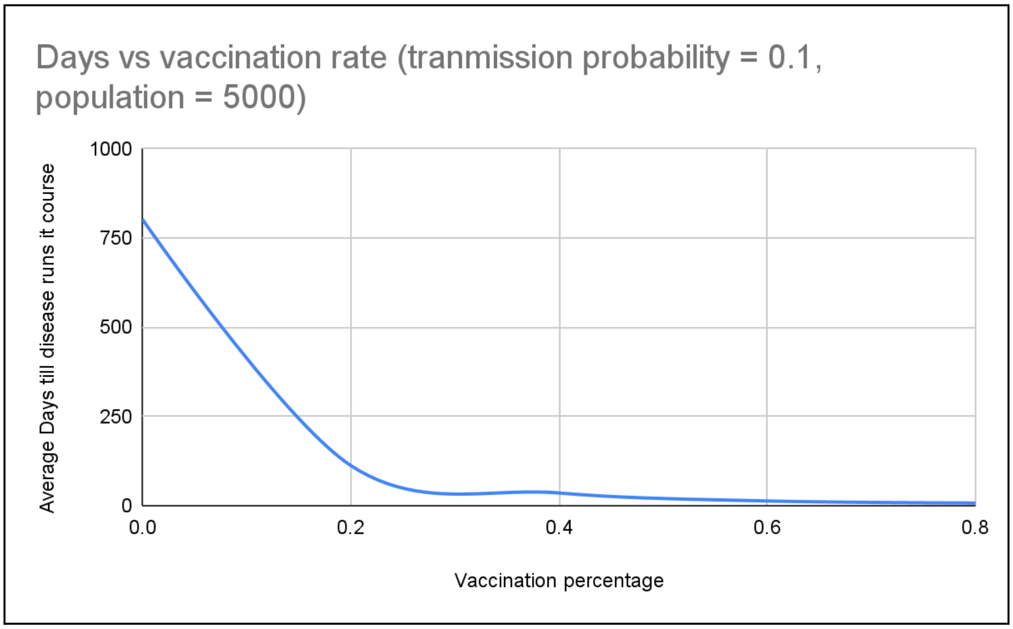


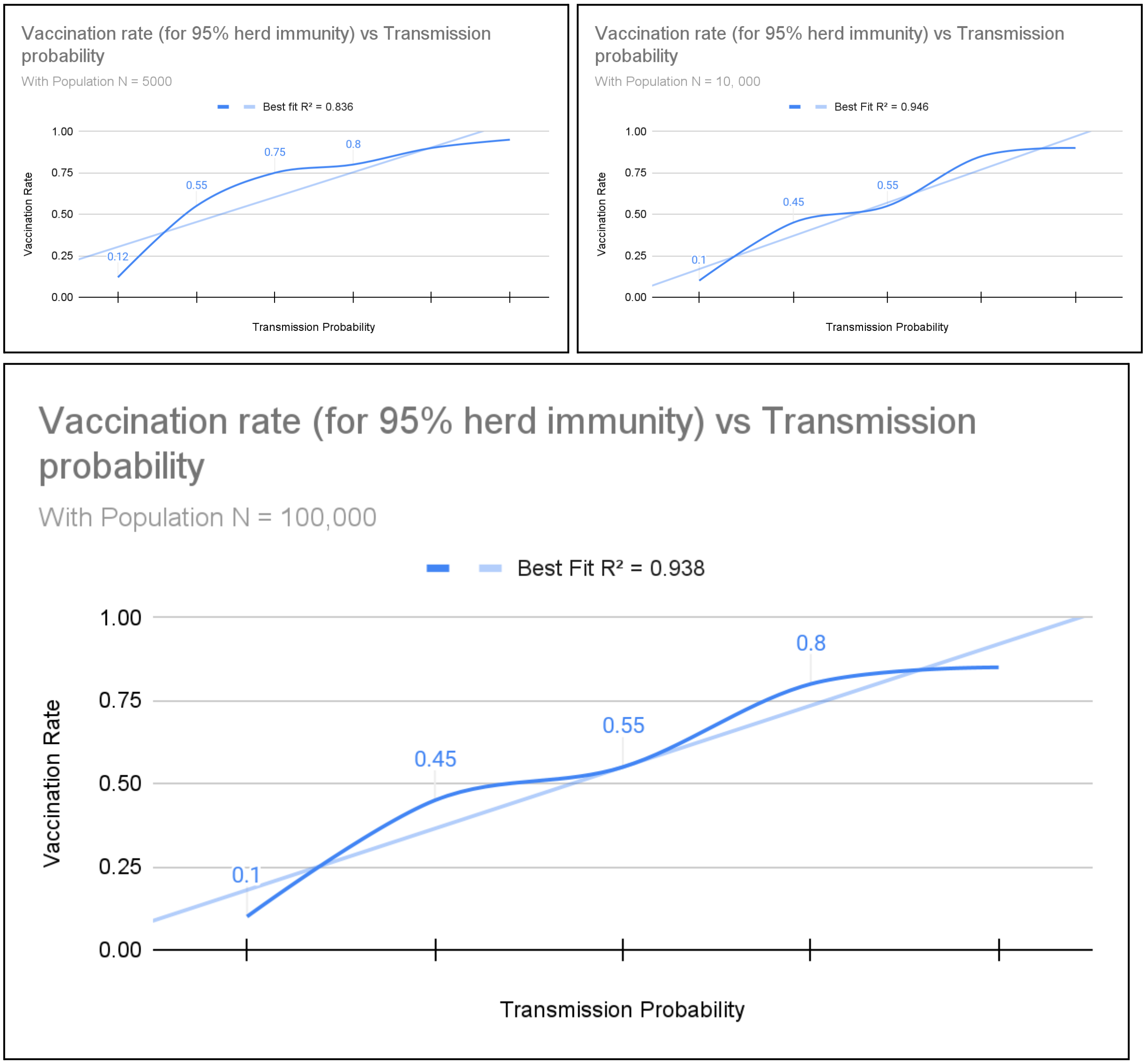
Figure 2: Days For Disease To Run Course Vs Vaccination Rate of Population

