# Herd Immunity

CSCI 4314: Final Project Report (Professor Orit Peleg)

Project Members: Curry Bushner & Gayathri Gude

Thursday, May 9, 2019

Github Link: https://github.com/gayathrigude/CSCI4314

**Abstract**:

Vaccination is a cornerstone of public health in the modern world. It protects individuals from infection of disease. With high rates of vaccination in a population, vaccination helps to protect even the unvaccinated. This name of the phenomena leading to the protection of the unvaccinated population is Herd Immunity. Essentially, the higher vaccination rate of a population, the lower the likelihood of susceptible individuals coming into contact with infectious individuals. Our goal in this project was to create a model exhibiting the effects of this phenomena using the common SIR spreading model.

**Introduction:**

This project was directly inspired by the University of Pittsburgh's Framework for Reconstructing Epidemiological Dynamics, or FRED. This software allows one to view the spread of measles through numerous counties across the US, including Boulder. It shows the difference in infection rates between communities with 80% vaccination and 95% vaccination rates [2].

The SIR model was extensively studied to create various depictions of graphs.  In addition, a simulation was created to represent populations at varying levels of vaccination prevalence and disease vitality within Boulder. Different rates of recovery were also studied as part of the parameters of the project, such as: susceptibility, infectibilty, and recoverability.

**Related Work on Vaccinations:**

A 2012 article from the Daily Camera stated that Colorado's vaccination rate had dropped from 92% to 85% and that 11% of elementary students in Boulder had not received all recommended vaccines [1]. There has been a lot of research regarding vaccination in Colorado. The state has been ranked 45th in childcare vaccination. Quite evidently, the state of Colorado is lagging in vaccination treatments. Boulder, specifically, is at an even lower vaccination usage compared to the rest of the state of Colorado [3, 4].

An interactive simulation by the University of Berlin's Dirk Brockman was also a major source of inspiration [6]. This simulation allows users to vary vaccination rates and population reproduction rates [6]. With this simulation, it was made possible to obtain the following images, with green, red, tan, and grey avatars representing vaccinated, infected, susceptible, and recovered individuals, respectively. These images show the spread of disease in a society with low, medium, and high vaccination rates:

(i) Low Vaccination: In Figure 1, a society is depicted with there is a zero vaccination rate; this is quite extreme of a case, and not the most prevalent society. Generally, as you can see below, when a disease spreads with zero vaccinations to inhibit its spread, nearly the entire un-immunized population becomes infected.

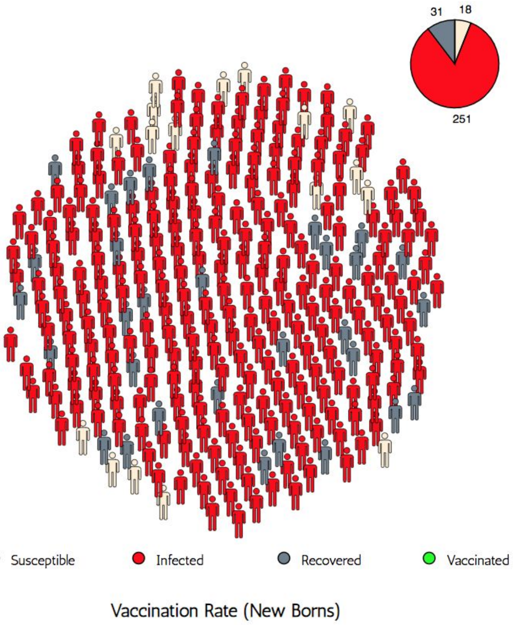


Figure 1

(ii) Medium Vaccination: Figure 2 depicts a society with a 50% vaccination rate as a disease spread through the population. As you can see, the disease still manages to infect nearly the entire susceptible population.

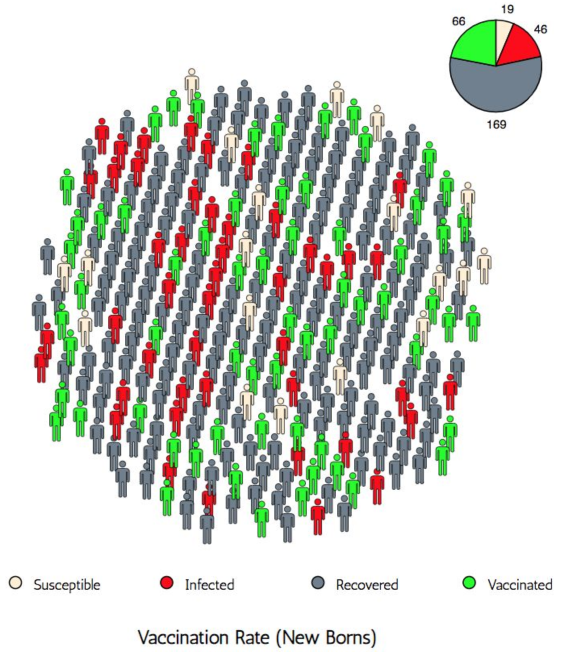


Figure 2

(iii) High Vaccination: Figure 3, the spread of infection is depicted in a society with a 90% vaccination rate. One can see the effects of herd immunity as much of the susceptible population is protected from infection by their vaccinated counterparts.