With those graphics in mind, we can begin looking into what vaccination level is needed to achieve herd immunity for the population. For the purposes of this project, I will set the metric for herd immunity to be the probability of a non-vaccinated individual that never gets sick to be 95%. Now we have to figure out what percentage of the population needed to be vaccinated to make this happen. From Figure 1, we can see that the days required for the disease to run its course is proportional and often similar in magnitude to the total number of infected individuals. From Figure 2, we can see that the average days required for the disease to run its course drastically drops around a vaccination percentage of 0.2 (which is coincidentally about double the transmission rate). From these two observations, we can assume that the herd immunity case occurs when the vaccination rate is approximately double that of the transmission probability of the disease in question, at least for a small population size. To test this hypothesis, below is a figure, Figure 3, that covers varying transmission probabilities and vaccination rates for a set population size of 5000 people.

Transmission Probability	Total Infected	Total Susceptible	% Vaccinated	% Herd Immunity
0.1	385	4115	0.1	0.9144444444
0.1	337	4113	0.11	0.9242696629
0.1	172	4228	0.12	0.9609090909
0.1	62	4288	0.13	0.9857471264
0.25	2736	1764	0.1	0.392
0.25	2064	1936	0.2	0.484
0.25	1731	1769	0.3	0.5054285714
0.25	1239	1761	0.4	0.587
0.25	523	1977	0.5	0.7908
0.25	423	1577	0.6	0.7885
0.25	106	1394	0.7	0.9293333333
0.25	53	1197	0.75	0.9576
0.25	21	979	0.8	0.979
0.5	1912	588	0.5	0.2352
0.5	439	811	0.75	0.6488
0.5	135	865	0.8	0.865
0.5	91	659	0.85	0.8786666667
0.5	40	460	0.9	0.92
0.5	11	239	0.95	0.956

Figure 3: Vaccination Percentage & Herd Immunities

As we can see in the table, the vaccination rate required for herd immunity can vary greatly based on the transmission rate of a disease. For a disease with a low transmission rate such as the rate found for COVID-19 previously (for present day, when it initially broke out it was much higher), the vaccination rate required for herd immunity is also relatively low. For diseases with higher transmission rates, the required vaccination rate for herd immunity is relatively much higher. Previously, it was seen that the vaccination rate required for immunity was about double the transmission rate of a disease. This was concluded from a small population size, so it is necessary to check if there is an effect of population size on this behavior. On the next page are some plots of vaccination rate required for 95% herd immunity vs transmission rate with varying population sizes.



Figures 4-6 (Left, Right, Bottom): Vaccination Rate Required for Herd Immunity vs Transmission Probability

As we can see from the plots, the vaccination rate and transmission probability have a relatively linear relationship that is more of a one-to-one relationship than a two-to-one (in other words the vaccination rate must be double the transmission rate for herd immunity to be achieved). By increasing the population to a more realistic sample size, we see that the trend between vaccination rate required for herd immunity and transmission rate converges towards a one-to-one ratio, where the vacation rate need only be equal to the transmission rate for herd immunity to be achieved. These simulations already account for a simple mutation rate where the population mutates with a minimal rate of every 100 transmissions and a maximum rate of every 1000 transmissions (base of 100, but if the population is more than 100, it is 1% of the total population size, up until 1000 is reached).

b. Future Analysis

There are several things worth delving into deeper should a more detailed analysis be desired. In this section, I will cover a couple simpler analyses such as mutation rates and effects on vaccination rate required for herd immunity, as well as the effect of incubation period on the spread of the disease.

Regarding the mutation rate, the general trend is that the time required for the disease to run its course increases substantially, with the increase in simulation time likely being also correlated with the magnitude of the mutation rate relative to the total size of the population. Since the a increased mutation rate (i.e. a lower number of transmissions are required before a disease can mutate) is proportional to longer durations of simulations (i.e. the disease stays in the population longer and infects more people), the vaccination rate required for herd immunity will also naturally increase proportionally to counteract the increased infectivity of the disease and its mutations.

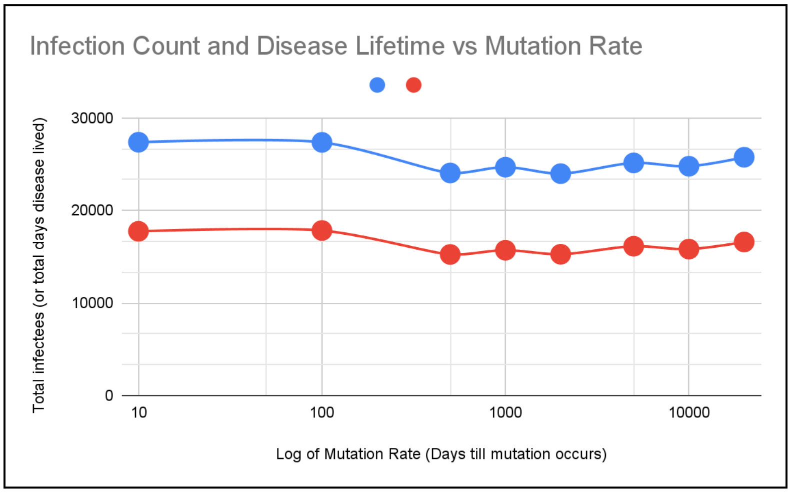


Figure 7: Infection Count & Disease Lifetime Vs Log of Mutation Rate

As we can see in the above figure, the duration of time that a disease will stay in the population as well as the total number of people infected increase with an increased mutation rate just like expected. Upon further observation we can assume that after a threshold mutation rate is met, slowing down the mutation rate barely has any further effect on the disease spread.

On the matter of incubation, the extended period in which an individual person can infect others allows the disease to infect more people and also remain in the population for longer as a result. Additionally, in a realistic situation, having an incubation period means that a sick individual is less likely to social distance or isolate while being sick and able to transmit the disease than they would have if they saw symptoms shortly after being infected (in other words, if there was no incubation period and symptoms showed soon after the initial infection rather than days later). This will likely also play a role in increasing the correlated vaccination rate required to attain herd immunity, as the infectivity of the disease overall is increased significantly.

Moving onto other potential topics not explored in this report, some functionalities worth exploring are the social distancing practices of sick persons which may help partly negate the spread of a disease and its mutations.

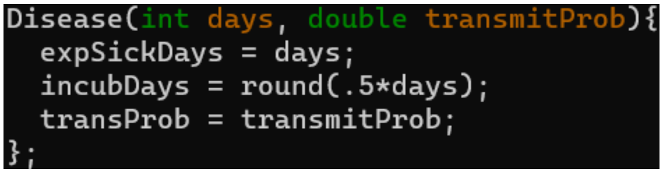
References

1“CDC Covid Data tracker,” *Centers for Disease Control and Prevention* Available: https://covid.cdc.gov/covid-data-tracker/#datatracker-home.

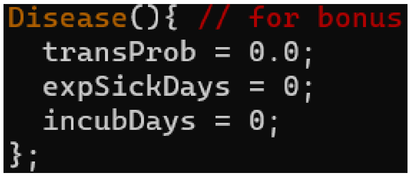
2*What to do if you test positive for covid-19* Available: https://doh.wa.gov/sites/default/files/2022-02/COVIDcasepositive.pdf.