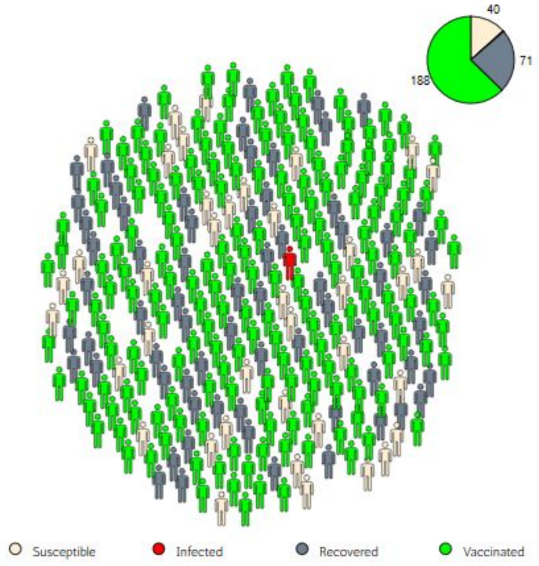


Figure 3

It is important to note that these models work by introducing new individuals with a probability to be vaccinated, and that the models are not initialized with a percentage of the population vaccinated. This is why the vaccination rates do not match the true vaccinated population of the simulations. Brockman's simulation also has dynamic network structure with a small degree of movement for its individuals. We encourage readers to try the simulation themselves.

**Related Work on the SIR Model:**

Most disease spreading models implement a variation of an SIR Model. The SIR Model is a differential equation based model [8]. SIR is an acronym for: susceptible, infected, and recovered zones within a population. This implements, essentially, a mathematical model to show population dynamics when a disease, along with a vaccination is present within a society [8]. In this model, the following variables are recorded for the mathematical calculations to create a dynamic model:

- S = S(t) → Number of Susceptible Individuals
- I = I(t) → Number of Infected Individuals
- R = R(t) → Number of Recovered Individuals
- b → Number of Neighbors of Each Node
- k → Fraction of Infected Expected to Recover at Each Time Step

The following are the SIR Model set of equations which could be used to find the expected change in population states for an SIR model [8]. Each represents the change in susceptible, infected, and recoverd populations after a simulation time step.

- $dS/dt = -b*S(t)* I(t)$
- $dR/dt = k*I(t)$
- $dl/dt = b*S(t)*I(t)-k*I(t)$

In our work, we focus on implementing an SIR simulation, not examining the math behind it.

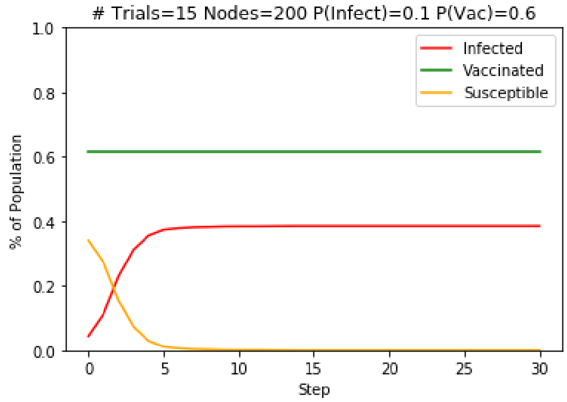
**Methods:**

We built two different models during our project using *Python*. We relied on the python package, *Network*, which provides several out of the box tools to create network models.

We initially attempted to model an extremely simplified version a college campus social structure. This first model used a modified preferential treatment model to generate the structure of its network. During the construction of a network using preferential attachment,new nodes are more likely to be attached to nodes of high degree. This is usually done by randomly choosing from a list of endpoints of every edge in the network.

Our first model's network was constructed in two steps. Given **N** nodes to add to the network, we gave every node a baseline **X** number of preferentially attached neighbors. Then for some portion of **N**, we preferentially attached another **Y** neighbors with probability **P**. This meant one set of nodes had **X** neighbors, and another set of nodes had **X+P×Y** expected neighbors. For this model we only implemented states of *Infection*, *Susceptible*, and *Vaccinated* in our spreading process. The spreading process itself will be described later. The graph structures obtained by this model proved to be too highly connected. The disease was able to spread to all uninfected nodes because of the abundance of edges. To reach the entire network, only a small number of edges would need to be crossed.