Appendices

Graphics



```
Disease(int days, double transmitProb){
  expSickDays = days;
  incubDays = round(.5*days);
  transProb = transmitProb;
};
```



```
Disease(){ // for bonus
  transProb = 0.0;
  expSickDays = 0;
  incubDays = 0;
};
```

Figures

Transmission Probability (p)	Percent Vaccinated	Total infected (then recovered on average)	Total susceptible	Days Disease ran its course (measured in days)
0.2	0.0	18	4982	28.3
0.2	0.5	9	4991	12
0.4	0.0	132.1	4867.9	173.4
0.4	0.5	14.2	4886.8	13.78
0.8	0.0	910.6	4099.5	940
0.8	0.5	63	4937	9.4

Figure 1: Neighbor-Infection Model

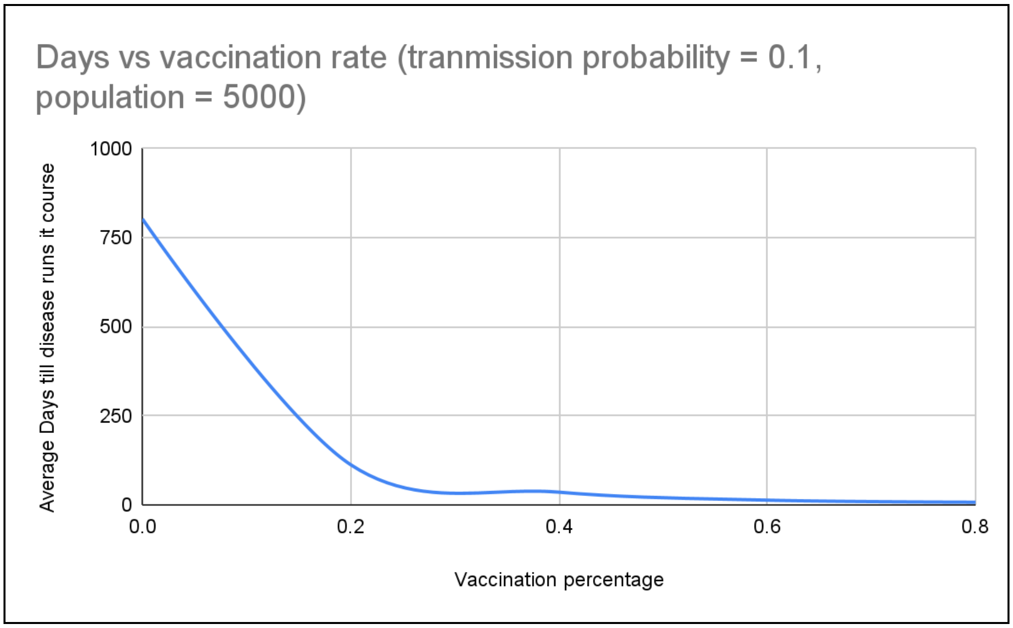
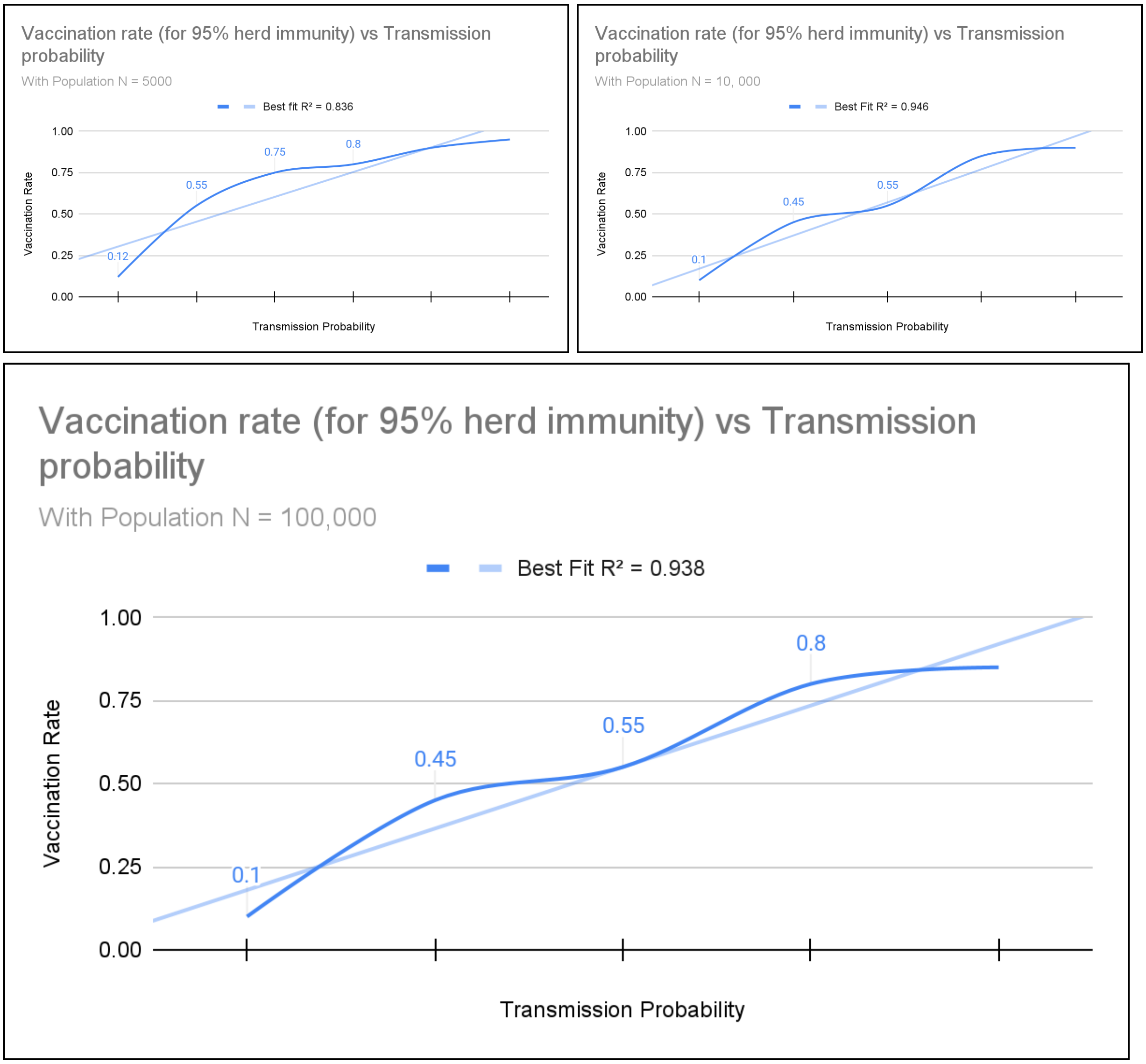


Figure 2: Days For Disease To Run Course Vs Vaccination Rate of Population

Transmission Probability	Total Infected	Total Susceptible	% Vaccinated	% Herd Immunity
0.1	385	4115	0.1	0.9144444444
0.1	337	4113	0.11	0.9242696629
0.1	172	4228	0.12	0.9609090909
0.1	62	4288	0.13	0.9857471264
0.25	2736	1764	0.1	0.392
0.25	2064	1936	0.2	0.484
0.25	1731	1769	0.3	0.5054285714
0.25	1239	1761	0.4	0.587
0.25	523	1977	0.5	0.7908
0.25	423	1577	0.6	0.7885
0.25	106	1394	0.7	0.9293333333
0.25	53	1197	0.75	0.9576
0.25	21	979	0.8	0.979
0.5	1912	588	0.5	0.2352
0.5	439	811	0.75	0.6488
0.5	135	865	0.8	0.865
0.5	91	659	0.85	0.8786666667
0.5	40	460	0.9	0.92
0.5	11	239	0.95	0.956

Figure 3: Vaccination Percentage & Herd Immunities



Figures 4-6 (Left, Right, Bottom): Vaccination Rate Required for Herd Immunity vs Transmission Probability

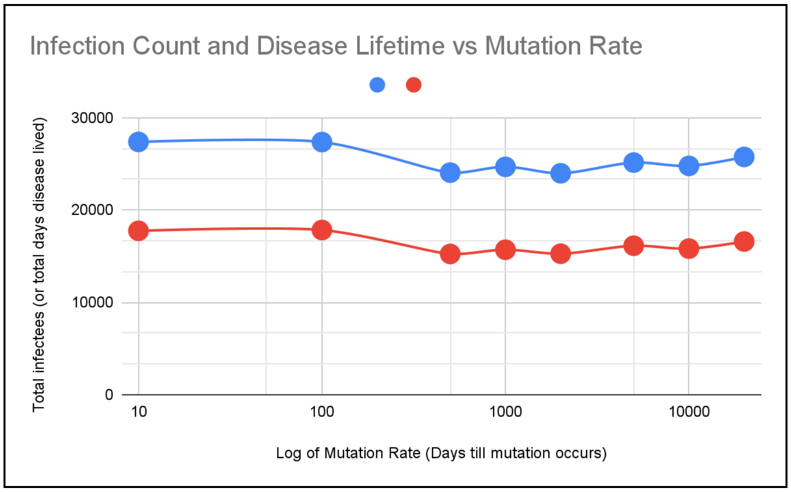


Figure 7: Infection Count & Disease Lifetime Vs Log of Mutation Rate