The following graph shows the complete infection of susceptible nodes:



After noticing the downfalls of this model, we decided to build a graph that could be represented in a 2D plane without crossedges. This would allow an easier visualization and comparison of our algorithm with the Brockman algorithm.

We constructed hexagonal (degree = 3) and triangular lattice (degree = 6) graphs using *networkx*, and proceeded to add additional states to our spreading process.

The parameters of our spreading process include:

- **G_type** - either Triangular, or Hexagonal
- **m** - Number of Rows
- **n** - Number of Columns
- **steps** - Number of spreading process iterations
- Rates:
  - **P_vac** - Vaccination
  - **P_inf** - Infection
  - **P_rec** - Recovery
  - **P_rep** - Birth
  - **P_death -** Death

Our network's node $i$ has a variable $S_i$ corresponding to *Vaccinated*, *Susceptible*, *Infected*, *Recovered*, or *Dead*. First, we initiated the spreading process by looping through the nodes in the graph. With probability **P_vac**, a node gains immunity to the disease. If it does not gain immunity, with **P_inf**, the node becomes *Infected*. Otherwise, the node's state $S$ is designated *Susceptible*. Second, we defined a function *spread_step*(), which simulated the spread of disease in a single time step. Again, we loop through each node in the graph. If a node is *Infected* we infect its *Susceptible* neighbors one by one with **P_inf**. Furthermore, for nodes that were infected at the beginning of this time step, each becomes *Recovered* with probability **P_rec**. If they do not recover, nodes become *Dead* with probability **P_death**. Finally, each node is re-initialized with probability **P_rep**, simulating reproduction in a population. A "reborn" node has its state determined the same way as during the disease initialization process.

After running our simulation, we used

*matplotlib*'s function *FuncAnimation* to create a dynamic plot showing each time step of the disease as it spreads through the network.

## Results

We present four images of graphs showing the spread of disease through a Hexagonal lattice network of 3721 nodes over 100 spreading steps. Each simulation has parameters: **P_inf**=.2, **P_rec**=0.1, **P_rep**=0.05, and **P_death**=0.01.

Figure 4 shows the initial state of a network with a 50% vaccination rate. Figure 5 shows the state of this network after 100 spreading steps. Figure 6 shows the initial state of a network with a 90% vaccination rate. Figure 7 shows the state of this network after 100 spreading steps. Green, yellow, red, cyan, and white nodes represent vaccinated, susceptible, infected, recovered, and dead individuals, respectively.

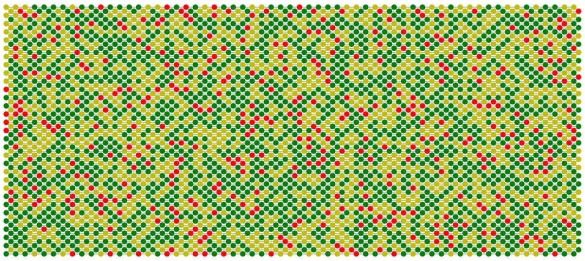


Figure 4

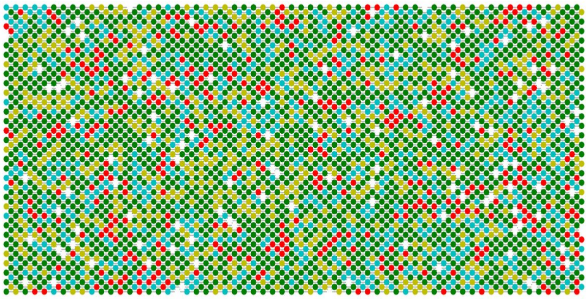


Figure 5

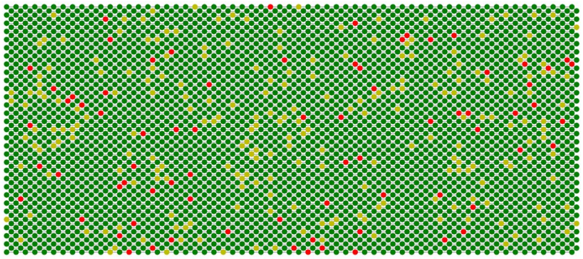


Figure 6

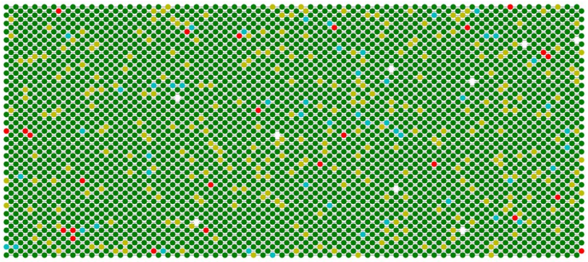


Figure 7

Figures 4 and 5 show how lower vaccination rates allow infections to spread and infect a large portion of the susceptible population. Figures 6 and 7 illustrate the effect of herd immunity as the majority of the susceptible population remains uninfected. These visualizations are exactly what we set out to achieve.

## Conclusion

The model that we created is remarkably similar to Brockman's simulation, but uses a static Triangular or Hexagonal lattice network structure instead of a a dynamic network structure. Otherwise, our simulation is extremely similar, with the addition of a probability of death for individuals. One key difference between our simulation and Brockman's is that ours initiates the population with vaccinated individuals, instead of having the produced via reproduction (replacement) of the population. Initially we didn't think that we would be able to have the spreading process shown in real time, but we were able to achieve this goal via animation functions of *matplotlib*.

### Future Works

Promising work for the future would be to create simulations illustrating the effects of antibiotic resistance. Much research has been conducted regarding this field, where it has been said that antibiotic resistance will increase and lead to higher mortality rates. There are six different ways antibiotic resistance is commonly modeled [9]. It would be intriguing investigating how the results of the models differ. The six common models are briefly described as follows:

- Single Strain
    - independent infections from sensitive and resistant strains
- Superinfection
    - hosts can be infected with both sensitive and resistant strains, but their contagiousness differs depending on their proportion of infection
- Exclusive Infection
    - hosts can only be infected with either the sensitive or resistance strain
- Replacement Infection
    - A resistant strain can replace a sensitive strain in one time step
- Unidirectional Conversion
    - A sensitive strain can convert to a resistant strain over multiple time steps
- Bidirectional Conversion
    - A sensitive strain can convert to a resistant strain, and vice versa, over multiple time steps

In all these listed models above, parameter sensitivity is accessed. These models give an understanding for how antibiotic resistance works and various extents of levels of resistance. Essentially, the intake of antibiotics will increase antibiotic resistance over time, making the use of antibiotics increasingly ineffective. This would be a interesting dynamic to simulate, and allows several different simulations to be created that apply to the same concept of antibiotic resistance [9].

### Project Contributions

This project was broken down into past research that led us into current findings. Though we both worked extensively together on the project as a hole, we had some focus points. While the code was worked on by both team members, Curry Bushner led the code implementation due to his knowledge in the needed python modules and geared the code implementation the correct way, while taking extra initiatives. Likewise, the background research, presentation (verbal and physical), and project implementation structure was led by Gayathri Gude, while taking those extra initiatives as well; however, both still heavily worked on it.

## Sources

[1] Bounds/Camera, A. (2012, January 30). Boulder Valley vaccination rates low. Retrieved from http://www.dailycamera.com/news/ci_19848081

[2] P. (2019). FRED U.S. Measles Simulator. Retrieved March 21, 2019, from/https://fred.publichealth.pitt.edu/measles

[3] School and child care immunization data. (2018, October 16). Retrieved from https://www.colorado.gov/pacific/cdphe/school-and-child-care-immunization-data

[4] Whaley, M., & Whaley, M. (2018, June 28). Colorado was a laggard in childhood vaccination rates for years. All that appears to be changing. Retrieved March 21, 2019, from https://www.denverpost.com/2018/06/28/colorado-rising-vaccination-rates/

[5] Wheaton, W.D., *U.S. Synthetic Population 2010 Version 1.0 Quick Start Guide*, RTI International, May 2014

[6] Brockman, Dirk, How Herd Immunity Works. Retrieved March 21, 2019 from http://rocs.hu-berlin.de/D3/herd/

[7] Newman, M. (2002). Spread of epidemic disease on networks. *Physical review. E, Statistical, nonlinear, and soft matter physics, 66 1 Pt 2*, 016128 .

[8] Smith, D., & Moore, L. (2019). The SIR Model for Spread of Disease - The Differential Equation Model. Retrieved April, 14, from https://www.maa.org/press/periodicals/loci/joma/the-sir-model-for-spread-of-disease-the-differential-equation-model

[9] Spicknall, I. H., Foxman, B., Marrs, C. F., & Eisenberg, J. N. (2013, August 15). A modeling framework for the evolution and spread of antibiotic resistance: Literature review and model categorization. Retrieved May 1, 2019, from https://www.ncbi.nlm.nih.gov/pubmed/23660797